

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



A DISTRIBUTIONAL SEMANTICS PERSPECTIVE OF
LEXICAL RESOURCES FOR AFFECT ANALYSIS: AN
APPLICATION TO EXTREMIST NARRATIVES

TESIS DOCTORAL

ÓSCAR ARAQUE IBORRA
Ingeniero de Telecomunicación

2020

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN



A DISTRIBUTIONAL SEMANTICS PERSPECTIVE OF
LEXICAL RESOURCES FOR AFFECT ANALYSIS: AN
APPLICATION TO EXTREMIST NARRATIVES

TESIS DOCTORAL

ÓSCAR ARAQUE IBORRA
Ingeniero de Telecomunicación

2020

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
TELEMÁTICOS

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE MADRID



A DISTRIBUTIONAL SEMANTICS PERSPECTIVE OF
LEXICAL RESOURCES FOR AFFECT ANALYSIS: AN
APPLICATION TO EXTREMIST NARRATIVES

AUTOR:

ÓSCAR ARAQUE IBORRA
Ingeniero de Telecomunicación

TUTOR:

CARLOS A. IGLESIAS
Doctor Ingeniero de Telecomunicación

2020



Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día _____ de _____ de _____.

Presidente: _____

Vocal: _____

Vocal: _____

Vocal: _____

Secretario: _____

Suplente: _____

Suplente: _____

Realizado el acto de defensa y lectura de la Tesis el día _____ de _____ de _____. en la E.T.S.I. Telecomunicación habiendo obtenido la calificación de _____.

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

A mi padre, a mi madre, y a todas las personas que me han ayudado en este camino.

Agradecimientos

Durante el desarrollo de la tesis me han acompañado un gran número de personas, y me gustaría darles a todas mi más sentido agradecimiento. Sin ellas no hubiera sido posible.

Quiero empezar con mi familia, que me ha apoyado desde incluso antes de empezar, dándome ánimos. A mis padres, que han fomentado mi carrera desde el inicio. A mis tíos y tías, que han estado para interesarse y darme fuerzas, y a mi abuela, que siempre tiene buenas palabras.

Claramente, a Cristina, que ha estado siempre a mi lado, ayudando en todo momento, y también porque me ha seguido en mis peripecias y viajes que, sin ella, no habrían sido iguales. Si hay una persona que ha tenido que aguantarme, es ella. Muchas gracias.

A Carlos Ángel, que ha sido un excelente tutor con el que he podido aprender y formarme. Esas largas charlas en el despacho han servido más de una vez y más de dos para meterle caña al trabajo, y también a mí mismo. También al resto del GSI: J, Álvaro, Sergio, por el día a día, las comidas y los cafés, sin olvidar el MK. A los que han pasado por los laboratorios y han hecho muchas tardes amenas: Manu, Alberto, Enrique y tantos otros. En Teleco, a todos los compañeros que me han acompañado en el viaje, siempre con una risa: Adri, Iván, Alfonso, y tantos otros.

No me quiero olvidar a la gente que me acogió en Milton Keynes, en especial a Miriam y a Martino, que hicieron de la estancia algo tremendamente agradable y divertido. Por último, le doy las gracias al DIT y a la ETSIT por haberme acogido y hacer del trabajo un placer.

No puedo olvidarme de la gente que me ha acompañado desde hace más de una década: Belén, Tomé, Rodri, David, Javi² y Floren. También quiero agradecerle el apoyo a toda la gente de televallecas: Lara, Álvaro, Raquel, Silvia, Sergio, Sandra, Kuba, Berta, Jose, y todos los demás.

A todos, mil gracias.

Resumen

Los campos de Análisis de Sentimientos y Emociones son prominentes en el Procesado de Lenguaje Natural (NLP, en inglés). Modelar emociones requiere de un número de técnicas que, en la mayoría de los casos, se comparten con otras áreas del NLP. Es este contexto, esta tesis aborda el desarrollo de métodos de aprendizaje automático novedosos a través de la combinación de características superficiales y profundas en Análisis de Sentimientos y Emociones. Tras observar los resultados obtenidos, los métodos desarrollados han sido adaptados a otras áreas del NLP, transfiriendo así los resultados obtenidos a campos que se pueden beneficiar en gran manera de dichas técnicas.

De esta manera, hemos diseñado una taxonomía que permite clasificar diferentes tipos de técnicas en Análisis de Sentimientos, según las características y combinaciones usadas. Esta taxonomía nos ha permitido desarrollar varios modelos de aprendizaje automático que combinan características superficiales y profundas, además de un variedad de modelos de aprendizaje, sobre todo las técnicas de aprendizaje profundo. Estos modelos han sido evaluados mediante en entornos de Análisis de Sentimientos, a niveles de documento y aspecto. A continuación, hemos contribuido con una técnica para la generación de léxicos de sentimientos específicos del dominio, usando el algoritmo de retro-propagación. En una línea similar de trabajo, también hemos extendido un método para la generación de léxicos, obteniendo un recurso bilingüe anotado en Inglés e Italiano.

La contribución central de esta tesis es un modelo que explota la similitud semántica en representaciones distribuidas a través de léxicos, extrayendo características mediante el uso de *word embeddings* y un léxico de sentimientos: el modelo SIMON. Hemos observado que SIMON ha mostrado resultado positivos en la evaluación experimental, y ofrece una alternativa potente a las técnicas clásicas de uso de léxicos.

A la luz de esta última contribución, esta tesis estudia la adaptación de los métodos desarrollados a otros campos. Más concretamente, esta tesis ha contribuido a la (a) detección de radicalización, donde hemos adaptado el modelo SIMON, combinándolo con un método de extracción de características emotivas; y a la (b) estimación de valores morales, en la que se ha usado SIMON para emplear un nuevo léxico que hemos generado.

Abstract

Sentiment and Emotion Analysis are prominent fields in Natural Language Processing (NLP) and have contributed to its progress. Modeling emotions requires a number of techniques and methods that, in most of the cases, are shared with other NLP areas. In such a context, this thesis addresses the development of novel machine learning methods through the combination of both surface and deep features for Sentiment and Emotion Analysis. Observing the results obtained, the developed methods have been adapted to other NLP areas, transferring the obtained results to fields that can largely benefit from such novel techniques.

In this way, we have designed a taxonomy that classifies different approaches in Sentiment Analysis, attending to the features and combinations used. This taxonomy has allowed us to develop several machine learning models that combine both surface and deep features, as well as a variety of learning models, with an especial focus on deep learning approaches. These models have been evaluated in document and aspect-level Sentiment Analysis frameworks. Following, we contributed with a novel technique for generating domain-specific sentiment lexicons through the backpropagation algorithm. In a similar line of work, this thesis also extends a method for generating emotion lexicons, obtaining a bilingual resource with emotion annotations in both English and Italian.

The thesis' core contribution is a similarity-based perspective on lexicons that extracts features exploiting both a word embedding model and a sentiment lexicon: the SIMilarity-based sentiment projectiON (SIMON) model. We have observed that SIMON has shown positive results in the experimental evaluation, offering a compelling alternative to classical lexicon usage techniques.

In light of this last contribution, this thesis studies the adaptability of methods that are developed in the context of Sentiment and Emotion Analysis to other fields. More concretely, this thesis has contributed to radicalization detection and moral value estimation. In the area of radicalization detection, we have adapted the SIMON model, combining it with an emotion-driven feature extraction method. Similarly, for moral value estimation, SIMON has been used to exploit a novel lexicon we have generated.

Contents

Resumen	I
Abstract	III
Contents	V
Reader's guide	1
1 Introduction	3
1.1 Motivation	4
1.2 Background	10
1.2.1 Lexicons	10
1.2.2 Word Embedding Models	15
1.2.3 Semantic Similarity	18
1.2.4 Affect Analysis of Online Radicalism	19
1.2.5 Moral Foundations Theory	24
1.3 Hypotheses	26
1.4 Objectives	27
1.5 Document outline	28
2 Methodology	29
2.1 Introduction	30
2.2 Machine Learning Experimentation	31
2.3 Evaluation	33

2.4	Statistical Analysis	34
3	Publications	37
3.1	Introduction	38
3.2	List of central publications	38
3.2.1	An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity	38
3.2.2	MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction	55
3.2.3	A Semantic Similarity-based Perspective of Affect Lexicons for Senti- ment Analysis	67
3.2.4	DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques	82
3.2.5	Enhancing Deep Learning Sentiment Analysis with Ensemble Tech- niques in Social Applications	95
4	General Discussion, Conclusions and Future Research	107
4.1	Overview	108
4.2	Scientific results	110
4.2.1	Objective 1: Surface and Deep methods framework	110
4.2.2	Objective 2: Definition of a lexicon generation methodology	114
4.2.3	Objective 3: Definition of a similarity-based model over distributed representations	119
4.2.4	Objective 4: Transference of a similarity-based model that exploits lexicons to other domains	125
4.2.5	Other	130
4.3	Software Publication	131
4.4	Conclusions	132
4.5	Future Research	137

Bibliography	141
List of Figures	159
List of Tables	161
Glossary	163
Appendix A Publications	165
A.1 Summary of all publications	165
A.2 Rest of publications	167
A.2.1 Radical Text Detection based on Stylometry	167
A.2.2 Senpy: A Framework for Semantic Sentiment and Emotion Analysis Services	176
A.2.3 GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immi- grants and Women on Twitter	180
A.2.4 How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining	189
A.2.5 Neural Domain Adaptation of Sentiment Lexicons	198
A.2.6 Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets	205
A.2.7 Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis	212
A.2.8 Senpy: A Pragmatic Linked Sentiment Analysis Framework	221
A.2.9 Aspect Based Sentiment Analysis of Spanish Tweets	230

Reader's guide

This thesis is presented in the form of a compendium of publications. That is, this document does not pretend to be a full monograph. Instead, the essence of this document lies in the collection of publications by the author. Such publications must comply with the following requirements: the Ph.D. student must appear as the first author, and at least three of these publications must be published in Q1 or Q2 ranked journals, as indexed by JCR or Scopus. These conditions are satisfied, as can be observed in Chapter 3 and Appendix A. Following the guidelines by Universidad Politécnica de Madrid, the structure of this document must comply with a specific structure. Thus, this section intends to guide the reader into this document.

Chapter 1 introduces the main topics to which this thesis revolves. In this way, it details the motivation (Section 1.1) and background (Section 1.2) that have driven the research done during the Ph.D. Following, this chapter also discovers the hypotheses (Section 1.3) and objectives (Section 1.4) that we have followed. These objectives and hypotheses have guided our activities.

In Chapter 2, the full methodology of the thesis is detailed. This chapter describes how we have performed experimentation and model evaluations. It also describes the best practices for machine learning experimentation (Section 2.2), the evaluation methods (Section 2.3), as well as the use of statistical analysis (Section 2.4), a field that has been central to this thesis. These techniques define how we have operated during the accomplished research and how posterior conclusions can be taken from the obtained results.

Chapter 3 contains the central publications of this thesis. Each paper is presented in a separate section to facilitate reading, and it includes a preliminary table with relevant details (e.g., authors, title, journal, impact factor). Throughout the document, there are several publication tables, and all of them refer to the section where the paper can be read in full. Please note that in Chapter 3, not all publications by the Ph.D. student are included. This is done following the guidelines by Universidad Politécnica de Madrid, as this chapter must only include the publications that fulfill the described requirements. A full collection of publications can be found in Appendix A.

Chapter 4 contains a general discussion about the thesis. Firstly, we present an overview of the results obtained during the development of the thesis (Section 4.1). Once those have been described, the scientific results are outlined in relation to the previously presented thesis' objectives (Section 4.2). This chapter also describes the software tools that have emerged as part of the research process (Section 4.3). Finally, the chapter concludes with the main conclusions gathered (Section 4.4), and with some work lines that can be addressed in the future (Section 4.5).

As mentioned, Appendix A contains all the publications of this thesis. A complete table can be found in Section A.1, while Section A.2 contains the full texts of the publications that do not fulfill the university requirements. We would like to stress that several publications have contributed significantly to this thesis and are discussed throughout the document, although do not appear in Chapter 3.

CHAPTER 1

Introduction

Research requires much effort in preparation and documentation. Knowing previous research is vital and must not be ignored. This chapter discusses the context of this Ph.D. thesis, which includes its motivation, relevant background, objectives, and hypotheses.

“On the Shoulders of Giants.” A popular search engine.

1.1 Motivation

The consistent growth of users and user-generated content in many web sites, social networks, and online consumer platforms such as Twitter, Amazon, and Yelp has incremented the quantity of opinionated information available on the Internet. This tendency, combined with the fast-spreading nature of content online, has turned online user-generated content into a precious asset. Said information has inherent value for many online and offline businesses, as well as research entities, since language usage reflects our opinions, thoughts, emotions, values, and culture, as we communicate with others. Online opinions can, for example, directly affect future sales in a wide range of fields such as hotels, e-commerce, restaurants, and electronics (Chevalier et al., 2006; F. Zhu et al., 2010). Consequently, interest in Natural Language Processing (NLP) techniques as a way of automatically extracting and analyzing user-generated content has risen (Liu, 2015).

In this context, many NLP methods are being developed and applied in order to analyze the massive quantity of information that is available on the Internet. In particular, this thesis revolves around Sentiment Analysis (SA) and Emotion Analysis: two increasingly growing tasks (Liu, 2015) whose goals are the study of opinions, sentiments, and emotions expressed in text generated by a human party. In Sentiment Analysis works, text is generally annotated into several categories, being *positive*, *negative*, and *neutral* the most common. These annotations can be associated with a numerical sentiment score, usually in the $[-1, 1]$ range.

Since the study of opinions in text comprises a broad range of possibilities, Sentiment Analysis tasks can be classified into three categories, attending to the granularity level (Liu, 2015): *document level*, as in a movie review that can span one or more paragraphs; *sentence level*, where sentiment is bounded to a single sentence of text; and *aspect level*, in which case sentiment is associated with a word or small group of words (e.g., *good food*, *short battery duration*).

In comparison, Emotion Analysis represents an extension over the negative/positive distinction that is done in Sentiment Analysis, considering individual emotions rather than a binary label. In this way, among the most common emotions that are measured, we can find *anger*, *disgust*, *joy*, *fear*, *sadness* and *surprise*, which roughly correspond to the ones described in the work of Paul Ekman (Ekman, 1992). In such a light, one can consider Emotion Analysis as an extension over Sentiment Analysis (Staiano et al., 2014), both belonging to Affect Computing.

Even though both Sentiment Analysis and Emotion Analysis represent different research

problems, both share many challenges, methodologies, and, in general, approaches. As such, and in order to facilitate the reading, many of the descriptions that appear in the rest of the section apply both to Sentiment and Emotion Analysis, even if only sentiment is mentioned.

Due to the subjective nature of Sentiment Analysis and Emotion Analysis, it is of no surprise that sentiment and emotion words are a key indicator of subjective polarity. For instance, *good*, *great* and *brilliant* can be considered positive words while *bad*, *worse* and *disastrous* can express negative attitudes. Consequently, sentiment lexicons, which gather sentiment words, are used extensively by the research community (Ravi et al., 2015a). Sentiment lexicons can be organized in three types, attending to which information is contained in them (Wu et al., 2013; Peng et al., 2018): (i) those who contain only sentiment words, a list of sentiment-bearing words; (ii) the ones that are formed by both sentiment words and polarity orientations, a list of words with only positive and negative annotations; and (iii) the lexicons that offer sentiment words with orientation and intensity (Taboada et al., 2011), a list of words with scalar numerical values.

The use of lexicon-based techniques has a number of advantages (Taboada et al., 2011). First, the linguistic content can be taken into account through mechanisms such as sentiment valence shifting (Shanahan et al., 2006) considering both intensifiers (e.g., *very bad*) and negations (e.g., *not happy*). In addition, sentiment orientation of lexical entities can be differentiated based on their characteristics. Moreover, language-dependent characteristics can be included in these approaches. Nevertheless, lexicon-based approaches have several drawbacks: the need of a lexicon that is consistent and reliable (Taboada et al., 2011), as well as the variability of opinion words across domains (Peter D Turney, 2002), contexts (Ding et al., 2008), and languages (Perez-Rosas et al., 2012). These dependencies make it hard to maintain independent domain lexicons (Qiu et al., 2009).

The most common approach that makes use of sentiment or emotion lexicons is keyword matching, also called keyword spotting (Cambria, Schuller, et al., 2013). Basically, this technique involves detecting the presence of certain sentiment bearer words, thus obtaining the sentiment estimation as an aggregate of the associated sentiment values. Although this method is certainly simple and computationally cheap, it is also limited, as it happens in the case of domain adaptation. Sentiment words can display variations in their polarity values depending on the contextual domain (Peter D Turney, 2002), language (Perez-Rosas et al., 2012), or even context (Ding et al., 2008), causing lexicon-based approaches to decrease their classification performance. An additional challenge that arises in relation to the use of sentiment lexicons is the selection between the variety of these resources. It is still an open question how to select and process sentiment and emotion lexicons (Bravo-Marquez

et al., 2014), as well as how this selection affects the performance of sentiment and emotion classification tasks.

As discussed, exploiting sentiment and emotions lexicons constitutes a research problem that is being actively studied. Similarly, obtaining high-quality lexicons is an active subject of research (S. M. Mohammad and Peter D. Turney, 2010). Generally speaking, lexicon acquisition has been implemented through three distinct methodologies: by manual creation (e.g., crowdsourcing annotations), by a dictionary-driven method, or by automatic derivation from already annotated corpora. While the two first approaches provide more precise lexicons, the latter usually grants a higher coverage. Regardless of the chosen approach, when used as baselines or as additional features for more complex learning models, lexicons’ quality is not usually evaluated. That is, the performances used for comparison are rather weak, a fact that could be arguably seen to slow progress in the field.

Additional challenges that arise when trying to obtain sentiment or emotion lexicons are related to the lexicons’ annotations. It is worth taking into account the fact that, as discussed previously, words can vary their sentiment or emotion values across different uses of the language, time, and domain (Hamilton et al., 2016). Also, even the selection of sentiment or emotion words are dependent on the domain (Peter D Turney, 2002). For these reasons, the task of automatically generating domain-independent and domain-adapted lexicons flexibly represents a significant research problem.

While sentiment and emotions lexicons are indeed a relevant resource frequently used in the field, nowadays the predominant approaches rely on *machine learning* techniques (Pang et al., 2002; Read, 2005; Wang et al., 2012). Traditional approaches use the Bag Of Words (BOW) method, where a document is mapped to a numerical feature vector, which offers a machine learning algorithm a unified representation of the analyzed text. Although the Bag Of Words approach is quite popular, simple, and efficient, a fraction of the information from the original natural language is lost (Xia et al., 2010); e.g., word order is disrupted, and syntactic structures are broken. Accordingly, to improve the performance offered by the BOW method, various types of features have been exploited, such as higher-order *n-grams* (Pak et al., 2010). Another type of feature that can be used is Part-of-Speech (PoS) tagging, which extracts information along the lines of syntactic analysis, as described in Gimpel et al. (2010). Some authors refer to this kind of features as surface forms, as they consist of lexical and syntactical information that relies on the patter of the text, rather than on its semantic aspects. In this thesis, we follow such a nomenclature.

In practice, extracting complex features from text, examining which features are relevant, and selecting a learning algorithm are fundamental questions that are present in the

machine learning methods (A. Agarwal et al., 2011; Sharma et al., 2012; Wilson et al., 2009). Mentioned traditional approaches rely on manual feature engineering, which is costly and time-consuming.

As advancement in the effort to effectively obtain text representations, we find *distributed representations*, or more concretely word embeddings (Mikolov et al., 2013; Le et al., 2014), which are considered a relevant characteristic of state-of-the-art sentiment and emotion analysis systems.

These techniques encode text into fixed-length vectors that can be used directly by machine learning methods, being neural networks the most used recently (Collobert et al., 2011). The use of (deep) neural networks enables the system to learn complex features automatically extracted from data, minimizing manual domain-oriented efforts (Bengio, 2009; Alpaydin, 2014). Lately, neural networks have shown excellent performance in Natural Language Processing tasks, including sentiment and emotion analysis (Collobert et al., 2011).

The main idea of neural techniques is to learn complex features extracted from data with minimum external contribution (Bengio, 2009) using deep neural networks (Alpaydin, 2014). Contrary to the mentioned traditional methods, neural learners do not need to be passed manually crafted features: they automatically learn new complex features.

Nevertheless, a requirement of these approaches is the use of large amounts of data, normally annotated (Mikolov et al., 2013). Both automatic feature extraction and availability of resources are very important when comparing the traditional machine learning approach and novel deep learning techniques. Efficiently training neural networks in small datasets is still an open challenge, whose resolutions could improve sentiment and emotion analysis systems.

However, it is not clear whether the specialization capacity of traditional approaches towards a domain can be surpassed with the generalization capacity of deep learning-based models in all NLP tasks, or if it is possible to successfully combine these two techniques in a wide range of applications. Besides, when attending to the application of neural methods to sentiment and emotion analysis, subjective prior knowledge is still a key asset. This takes us back to the effective use of sentiment and emotion lexicons along with neural architectures.

While some core contributions of this thesis are framed in sentiment and emotions analysis, during the development of the thesis a number of advancements have been made in other fields. In this way, the foundations, techniques, and models that have been developed and studied can be applied to solve other problems that fall outside of the sentiment and emotion estimation of texts. An overview of the domains where such techniques have

been applied can be seen in Figure 1.1. In particular, this thesis has contributed to the study and automatic analysis of texts concerning radical narratives, and Moral Foundations Theory (MFT).

Application of Natural Language Processing techniques to social science, psychology, and other fields is common and can yield exciting results (Crowston et al., 2012; Lin et al., 2018; Fernandez et al., 2018). It is clear that the growing amount of human-generated text is of particular interest to social scientists. Still, interesting as this data for scientists may be, it is also clear that single researchers can not manually analyze even a portion of all the available materials. Sampling, analyzing, and categorizing requires a fair amount of domain knowledge. Thus, technologies and methods that support researchers with these massive problems are becoming more important and are nowadays widely used (Wiedemann, 2013).

In this line of thought, as mentioned, this thesis has contributed to different fields by applying, adapting, and designing methods that were initially developed for affect analysis. Also, in some cases, fundamental sentiment or emotion ideas have been adapted to contribute to such fields.

As motivation, we would like to emphasize the application to radicalization detection firstly. Nowadays, modern terrorist organizations are adapting and changing in order to achieve their objectives, overcoming counter-terrorist actions. In this sense, terrorists are effectively using the Internet as part of their propaganda and recruitment strategies (Birmingham et al., 2009). Consequently, online radicalization detection represents a significant concern for counter-terrorists organizations, Law Enforcement Agencies, and the populations in general. Digital traces left on social networks can be used as social signs to detect online radicalism, based on text mining and social analysis techniques (Birmingham et al., 2009), which analyze different aspects of these interactions such as sentiment, emotions, topics, and the social context of users and semantics (Saif et al., 2017).

Additionally, social science studies have highlighted the role of emotions for understanding radicalization (Stekelenburg, 2017; Rice, 2009). A previous work (Stekelenburg, 2017) argues that the radicalization process is fueled by the emotions of anger, contempt, and disgust. Considering these observations, we have detected an interesting application that combines emotion information with domain-aware text representations for radicalization detection in social networks.

Although radicalization is indeed an exciting application field, an additional field where computational approaches can be utilized in social sciences is the study of morality. Moral values influence the way we rationalize and form an opinion upon controversial topics (Sagi et al., 2014b). They are also closely related to our political views (Miles et al., 2015) that

may affect immigration (Grover et al., 2019), political extremism (Alizadeh et al., 2019), and partner violence (Vecina et al., 2015). Also, it has been previously studied how moral values can be used to predict violent protests based on user-generated text (Mooijman et al., 2018). Moral Foundations Theory represents the most established theory in psychology and social sciences. It is also being adopted in the computational social science field since it defines a clear taxonomy of moral values by means of the Moral Foundations Dictionary (MFD) (Graham et al., 2009), which is an essential resource for Natural Language Processing applications.

Concerning the Moral Foundations Dictionary resource, Graham et al. (2009) highlight the difficulty of creating such a resource since linguistic, cultural, and historical context reflect on language usage. Considering the most important limitations of the Moral Foundations Dictionary, we find: (i) a limited amount of terms (lemmas and stem of words); (ii) terms that are rarely used in everyday language (e.g., *homologous*, *apostasy*); (iii) a binary annotation to a bipolar scale, called vice and virtue, that does not indicate strength. In this context, a resource that offers more coverage and an improved numerical annotation could significantly foster research in this area.

In summary, although sentiment and emotion analysis benefits from advancements in Natural Language Processing techniques like distributed representations, many challenges remain in relation to the combination of subjective knowledge with advanced neural networks models.

Models that efficiently exploit these resources while maintaining a low complexity will surely benefit the field. Adaptation of such methods to specific domains is also a challenge, including its generation. Additionally, the use of methods and models initially developed for sentiment and emotion analysis is an exciting field that can benefit both affective computing and social sciences. Adapting and improving these models to function under such conditions opens new research problems.

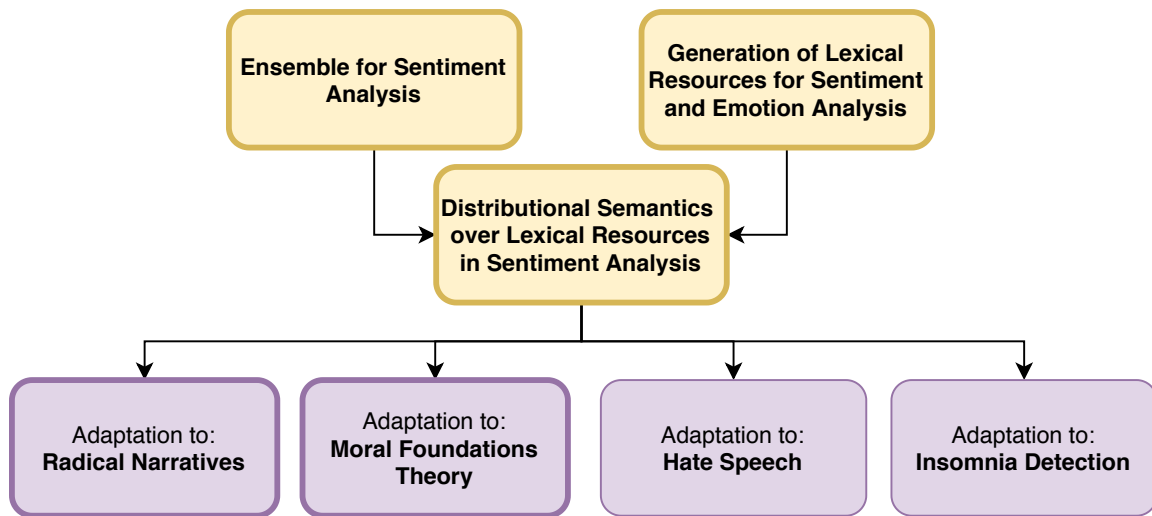


Figure 1.1: Overview of the research fields where this thesis has made contributions. Core contributions are delineated stronger.

1.2 Background

1.2.1 Lexicons

Previously, we have described the importance of lexicons in Sentiment Analysis. How to effectively use a sentiment lexicon is a common problem that appears in the vast majority of Sentiment Analysis works. An interesting survey that tackles the use of sentiment lexicons for Sentiment Analysis is done in (Ravi et al., 2015b). Numerous sentiment lexicons can be used, and determining which characteristics determine the final performance of a system that uses them is an open challenge. In this sense, the work in (Bravo-Marquez et al., 2014) treats some of these issues, analyzing several sentiment lexicons, and how they can be complemented in a Sentiment Analysis system.

One relevant attribute of sentiment lexicons is the creation methodology, which can be divided into three main categories (Liu and L. Zhang, 2012; Araque, Guerini, et al., 2017): (i) manual approach, (ii) dictionary-based approach, and (iii) corpus-based approach. The manual approach is made by humans, resulting in a very time-consuming process; thus, lexicons generated in such a manner are typically combined with the other two, that make use of automatic strategies. Many dictionary-based methods involve bootstrapping from a small set of opinionated words, using them as seeds to search related words in a known dictionary such as WordNet (Miller, 1995) or SentiWordNet (Baccianella et al., 2010). Traditionally, dictionary-based approaches are not able to obtain domain-specific orientations (Liu and L. Zhang, 2012). Although, in order to generate domain-adapted dictionaries, some recent

works use unsupervised pattern-based approach (Agathangelou et al., 2018), as well as morphosyntactic information (Vulcu et al., 2014). Despite this, corpus-based techniques can be used to alleviate the issue mentioned above. These last type of lexicon generation methods rely on co-occurrence patterns detected on an unsupervised corpus, and a seed set of opinion words to locate opinionated words in the corpus (Yadollahi et al., 2017; Hamilton et al., 2016).

As done in this thesis, sentiment lexicons have been used as features in supervised machine learning scenarios. In (Kouloumpis et al., 2011), lexical resources are complemented with n-gram, Part-of-Speech (PoS), and micro-blogging specific features, such as the presence of emoticons. In this way, the authors show that both lexicon and micro-blogging features are useful in their validation. Similarly, the work in (S. Mohammad, Kiritchenko, and X. Zhu, 2013) makes use of two sentiment lexicons, combining this information with n-grams, PoS. All these features are used by an SVM-based classifier, resulting in a state-of-the-art system in a Sentiment Analysis competition. In some cases, lexicon features have been complemented with domain-specific features, as in (Jiang et al., 2011), where the search query determines the domain. Also, several lexicons can be used at the same time, as in the case of (Zirn et al., 2011). This last work integrates several lexicons using Markov logic with information about relations between neighboring words.

While many sentiment lexica have been produced, fewer linguistic resources for emotion research are described in the literature. One of the first is WordNet-Affect (Strapparava and Valitutti, 2004), a manually-built extension of WordNet. About 1k lemmas are assigned a label taken from a hierarchy of 311 affective labels, including Ekman’s six emotions (Ekman and Friesen, 1971). AffectNet (Cambria, Mazzocco, et al., 2011) is a semantic network containing about 10k items; emotions are represented in a vector space where the dimensions are “pleasantness”, “attention”, “sensitivity”, and “aptitude”. Each of these dimensions can have a value between -3 and +3 for a total of 24 distinct emotions that can be combined.

The Affect database (Neviarouskaya et al., 2010) contains 2.5k lemmas taken from WordNet-Affect and has been manually enriched by adding the strength of association with Izard’s basic emotions (Izard, 1977). EmoLex (S. Mohammad and P. Turney, 2013) is a crowdsourced lexicon containing 14k lemmas, each annotated with binary associations to Plutchik’s eight emotions (Plutchik, 1980).

A recent resource is the NRC Affect Intensity Lexicon (S. Mohammad, 2018), which includes 6k entries manually annotated with joy, fear, anger, and sadness. In a similar line of work, the NRC Hashtag Emotion Lexicon (S. Mohammad and Kiritchenko, 2015) has a coverage of over 16k unigrams. This resource has been automatically inferred

from microblogging messages distantly annotated by emotional hashtags. As such, this lexicon is particularly useful when applied to the Twitter domain.

In (Bandhakavi et al., 2017), a generative unigram mixture model is proposed as a solution for creating domain-specific emotion lexica by jointly modeling “emotionality and neutrality of words”, starting from a corpus of labeled documents.

Apart from theories of basic emotions, the literature has also proposed several non-categorical models, where emotions are represented as points in a vector space. The Valence-Arousal-Dominance (VAD) model, in particular, is worth mentioning since multiple resources are based on it. ANEW (Bradley et al., 1999) is a set of 1k words manually annotated. A crowdsourced extension of it (Warriner et al., 2013) provides fine-grained VAD ratings for 14k frequent-usage words. A recent work, the NRC Valence, Arousal, Dominance Lexicon (S. M. Mohammad, 2018), contains 20k terms manually annotated using Best-Worst Scaling (Kiritchenko et al., 2016). While many of these resources were used for Sentiment Analysis, considering only the valence scores (which roughly corresponds to the “sentiment” or “polarity” of a word), the VAD model actually represents emotions.

A recent work has demonstrated the possibility of seamlessly converting VAD annotations between Ekman’s categorical emotions, and vice versa (Buechel et al., 2018). Recently, a multilingual study using the GRID dictionary extended the VAD model to include a new dimension, Novelty, but no emotion lexicon is using this model yet (Fontaine et al., 2013).

As part of this thesis, the DepecheMood lexicon is extended **“MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”** (Section 3.2.2). The original DepecheMood lexicon (Staiano et al., 2014) and its extension, are the largest among these emotion lexica. They are automatically derived from a corpus of news articles with annotated emotions, and thus they can be assumed to be general-purpose. Their set of emotions is also peculiar: while most of the other lexica are related or easily mapped to Ekman’s, Plutchik’s or Izard’s sets, the categories “annoyed” and “amused” are unique for these resources.

As a summary, Table 1.1 shows a comparison of these emotion lexica. The “Annotation type” column indicates whether a lexicon reports binary values for an emotion, or a number indicating its strength. Similarly, the “Annotation process” column indicates whether the generation process is manual or automatic (e.g., obtained as a result of an algorithm).

In a parallel effort in the field, many works have been oriented to domain adaptation of sentiment lexicons. Given the idiosyncrasy of sentiment traits in a given domain, a necessity for prior domain knowledge arises. In this thesis, we understand *domain-oriented*

Domain	Name	No. entries	Annotation type	Annotation process	Reference
Sentiment	General Inquirer	4k	Categorical	Manual	Stone et al. (1966)
	ANEW	1k	Numerical	Manual	Bradley et al. (1999)
	-	6k	Categorical	Automatic	Hu et al. (2004)
	SentiWordNet	117k	Numerical	Automatic	Baccianella et al. (2010)
	AFINN	2k	Numerical	Manual	Nielsen (2011)
	SO-CAL	4k	Numerical	Manual	Taboada et al. (2011)
	-	14k	Numerical	Manual	Warriner et al. (2013)
	SentiWords	144k	Numerical	Automatic	Gatti et al. (2016)
Emotion	Fuzzy Affect	4k	Numerical	Manual	Subasic et al. (2001)
	NRC VAD	20k	Numerical	Manual	S. M. Mohammad (2018)
	WordNet-Affect	1k	Categorical	Manual	Strapparava and Valitutti (2004)
	Affect database	2k	Numerical	Manual	Neviarouskaya et al. (2010)
	AffectNet	10k	Categorical	Automatic	Cambria, Mazzocco, et al. (2011)
	EmoLex	14k	Categorical	Manual	S. Mohammad and P. Turney (2013)
	NRC Hashtag	16k	Numerical	Automatic	S. Mohammad and Kiritchenko (2015)
	NRC Affect	6k	Numerical	Manual	S. Mohammad (2018)

Table 1.1: Overview of related Sentiment and Emotion lexica.

or *domain-adapted* lexicons as resources that have, in one way or another, specific domain knowledge.

There is a great variety of statistical information extraction methods for the generation of domain-oriented lexicons. Nevertheless, some works share some common methodology, as is the case of *optimization-based* approaches. For instance, the work shown in Du et al. (2010) proposes an information bottleneck problem that blends cross and within-domain knowledge. This problem is then solved with an iterative optimization method that, upon convergence, obtains the desired lexicon. Choi et al. (2009) explores a different approach, where a linear programming technique is aimed to directly reflect the characteristics of a particular domain. This method exploits the relation between words and opinion expressions to compute the most likely polarity value in that domain. Another explored technique is presented in Lu et

al. (2011), in which several opinion signals are extracted from an unlabeled opinionated text collection. These signals are combined into an optimization problem and then transformed into a linear programming formulation by means of a mathematical transformation. In this way, the method can learn the sentiment of words and aspects.

Another common technique used when generating domain-specific lexicons is *label propagation*. In this line of work, the technique explained in Qiu et al. (2009) introduces a method called double propagation. This method consists of several extraction rules that are designed based on dependency trees and exploit the relations between modifier sentiment words and their associated topics, as well as the sentiment and topic words. Similar research (Huang et al., 2014) proposes an automatic construction strategy based on constrained label propagation. The strategy firstly extracts candidate sentiment terms, and then it uses domain and morphological constraints to spread to the entire set of candidates, improving the lexicon. Label propagation can also be applied to different structures, such as a lexical graph. This approach is taken in Hamilton et al. (2016), where a set of high-quality word embeddings are transformed into a lexical graph by connecting each word with its nearest neighbors in the semantic space. Then, the sentiment labels are propagated from a set of seed words through the graph using a random walk algorithm.

Lastly, other methods directly extract statistical information from corpora and transform this knowledge into a domain-adapted sentiment lexicon. The method described in Almatarneh et al. (2017) uses frequency information to assign different weights to positive and negative words in the domain of movie reviews. Another example of such approaches is Demiroz et al. (2012), which uses the *TF-IDF* score for each sentiment polarity to adapt an existing general lexicon to a specific domain. Also, the works described in Staiano et al. (2014) and Song et al. (2016) describe the *Compositional Semantics* methods that, parting from a frequency measure of the words in the corpus and a set of emotion labels (this could be used in sentiment as well), builds a lexicon from a word-emotion matrix.

This thesis addresses lexicons in various ways. We have seen how extended utilization methods are straightforward, but lack in representational power. Following this research line, this thesis has contributed with the work described in **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3). Another challenge lies in lexicon generation. Automatically deriving a lexicon that has both high coverage and quality annotations is a difficult task, and many works address this issue. In this line, this thesis offers two approaches, one that exploits a known neural network algorithm (**“Neural domain adaptation of sentiment lexicons”** (Section A.2.5)), and another that uses a simple and effective statistical method (**“DepecheMood++: a Bilingual Emotion**

Lexicon Built Through Simple Yet Powerful Techniques” (Section 3.2.4)).

1.2.2 Word Embedding Models

Continuous word vector representations contain syntactic and semantic regularities present in the language, expressed as relation offsets in the vector space (Mikolov et al., 2013). Most common approaches train unsupervised word embeddings models without a specific objective, but rather to capture language knowledge. This type of word vectors, which are trained using co-occurrence information, is typically called generic or pre-trained word vectors. In this thesis, pre-trained word vectors are used for all types of proposed feature extraction methods.

In the same way that bag-of-words features are exploited for textual representation in Sentiment Analysis, word embeddings can be similarly used (Mikolov et al., 2013). Some straightforward approaches that use embeddings as features for classification have been studied. The work contained in (Schnabel et al., 2015) studies the effectiveness of using word embeddings applied to several tasks, one of them being Sentiment Analysis. This study offers an overview of unsupervised embedding techniques and how they can obtain meaningful text representations. In the work by D. Zhang et al. (2015), an SVM classifier is trained over embedding representations to predict sentiment polarity. Through their results, the authors argue that embeddings contain deep semantic features between words, which can be positive for Sentiment Analysis. On top of this, word embeddings based techniques are extensively used in public challenges where competitors aim at obtaining the highest scores in the task of Sentiment Analysis (Nakov et al., 2016; Rosenthal et al., 2017). An interesting work that uses embedding with special consideration to the reduction the computational complexity is shown in (Joulin et al., 2016); curiously enough, the authors report a similar accuracy to that of more complex neural models in several tasks, including Sentiment Analysis. For a more detailed description of how to train and use word embeddings, we refer to (Lai et al., 2016).

One relevant limitation of the presented word embedding models is that they ignore the morphology of words, assigning a distinct vector to each word. In light of this, newer models have been proposed that exploit additional sources of information in order to obtain more robust word representations. Bojanowski et al. (2017) propose the *FastText* word embedding model, where each word is represented by a character n -gram. In this way, a vector representation for a specific word is the result of the sum of the vectors that represent the character n -grams that form the said word. This method trains fast and has the advantage that allows computing representations of words that have not been previously

seen in the training data. In a complementary work (Joulin et al., 2016), the authors show how this kind of embedding model can reach high performances in several NLP tasks, including Sentiment Analysis.

A similar work is presented in Peters et al. (2018), introducing the Embeddings from Language Models (ELMo) model. This model uses both complex characteristics of word use (e.g., semantics and syntax) and variations across linguistic contexts. These word representations are learned from the internal states of a specialized neural network, pre-trained on a large text corpus. Such embeddings can be easily added to a learning system, and that they can boost the performance on significant NLP problems, being one of those Sentiment Analysis.

In the line of using embedding-based approaches, many authors have proposed complex neural architectures that are able to exploit word vectors successfully. One architecture family is Recurrent Neural Networks, which can adapt to different sequence sizes; these results especially useful in text processing, as a document can be modeled as a sequence of sentences, words, or even characters. As shown by Socher et al. (2013), the study of compositionality in sentiment classification tasks has proven to be relevant. This work proposes the Recursive Neural Tensor Network (RNTN) model, and it also shows how RNTN outperforms previous models on both fine-grained and binary Sentiment Analysis tasks.

The RNTN model represents a phrase using word vectors composed from the structure given by a parse tree, computing vectors for higher nodes in the tree using a tensor-based composition function. Another interesting work (Dai et al., 2015) tackles the use of Long Short-Term Memory (LSTM) networks, which read the input sequence into a vector. These vectors are used then to predict the sentiment label. A similar approach (Tai et al., 2015) leverages the use of both LSTM and parse trees, outperforming previous systems and strong baselines in sentence relatedness and sentiment classification tasks. As an additional technique, attention mechanisms are normally used in recurrent architectures (Bahdanau et al., 2014), which allows models to search for parts of the input relevant to the problem. Building on top of this, the work in Yang et al. (2016) incorporates knowledge of document structure, developing an attention-based approach under the assumption that not all parts of a document are equally important. In this way, the model is allowed to distinguish the importance of the different parts that compose a document, resulting in benefits in Sentiment Analysis.

Another studied family of neural networks is Convolutional Neural Network (CNN) architectures. CNN models were initially proposed in the field of computer vision (Lecun et al., 1998), although they have been successfully applied to a range of NLP tasks (Collobert et al., 2011). A representative study of the use of CNN models in Sentiment Analysis is

described in Kim (2014).

This work uses pre-trained vectors as well as embeddings that are allowed to be fine-tuned during training, which improves the model. Similarly, the work in Severyn et al. (2015) shows that the parameter initialization technique in the embeddings of a CNN model profoundly affects the final result. Also, this work incorporated noisy annotations from data to further refine the weights of the model, optimizing it to perform better in the task at hand. In Wehrmann et al. (2017), the authors propose a character-based CNN model that can be trained for several languages, without requiring any machine processes, comparing these results with the more traditional word-based encoding.

Taking this into consideration, it seems natural that some authors have studied the applicability of hybrid architectures, with the aim of leveraging both RNN and CNN models. In T. Chen et al. (2017), the proposed model aims at combining these two types of architectures into a unified approach that firstly classifies documents according to the number of opinion targets with a recurrent model and later applies a convolutional network to perform Sentiment Analysis. The work in Duyu Tang et al. (2015) tackles the encoding of semantic relations between sentences in a document by combining, on the one hand, a convolutional network that learns sentence representations, and on the other hand, a recurrent network that encodes the sentence relations.

Following, a new neural architecture has been proposed: the *Transformer* (Vaswani et al., 2017). The Transformer, in the context of NLP, is a relatively novel architecture that aims to solve sequence-to-sequence tasks while handling long-range dependencies. The main characteristic of the Transformer is that it is based solely on attention mechanisms, without any CNN or RNN architectures. Such an architecture allows for significantly more parallelization than the previous state-of-the-art models, which rely heavily on RNN-based architectures. Additionally, the Transformer also improves the accuracy in important NLP tasks, such as machine translation, and Sentiment Analysis (Ambartsoumian et al., 2018).

Based on the advancements that the Transformer brought to the NLP community, Devlin et al. (2018) presented Bidirectional Encoder Representations from Transformers (BERT). This Transformer-based language model trains deep bidirectional representations from unlabeled text by jointly observing both the left and right contexts. The BERT model has revolutionized the field, penetrating into all tasks within NLP. As a relevant example, BERT has been successfully applied to Sentiment Analysis (Sun et al., 2019).

Although the described neural architectures have high representational power, there have been efforts towards adapting embeddings models to the task of Sentiment Analysis. A pioneer work that uses embeddings applied to Sentiment Analysis is that of Maas et al. (2011).

In this work, both semantic and sentiment knowledge is captured in the embeddings. Similarly, the work in D. Tang et al. (2016) proposes a model that generates semantic sentiment embeddings, capturing the context of sentiment together with co-occurrence information. In this way, generated sentiment vectors can be directly used for Sentiment Analysis tasks without feature engineering. Sentiment signals that are incorporated into word embeddings can come from different sources, as in the case of the work in Li et al. (2017), where the model includes sentiment information from both word and document level. An alternative method for generating sentiment embeddings is possible: refining a pre-trained embeddings model (Yu et al., 2017). This work adjusts the original word embeddings, allowing them to be closer to both semantic and sentiment-wise similar words; while at the same time moving them further away to dissimilar words also in the semantic and sentiment plane.

Word embedding models are a key part of this thesis and are used extensively in the following publications: **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5), **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3), **“An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity”** (Section 3.2.1), and **“MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”** (Section 3.2.2).

1.2.3 Semantic Similarity

Corpus-based semantic similarity methods are based on word associations learned from large text collections following the distributional hypothesis (Peter D Turney, Pantel, et al., 2010). Two words are assumed to be similar if their surrounding contexts are similar, or if they appear together more frequently. The computation of corpus-based methods is based on statistics of word distributions through word co-occurrences. According to different computational models, there are count-based methods, e.g., Pointwise Mutual Information (Church et al., 1990) or Normalized Google Distance (Gligorov et al., 2007), and predictive methods, such as Word2Vec (Mikolov et al., 2013). Count-based methods count word co-occurrences and construct a word-word matrix, in which those co-occurrence statistics are directly applied with probabilistic models (Church et al., 1990), matrix factorization (Pennington et al., 2014) and dimension reduction (Levy et al., 2015). Predictive-based methods directly learn dense vectors by predicting a word from its surrounding context. In this thesis, we use the predictive-based word embedding tool Word2Vec (Mikolov et al., 2013) to learn dense vector representation of words, because it has been reported to have good performance in many applications (Baroni et al., 2014). As suggested by the Word2Vec authors (Mikolov et al., 2013), the Continuous Bag of Words (CBOW) model is more computationally efficient and

suitable for larger corpus than the skip-gram model.

Thus, the CBOW model is used to train word vectors in a neural network architecture, which consists of an input layer, a projection layer, and an output layer to predict a word given its surrounding words with a certain context window size. Formally, given a sequence of training words $\{w_1, w_2, \dots, w_T\}$, each word vector is trained to maximize the average log probability:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq k \leq c, k \neq 0} \log p(w_t | w_{t-k}, \dots, w_{t+k}) \quad (1.1)$$

where k is the context window size and $p(w_t | w_{t-k}, \dots, w_{t+k})$ is the hierarchical softmax of the word vectors (Mikolov et al., 2013). Having the trained word vectors (the dimension is predefined empirically and we set 300 in our experiments), word similarity is computed using standard cosine similarity.

Although the training process relies on a neural network-based supervised prediction model, the real training results are the vector representation of words instead of the neural network prediction model. Because of such an idea, the training of word embedding is unsupervised and can be applied in various textual corpus without labeled datasets. Furthermore, due to the simple neural network architecture and the use of hierarchical softmax, Word2Vec is able to address large corpora, and the training is more efficient.

Semantic similarity models are used frequently in NLP, but their application to lexical resources has not been thoroughly studied. Extracting knowledge from a lexicon in a manner that is directly usable by a learning model represents an area that has not been explored. This same idea partly motivates the work described in “**A semantic similarity-based perspective of affect lexicons for sentiment analysis**” (Section 3.2.3), where a novel method for extracting features from a lexicon by means of a similarity model is described.

1.2.4 Affect Analysis of Online Radicalism

The main problems that have been addressed in previous research for the automatic processing of online radicalization are analysis, detection, and prediction of radicalization (Correa et al., 2013; Fernandez et al., 2018).

Online radicalization analysis aims at providing actionable information to improve Law Enforcement Agencies (LEA)’s decision making processes. According to Correa et al. (2013), analysis approaches fall into two main categories: network-based (leaders, communities, and topology characteristics) and content-based (authorship identification, stylometric analysis, website activities, affect analysis and usage).

The works focused on detection can be classified into web mining and text mining approaches (Correa et al., 2013). Web mining works aim at detecting radical online content using techniques such as focused crawling (S. Agarwal et al., 2015b), while text classification techniques aim at developing a binary classification model based on text features (Rowe et al., 2016) combined with other features such as social dynamics (Rowe et al., 2016; Fernandez et al., 2018).

Regarding detection, different types of analysis have been proposed for understanding better online radicalization, including network and content-based analysis. While network analysis takes into account the social interactions, a content-based analysis is focused on analyzing several aspects of radical texts, such as affects, topics, or stylistic features.

Finally, several works have addressed prediction. Ferrara et al. (2016) propose a machine learning framework for detecting extremist supporters, predicting extremist user content adoption (i.e., a retweet of extremist content) and predicting interactions with extremists (i.e., reply direct messages from extremists). The framework considers three types of features: user and activity, timing, and network. S. Agarwal et al. (2015a) presents a survey focused on two problems, the automatic identification of online radicalization (hate promoting content, users and hidden communities) and the prediction of civil unrest related events (protests, riots, public demonstrations). The survey concludes that most works have used spatiotemporal features as discriminatory ones for predicting events. López-Sánchez et al. (2018) propose a system for predicting radicalization risks. They propose to generate alarms based on the radicalization influence of monitored users and the emotional load of the received retweets.

Table 1.2: Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.

Type	Ref.	Data	Method	Affect Model
A	(Abbasi et al., 2007)	Forum posts	Linear regression	Intensity of hate and violence terms
A	(Birmingham et al., 2009)	YouTube Group	Sentiment analysis of most frequent terms	Sentiment polarity using SentiWordNet.

Table 1.2: Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.

A	(Rowe et al., 2016)	pro-ISIS Twitter users	Definition of windows per analyzing the interaction and an algorithm for calculation the adoption of a term based on lexical, sharing and interactions homophily	Sentiment analysis (polarity) using MPQA (Deng et al., 2015) and Ar-SenL (Badaro et al., 2014)
A	(Jurek et al., 2015)	Twitter	Lexicon based Sentiment analysis	Sentiment (polarity) using SentiWordNet (Baccianella et al., 2010)
A	(H. Chen, 2008)	Dark Web Forum postings	Support Vector Regression ensemble	Manual annotation of the intensity (0-1) of violence, anger, hate, racism and sentiment
A	(Vergani et al., 2015)	Dabiq radical magazine	Sentiment analysis based on LIWC (Linguistic Inquiry and Word Count) (Pennebaker et al., 2001)	Anger, anxiety and emotion tone from LIWC
A	(Chalothorn et al., 2013)	Forum posts	Threshold of negative words to calculate sentence polarity	Sentiment (polarity) using SentiWordNet (Baccianella et al., 2010)
A	(Ghajar-Khosravi et al., 2016)	ISIS fan girls tweets and control datasets	Lexicon based Sentiment Analysis toward some topics identified by keywords	Sentiment polarity
A	(Dewan et al., 2017)	Images and text from Facebook posts re- lated to Paris attack	Domain transfer learning with SentiBank Borth et al., 2013 for images and LIWC (Pennebaker et al., 2001) for texts	Sentiment polarity

Table 1.2: Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.

D	(Ashcroft et al., 2015)	pro and anti-ISIS tweets	Sentiment analysis (polarity) based on CoreNLP (Manning et al., 2014) that uses a Recursive Deep Model over a Sentiment Treebank.	Feature based classification using SVM, Naive Bayes and Adaboost
D	(S. Agarwal et al., 2015c)	Twitter	Single-class SVM and KNN	Binary model to indicate if a tweet includes terms that indicate a negative emotion
D	(Saif et al., 2017)	pro and anti-ISIS tweets	Lexicon based Sentiment Analysis	Polarity (sentiment): neutral, positive, negative using SentiStrength (Thelwall et al., 2012)
P	(López-Sánchez et al., 2018)	Twitter	Alerts based on user influence (followers, followed and messaged retweeted) and sentiment load of received mentions	Sentiment (polarity) using SentiStrength (Thelwall et al., 2012)

In this section, we provide a more detailed analysis of the use of affect technologies for analyzing online radicalization. As shown in Table 1.2, affect analysis has been applied in a wide array of domains, such as radical forums (Abbasi et al., 2007; H. Chen, 2008; Chalothorn et al., 2013); radical magazines (Vergani et al., 2015); and social networks, such as Twitter (Rowe et al., 2016; Ghajar-Khosravi et al., 2016; Jurek et al., 2015; Saif et al., 2017; Rowe et al., 2016), Facebook (Dewan et al., 2017) and YouTube (Bermingham et al., 2009).

Regarding the affect model, most works analyze the polarity (i.e., valence or sentiment) using Sentiment Analysis techniques (Bermingham et al., 2009; Jurek et al., 2015; Vergani

et al., 2015; Chalothorn et al., 2013; Ghajar-Khosravi et al., 2016; S. Agarwal et al., 2015c; Saif et al., 2017). Other works evaluate the intensity of some terms, such as hate or violence (Abbasi et al., 2007; Jurek et al., 2015). Finally, other works (H. Chen, 2008; Dewan et al., 2017) use LIWC’s categories for affective processes (Pennebaker et al., 2001): positive emotions, negative emotions, anxiety, anger, and sadness.

The advancements in Sentiment Analysis for radicalization studies have provided many insights. Abbasi et al. (2007) conclude that there is a linear relationship between the intensity scores for violence and hate affects, which is strong in Middle Eastern forums and weak in U.S. forums.

Another interesting aspect is the process of radicalization. Rowe et al. (2016) noticed when analyzing the radicalization process that users, before being activated, frequently discuss politics, using words like ‘Syria’ or ‘Egypt’ with negative sentiment. Once they are activated, they tend to use more religious words, and some words as ISIS are used with negative sentiment since they prefer the term ‘Islamic State’.

Some analyses have also considered the temporal evolution of the text with different purposes, such as the comparison of different radical blogs (H. Chen, 2008) or the language used in Dabiq radical magazine (Vergani et al., 2015).

Finally, another compelling aspect is how public opinion reacts when a terrorist attack has happened. Dewan et al. (2017) analyzed both the sentiment of images and texts posted on Facebook. They observed that the sentiment for textual posts was negative during the first few hours but gradually moved to the positive side over time. Contrarily, posted images reflected positive sentiment initially but moved towards negative after the first few hours.

There is not an agreement in current research regarding the importance of affect signals. Several studies (Chalothorn et al., 2013; Ashcroft et al., 2015; S. Agarwal et al., 2015c) highlight their importance; while in other experiments (Saif et al., 2017) sentiment features, which consist of both word unigrams and their sentiment, have no impact on the classification performance compared to using unigrams only.

We have seen that previous works tackle successfully incorporating sentiment information for radicalization detection. Although the literature indicates that emotions are connected to the radicalization process, computational approaches that insert Emotion Analysis methods are scarce. The work presented in **“An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity”** (Section 3.2.1) effectively exploits emotion information, incorporating it into the radicalization detection process. These affect cues are combined with textual representations based on word-embeddings, achieving

high-performance results.

1.2.5 Moral Foundations Theory

Psychologists and social scientists have systematically analyzed text data to address their research questions. Back in the late '60s, the Harvard General Inquirer dictionary (Stone et al., 1966) was the precursor of Sentiment Analysis which was to become, together with opinion extraction, a core theme of natural language processing (NLP).

Ever since, scientists gradually increased complexity moving from simple techniques (e.g., the unsupervised and partially rule-based approach of Peter D Turney (2002)) to sophisticated methods that try to determine the context of words (e.g., Muhammad et al. (2016)). The topics addressed also became more challenging tackling notions such as irony and sarcasm (Sulis et al., 2016). With time, not only the methods became more sophisticated, but also the tasks became more ambitious. Extensive studies on linguistic markers of sentiment and affect (Strapparava and Mihalcea, 2008; Liu, 2015; Muhammad et al., 2016; Poria, Gelbukh, et al., 2014) paved the way to assess more complex constructs such as personality (Schwartz et al., 2013; Yarkoni, 2010) and human values (Boyd et al., 2015; J. Chen et al., 2014). Moral values are considered to be a higher-level construct with respect to personality traits, determining how and when dispositions and attitudes relate with our life stories and narratives (McAdams et al., 2006). Here we provide a brief literature review of the studies regarding moral values assessment from textual data. As in sentiment and personality analysis, also, in this case, pioneer works followed a dictionary-based approach, while the current state-of-the-art performance is based on deep learning.

The first vocabulary developed to assess the moral values from textual data was the Moral Foundations Dictionary (MFD) (Graham et al., 2009). It was used together with the Linguistic Inquiry and Word Count (LIWC) software (Tausczik et al., 2010) to estimate moral traits and to investigate differences in moral concerns between different cultural groups. Clifford et al. (2013) employed the MFD for performing manual text analysis of 12 years of coverage in the New York Times, focusing on political debate in the US. Teernstra et al. (2016) assessed the political debate regarding the “Grexit” from approximately 8,000 tweets. They compared the performance of using the raw data, bi-grams, and the MFD features in employing basic machine learning models, namely, Naive Bayes (NB) and Maximum Entropy (ME). They concluded that pure machine learning is preferable to dictionary approaches since it has similar prediction accuracy while using fewer assumptions. In this thesis, we follow a similar approach to Teernstra et al. (2016); however, we propose an expanded version of the MFD, including also the moral valence per lemma. Moreover,

we employ logistic regression models to infer moral values from uni-grams combined with lexical features.

Dehghani et al. (2014) examined the differences between liberal and conservative moral value systems using a hierarchical generative topic modeling technique, based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003), to enable the unsupervised detection of topics in their corpus of liberal and conservative weblogs. They used small sets of words selected from the MFD as seeds to encourage the emergence of issues related to different moral concerns. They examined similarities and differences in how such matters are expressed between these groups. Consistently with findings in moral psychology, they demonstrate that there are significant differences in how liberals and conservatives construct their moral belief systems. Sagi et al. (2014a) employed the same framework to study moral rhetoric in text for a specific case study, the US Federal shutdown of 2013 (Sagi et al., 2014b), where they examined the role of morals in intra- and inter-community differences of political party retweets. In both works, they were based on the framework presented in Dehghani et al. (2014), where LDA was employed to create a co-occurrence matrix on which the similarity between the texts and the vectors representing the different MFT moral traits was computed. In a similar approach, Kaur et al. (2016) attempted to quantify the moral loadings of text, based on the Latent Semantic Analysis (LSA). They used a bag-of-words model, representing the entire corpus by a word-context matrix. Then they reduced its dimensionality obtaining low-dimensional word vectors, in which similar vectors represent similar meaning words. Our study is presenting a different approach since we do not use LSA representations, but rather pre-trained word embeddings models. Although pre-trained word embeddings do not contain domain-specific knowledge, they express language regularities encoded as offsets in the resulting vector space. The proposed representations based on **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3) precisely exploit the similarity between the analyzed text and a selection of words with moral content.

More recently, Garten, Boghrati, et al. (2016) employed the MFD to detect moral rhetoric in general, and more specifically, shifts in long political speeches over time. Then, based on psychological dictionaries and semantic similarity to quantify the presence of moral sentiment around a given topic, Garten, Joe Hoover, et al. (2018), proposed the Distributed Dictionary Representations (DDR) method. Showing promising results, DDR was also employed by Joe Hoover et al. (2018) to detect moral values in charitable giving. Later on, Garten, Kennedy, et al. (2019) extended the method, incorporating demographic embeddings into the language representations. Our approach is based on an expanded version of the MFD, with evaluated manual annotations regarding the moral valence of each lemma, that can be incorporated in computational frameworks.

In a study more similar to the one performed in this thesis, attempting to predict moral values involved in Twitter posts, Lin et al. (2018) proposed a method that automatically acquires background knowledge to improve the moral value prediction, pointing out the difficulty of the task also for human experts. Based on the work of Lin et al. (2018) and Graham et al. (2009), Mooijman et al. (2018) predicted the moral sentiment of the tweets. Their model consists of three layers, an embedding (lookup) layer, a recurrent neural network with long short-term memory (Hochreiter et al., 1997), and an output layer. The first layer converts words in an input tweet to a sequence of pre-trained word embeddings. The LSTM layer processes these embeddings and outputs a fixed-sized vector that encodes critical information for moral value prediction. A vector representing the percentage of words that match each category in the Moral Foundations Dictionary (Graham et al., 2009) are concatenated with the LSTM feature vector. Our approach is differentiating to this one since we employ word embeddings to compute the similarity between words rather than directly feeding them in a neural network architecture. Along the same line, Rezapour et al. (2019) investigated the relationship between moral values and stance towards a series of social issues. Their findings show that enhancing the original MFD improves the prediction accuracy of morality in text. They expanded the original MFD and employed a series of machine learning classifiers (SVM, RF, and LSTM) predicting the moral traits. This study underlines the importance of expanding the MFD; we go one step further, introducing the notion of moral “strength” while showing how more abundant information is improving the overall accuracy of the models.

Computational approaches to moral values are highly scarce, and one of the few computational resources is the MFD, which is limited in coverage. A higher-quality lexicon for this domain can benefit research, and thus a contribution of the work presented in **“Moral-Strength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”** (Section 3.2.2) is the extended lexicon of moral lemmas with the respective moral valence. To explore the properties and full potentials of the lexicon, we suggest three different models of increasing complexity, demonstrating the value of this resource. The proposed approaches range from feature engineering methods to a system which employs word embeddings of semantic similarity based on the work presented in **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3).

1.3 Hypotheses

In light of the described context and background of this thesis, we have formulated the following hypotheses, that guide the course of this work:

- **Hypothesis 1 (H_1)**. The combination of deep and surface textual features, including affect lexicons, enhances final Sentiment Analysis performance.
- **Hypothesis 2 (H_2)**. A method that exploits both distributional representations and affect lexicons through semantic similarity can be leveraged for Sentiment Analysis effectively.
- **Hypothesis 3 (H_3)**. A model implementing a semantic similarity-based exploitation of a lexicon for Sentiment Analysis can be transferred to other Natural Language Processing areas that show domain-specific traits (e.g., radicalization detection, moral value estimation).

1.4 Objectives

With the objective of testing the validity of the previous hypotheses, we propose the following objectives, which are oriented to satisfy the motivation of this thesis:

- **Objective 1 (O_1)**. Definition of a framework for the combination of surface and deep features, as well as the evaluation methodology.

Such a framework should cover different feature combination methods. Besides, a reference evaluation methodology that uses known datasets should be defined.

- **Objective 2 (O_2)**. Definition of a lexicon generation methodology based on the fundamental properties of neural networks and an evaluation procedure for affect lexicons.

The lexicon generation method should take advantage of the backpropagation algorithm, allowing affect annotations to be automatically generated.

- **Objective 3 (O_3)**. Definition of a model for the exploitation of semantic similarity in distributed representations.

The model should exploit the knowledge contained in both word embedding models and affect lexicons, offering a new perspective on how to extract features from text.

- **Objective 4 (O_4)**. Transference of a similarity-based model on lexicons to different Natural Language Processing domains, evaluating how specific domain knowledge can be inserted.

The domain adaptation method should be independent of the specific domain.

1.5 Document outline

The rest of the document is structured as follows. Chapter 2 describes the research methodology used to conduct and evaluate the results of this thesis, the phases followed, and guidelines on research methods. In Chapter 3, the full text of the publications that compose the thesis compendium is shown. Each paper is presented in a separate section, with a detailed table, and the full text of the paper as originally published. Chapter 4 presents an overview of the contributions made in this thesis. Following, the obtained results are detailed and related to the objectives to which they contribute. After that, the conclusions obtained during the development of this thesis are discussed to finish with some future work lines. Lastly, Appendix A contains a full list of the publications of this thesis.

Methodology

The scientific method defines a number of principles widely known, yet extremely difficult to apply correctly. Still, all researchers around the globe make daily efforts to adhere to these principles. This chapter presents the experimental methodology followed in this thesis.

“If we knew what we were doing, it would not be research”. Anonymous.

2.1 Introduction

Researchers use experiments to answer questions. An experiment can be characterized by the treatments and experimental units to be used, and the responses that are measured (Oehlert, 2010). The treatments are applied to the experimental units, culminating with the measure of one or more responses. Experiments help researchers answer questions, avoiding biases and errors that may appear in other mechanisms. Experiments allow for a direct comparison between different experimental units or treatments. More importantly, we are in control of experiments, which allows us to make more robust inferences about the nature of differences that can be seen in an experiment.

Treatments, units, and the method specify the experimental design. Although the analysis phase is not an intrinsic part of the experimental design, a wise experimenter will take into account the analysis when planning an experiment. In a general tone, a successful experimental design should avoid systematic errors, be precise and objective, allow for an estimation of error, and have broad validity. In practice, fulfilling all these requirements is not feasible, which may take an experimenter to adopt a compromise. For example, exploring a wide range of hyperparameters of a learning model can result in excessively long computation times. Instead, a reduced subset can be used in order to approximate to a suboptimal response.

This thesis is an experimental one. Experiments lie at the core of the work presented in this document. Through experimenting, a set of hypotheses has been heavily tested, which has allowed us to extract a series of interesting conclusions. Such a methodology follows, of course, the scientific method.

Given the domain the thesis addresses, many specific tools, methods, measures, and good practices have been needed. As an example, in the machine learning field, a number of conventions and methodologies are usually used, including this thesis. Also, and thanks to the advance of software engineering applied to both machine learning and Natural Language Processing, the experimental design, execution, and replication are easily done, allowing for an iterative workflow that facilitates the application of the scientific method.

We would like to highlight the importance of this last characteristic of experimentation: replication. Currently, many scientific fields find themselves in a replication crisis, aggravated by the increase in the overall number of yearly scientific publications. This thesis tries to overcome these limitations in such a context, outputting several software artifacts that allow for the easy replication of results, methods, and models. We believe that modern scientific works should consider the aspects mentioned above, and more importantly, in a field as

machine learning, which is intimately connected to software engineering.

The following sections describe the experimental methodology used during the development of this thesis. These sections stress requirements, recommendations, and good practices in the umbrella of machine learning experimentation.

2.2 Machine Learning Experimentation

In an effort to systematize the methodology in this thesis, we have followed a known proposal for machine learning-driven research (Alpaydin, 2014). This guideline defines a series of steps that, in reality, are the same for any type of experimentation (Montgomery, 2005). The overall discussion revolves around a fixed method for approaching a machine learning experiment, independently of the domain, the learning variant, and even the data used. As in this thesis, we have tackled several objectives and domains. This methodology has been applied repeatedly, varying and adapting for each case. In this way, we follow a series of steps:

1. **Definition of the aim of the study.** This first step aims to define the scope of the study, assessing what is the overall questions that, with our experiments, we try to answer. In some cases, we may be interested in comparing two or more learning approaches to a specific problem, or learning whether a particular approach is valid for a domain where it has not been studied before. In general, we usually have two or more learning systems, and using the data available, we want to order them in terms of error.
2. **Selection of the quality measure.** In this phase, a decision about the quality measure must be taken. While in other machine learning setting, a standard measure in the error rate is used. In sentiment and emotion analysis, a more frequent measure in the accuracy, precision, and recall. However, as the data user's complexity rises, a more informative measure is the F1-score, which summarizes the precision and recall into a single metric. This distinction is also helpful at the design phase since attending to a single metric facilitates the process of making decisions, rather than using several measures.
3. **Choice of factors.** Identifying the factors of the machine learning system depends on the aim of the study. For example, if the objective is to fix an algorithm and finding the best hyperparameters, then those are the factors. In this thesis, the objective is not usually to find the best hyperparameters for the designed algorithms, but rather comparing different learning approaches applied to a problem. In these cases, the

factors would be the different approaches themselves. Still, there are studies where some hyperparameter tuning is done. In those, we have to consider said parameters as factors also.

4. **Choice of experimental design.** In this step, some decisions must be made to divide the available data for train, development, and test. How this division is done is important, since using small datasets leads to high variance responses, with non-significant differences among them. Also, it is relevant to keep replication data to a minimum, as in over-sampling. Following recommendations in the literature, a train/development/test schema has been followed when the datasets allow it. When we have needed to resample the data, over-sampling has been avoided or at least compared against the alternative under-sampling, which does not generate “artificial” samples.
5. **Performing the experiment.** In this phase, many general software engineering recommendations are applicable. It is important to consider experiment replication for both future experiments and knowledge transference. For example, seeding the used random number generator and saving said seed are always good practices. One also needs to consider the effects of software aging. In fact, during this thesis’s development, much external software has been drastically changed, and previous data, results, and code need to be correctly adapted when re-used. Also, it is relevant to mention that comparisons should be unbiased. When comparing other methods to the one proposed in this thesis, all approaches should be investigated equally diligently.
6. **Statistical analysis of the data.** In this step, we need to analyze the obtained results so that whatever conclusion we get is not subjective or due to chance. In this way, we cast a hypothesis that must be answered by a series of statistical methods that receive as input the conditions of the experiments, as well as the results obtained. A significant example of this thesis is the Friedman text (Demšar, 2006), which is explained in more detail in Section 2.4.
7. **Conclusions.** Once all data is collected and analyzed, we are in the position of drawing objective conclusions. Although this constitutes the final step, a frequent conclusion is a need for further experimentation. Consistently with the literature, in this thesis, most of the designs and experiments are iterative. Thus, we establish a feedback link between the design phases and the experimentation analysis, where the last one provides feedback for the tuning of the first.

2.3 Evaluation

Section 2.2 describes, in a general manner, the phases that we follow in this thesis for the development, experimentation, and evaluation of machine learning systems. Such phases are applied widely in the community, and can even be used in experiments that fall outside the machine learning domain. In this section, we describe standard metrics that are used for evaluating a machine learning system. This serves as a reference for the contents of the rest of this document, as these metrics have been used extensively during this thesis.

To evaluate the prediction of a machine learning model, authors usually compare the available labels with the model output, considering the following matrix:

	Predicted class	
True Class	Positive	Negative
Positive	tp : true positive	fn : false negative
Negative	fp : false positive	tn : true negative
Total	p'	n'

Table 2.1: Confusion matrix for a two-class scenario.

These four cases comprehend all possible cases of wrong and right answers of a given machine learning model, which are:

- *True positives* (tp) are the documents correctly classified as positive.
- *True Negatives* (tn) are the negative documents that have been correctly classified as negative.
- *Type I errors* (or false positives, fp) are negative documents that have been incorrectly classified as positives.
- *Type II errors* (or false negatives, fn) are positive documents that have been incorrectly classified as negative.

Once this is defined, computing the usual evaluation metrics is straightforward. The first metric to consider is the *accuracy*, which encodes the fraction of correct predictions of a machine learning system:

$$\text{Accuracy} = \frac{tp + tn}{N} \quad (2.1)$$

Although accuracy is a widely known metric, it does not serve as a reliable measure when using skewed datasets. This is due to the fact that, if the distribution among classes is not even, the accuracy metric can result high, which could take the evaluation to draw wrong conclusions. Consequently, alternative measures can be used to consider uneven data classes.

In this context, the *precision* can be defined as the number of correct positive results divided by the number of all positive results.

$$\text{Precision} = \frac{tp}{tp + fp} \quad (2.2)$$

Similarly, the *recall* corresponds to the number of correct positive results divided by the number of positive results that should have been predicted.

$$\text{Recall} = \frac{tp}{tp + fn} \quad (2.3)$$

The *F1-score* can be interpreted as a weighted average of these two values, reaching its best score at 1, and worst at 0. The expression for the F1-score is:

$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

being P and R the precision and recall, respectively. In this way, the F-Score combines the precision and recall values into a single value, giving information on the performance of a classification system in a numerical piece of information.

2.4 Statistical Analysis

Once the performance metrics have been computed, some conclusions could be drawn. Nevertheless, performing a statistical analysis is a good practice, and should be done when possible (Alpaydin, 2014), avoiding subjective observations of the results. In this sense, during the development of this thesis, we have employed a number of statistical tools, with the aim of analyzing the evaluation results.

Among the used methods, it is worth to highlight the Friedman’s test (Demšar, 2006), which implements a comparison among different learning models based on their performance results on a number of datasets. Friedman’s test initially computes each learning model’s rank in each dataset, taking into account that the best performing method is assigned the lower ranking. An average of their ranks resolves ties. In this way, r_j^i is the rank of the j -th

of the k algorithms on the i -th of a total of N datasets. Friedman's test uses the comparison of average ranks:

$$R_j = \frac{1}{N} \sum_i r_j^i \quad (2.5)$$

With this average, the test states that the null-hypothesis is that all the algorithms are equal, so their ranks R_j are equal. Considering this, the Friedman statistic, with $k - 1$ degrees of freedom can be computed as:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right) \quad (2.6)$$

Nonetheless, it is shown that there is a more useful statistic that is distributed according to the F-distribution, and has $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom Demšar, 2006. This statistic is referred to as the Friedman F, and is expressed as:

$$F_F = \frac{(N - 1)\chi_F^2}{N(k - 1) - \chi_F^2} \quad (2.7)$$

If the null-hypothesis of the Friedman test is rejected, post-hoc tests can be conducted, complementing the statistical analysis. In this thesis, we have used the Bonferroni-Dunn test, which allows us to compare the results of several learning models to a baseline. This post-hoc test can be computed by comparing the critical difference (CD) with a series of pre-computed critical values (Demšar, 2006). The CD is computed as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \quad (2.8)$$

Considering this, in order to assess whether a particular model improves over a baseline, the difference in average ranking is computed and compared against the CD. If the difference between the two models is greater than the CD, the null-hypothesis is rejected, and it can be said that the model improves over the baseline with a statistical significance.

Publications

This chapter presents the central papers published as part of this Ph.D. thesis. As required, not all the publications of this thesis are included in this chapter. A full summary of publications, including those not directly related to the objectives of this thesis, is included in Appendix A.

3.1 Introduction

The following tables describe the main publications of this Ph.D. thesis. Since this thesis is presented as a compendium of publications, this section includes publications that fulfill the requirements. For a full list of the publications that are related to the thesis, please see Appendix A. The following color scheme has been used: gray for conference papers, and

blue for journal papers.

Table 3.1: Publications that are directly related to this thesis.

Page	Title	Year	Venue	Ranking
38	An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity	2020	IEEE Access	JCR 2019 Q1 (3.745)
55	MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction	2019	Knowledge-Based Systems	JCR 2019 Q1 (5.921)
67	A Semantic Similarity-based Perspective of Affect Lexicons for Sentiment Analysis	2019	Knowledge-Based Systems	JJCR 2019 Q1 (5.921)
82	DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques	2019	IEEE Transactions on Affective Computing	JCR 2019 Q1 (7.512)
95	Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications	2017	Expert Systems with Applications	JCR 2019 Q1 (5.452)

3.2 List of central publications

3.2.1 An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity

Title	An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity
Authors	Araque, Oscar and Iglesias, Carlos A.
Journal	IEEE Access
Impact factor	JCR 2019 3.745
Scopus citations	0 (02/09/2020)
Web of Science citations	0 (02/09/2020)
Google Scholar citations	0 (02/09/2020)
ISSN	2169-3536
Volume	8
Year	2020
Keywords	Affective computing, embedding-based semantic similarity model, emotion lexicon, emotion recognition, emotion signals, feature extraction, Internet, machine learning, modern terrorist groups, natural language processing, propaganda messages, radical detection domain, radical users, radicalism, radicalization detection, recruitment, recruitment purposes, social media, social networking (online), social networks, social signs, terrorism
Pages	17877–17891
DOI	https://doi.org/10.1109/ACCESS.2020.2967219
Online	https://ieeexplore.ieee.org/document/8962050
Abstract	The Internet has become an important tool for modern terrorist groups as a means of spreading their propaganda messages and recruitment purposes. Previous studies have shown that the analysis of social signs can help in the analysis, detection, and prediction of radical users. In this work, we focus on the analysis of affect signs in social media and social networks, which has not been yet previously addressed. The article contributions are: (i) a novel dataset to be used in radicalization detection works, (ii) a method for utilizing an emotion lexicon for radicalization detection, and (iii) an application to the radical detection domain of an embedding-based semantic similarity model. Results show that emotion can be a reliable indicator of radicalization, as well as that the proposed feature extraction methods can yield high-performance scores.

Received December 3, 2019, accepted December 27, 2019, date of publication January 17, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967219

An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity

OSCAR ARAQUE¹ AND CARLOS A. IGLESIAS¹

Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: Oscar Araque (o.araque@upm.es)

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant agreement no. 740934.

ABSTRACT The Internet has become an important tool for modern terrorist groups as a means of spreading their propaganda messages and recruitment purposes. Previous studies have shown that the analysis of social signs can help in the analysis, detection, and prediction of radical users. In this work, we focus on the analysis of affect signs in social media and social networks, which has not been yet previously addressed. The article contributions are: (i) a novel dataset to be used in radicalization detection works, (ii) a method for utilizing an emotion lexicon for radicalization detection, and (iii) an application to the radical detection domain of an embedding-based semantic similarity model. Results show that emotion can be a reliable indicator of radicalization, as well as that the proposed feature extraction methods can yield high-performance scores.

INDEX TERMS Affective computing, machine learning, natural language processing, terrorism, radicalism.

I. INTRODUCTION

Modern terrorist organizations are continuously changing and adapting to achieve their objectives as well as to overcome current counter-terrorist strategies. Terrorists have effectively used the Internet in their propaganda and recruitment strategy [1]. As a consequence, online radicalization detection is a significant concern. Digital traces on social networks can be used as social signs to detect online radicalism, based on text mining and social analysis techniques [1], which analyze different aspects of these interactions such as sentiment, topics, and the social context of users and semantics [2].

Recent works in social science [3], [4] have highlighted the role of emotions for understanding terrorism. In [3], van Stekelenburg argues that the radicalization process can be conceived as an emotional transformation consisting of three phases. In the first phase, the group perceives injustice from an out-group fueled by anger. In the second phase, the group adopts a moral superiority based on the emotion of contempt. Finally, in the third phase, the group decides to eliminate the out-group based on the emotion of disgust. Rice [4] proposes a research agenda in social science to understand better the role of emotions in the etiology of terrorism since this understanding can heighten the explanatory and predictive

capacity of the theories. Nevertheless, to our knowledge, previous works in the development of computational models for online radicalization detection and prediction have used only sentiment analysis. This work aims to bridge this gap by researching on the relevance of the emotions expressed in online radicalization publications.

Undoubtedly, there is a trend of studying these issues employing data exploitation techniques, such as machine learning. It is still an open challenge how such technology-driven approaches can be effectively used in the field of terrorism policing [5].

In the current work, we focus on investigating three main research questions (RQ):

RQ1: Can emotion information be used for radicalization detection? If so, how? It has been observed that emotions are related to hostility toward another group, which can evoke radical behavior [6]. Previous work has used sentiment signals, but exploiting emotion information for this task has not been studied thoroughly. Thus, this paper offers a view of how emotion features can be generated and used in a machine learning system for radicalization detection.

RQ2: Can semantic similarity-based features be used effectively for radicalization detection? In learning-based approaches, it is necessary to extract useful features from the input in order to allow the learning model to differentiate between data samples accurately. Currently, it is not clear

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li¹.

which kind of features to extract when performing radicalization detection, and a thorough study in this sense has not been made. Besides, semantic similarity has been successfully used in other Natural Language Processing (NLP) tasks and domains, such as Sentiment Analysis [7] and Entity Disambiguation [8]. In this way, this work explores the use of such a measure for radicalization detection.

RQ3: How can radical vocabularies be obtained and exploited? It is clear that radical content makes use of specific vocabulary to express such ideas. So, it is natural to think that integrating this kind of domain knowledge into the model can enhance its performance. Still, how to use and generate such vocabulary selections is an open challenge that has yet to be deeply studied.

In light of these questions, this paper proposes a machine learning method that aims to detect radical text in two domains: online press and the Twitter social network. For this end, the proposed approach generates distributed representations of the text that are fed into a machine learning classifier. These representations are generated through computing the similarity between the analyzed text and a particular lexicon. The similarity measure is obtained through a pre-trained word embeddings model. In this way, we intend to exploit the knowledge contained in word embedding models as well as in a lexical resource. Additionally, this paper also proposes a novel approach that makes use of an emotion dictionary to compute a statistical summary of the emotions present in the analyzed text. In this way, we aim to exploit the idea existing in the literature that emotion can have a role in the elicitation of radicalism [6].

To extend the current scope of this and future research works, we present a new dataset in this work. Such data has been collected from both radical and neutral online press sources that target ISIS-related topics. The original domain differs from the data usually used in previous works, which may allow for more extensive studies of radicalization in texts.

The rest of the paper is organized as follows. Section II presents an overview of the previous work regarding radicalization detection. Following, Section III presents the collected dataset. In Section IV, the proposed radicalization detection model is described. Later, in Section V, we describe the experimental setup which is aimed at evaluating the proposed model, and present the obtained results and their discussion. Section VI presents a comparison of our proposed model to previous works. Finally, the paper concludes in Section VII by depicting the conclusions drawn from the evaluation, as well as offering an outline of possible future lines of work.

II. RELATED WORK

In this section, we review approaches that are related to our proposal. First, Sect. II-A carries out a literature review that covers the works dealing with affect analysis in online radicalization sources. Then, Sect. II-B introduces related

approaches for exploiting affect signs and word embeddings in a text.

A. AFFECT ANALYSIS OF ONLINE RADICALISM

The main problems that have been addressed in previous research for the automatic processing of online radicalization are analysis, detection, and prediction of radicalization [27], [28].

Online radicalization analysis aims at providing actionable information to improve Law Enforcement Agencies (LEA)'s decision making processes. According to Correa and Sureka [27], analysis approaches fall in two main categories: network-based (leaders, communities and topology characteristics) and content-based (authorship identification, stylometric analysis, website activities, affect analysis and usage).

The works focused on detection can be classified into web mining and text mining approaches [27]. Web mining works aim at detecting radical online content using techniques such as focused crawling [29], while text classification techniques aim at developing a binary classification model based on text features [11] combined with other features such as social dynamics [11], [28].

Regarding detection, different types of analysis have been proposed for understanding better online radicalization, including network and content-based analysis. While network analysis takes into account the social interactions, a content-based analysis is focused on analyzing several aspects of radical texts, such as affects, topics or stylistic features.

Finally, several works have addressed prediction. Ferrara *et al.* [30] propose a machine learning framework for detecting extremist supporters, predicting extremist user content adoption (i.e., a retweet of extremist content) and predicting interactions with extremists (i.e., reply direct messages from extremists). The framework considers three types of features: user and activity, timing and network. Agarwal and Sureka [31] presents a survey focused on two problems, the automatic identification of online radicalization (hate promoting content, users and hidden communities) and the prediction of civil unrest related events (protests, riots, public demonstrations). The survey concludes that most works have used spatiotemporal features as discriminatory ones for predicting events. López-Sánchez *et al.* [26] propose a system for predicting radicalization risks. They propose to generate alarms based on the radicalization influence of monitored users and the emotional load of the received retweets.

In this section, we provide a more detailed analysis of the use of affect technologies for analyzing online radicalization. As shown in Table 1, affect analysis has been applied in a wide array of domains, such as radical forums [9], [15], [18], radical magazines [16], and social networks (Twitter [2], [11], [11], [14], [19], Facebook [20] and YouTube [1]).

Regarding the affect model, most works analyze the polarity (i.e., valence or sentiment) using sentiment analysis techniques [1], [2], [14], [16], [18], [19], [24]. Other works

TABLE 1. Summary of literature survey regarding the use of affective technologies for processing online radicalization signals.

Reference	Type	Data	Method	Affect Model
[9]	Analysis	Forum posts	Linear regression	Intensity of hate and violence terms
[1]	Analysis	YouTube Group	Sentiment analysis of most frequent terms	Sentiment (polarity) using SentiWord-Net [10]
[11]	Analysis	pro-ISIS Twitter users	Definition of windows per analyzing the interaction and an algorithm for calculation the adoption of a term based on lexical, sharing and interactions homophily	Sentiment analysis (polarity) using MPQA [12] and ArSenL [13]
[14]	Analysis	Twitter	Lexicon based Sentiment analysis	Sentiment (polarity) using SentiWord-Net [10]
[15]	Analysis	Dark Web Forum postings	Support Vector Regression ensemble	Manual annotation of the intensity (0-1) of violence, anger, hate, racism and sentiment
[16]	Analysis	Dabiq radical magazine	Sentiment analysis based on LIWC (Linguistic Inquiry and Word Count) [17]	Anger, anxiety and emotion tone from LIWC
[18]	Analysis	Forum posts	Threshold of negative words to calculate sentence polarity	Sentiment (polarity) using SentiWord-Net [10]
[19]	Analysis	ISIS fan girls tweets and control datasets	Lexicon based Sentiment Analysis toward some topics identified by key-words	Sentiment (polarity)
[20]	Analysis	Images and text from Facebook posts related to Paris attack	Domain transfer learning with SentiBank [21] for images and LIWC [17] for texts	Sentiment (polarity)
[22]	Detection	pro and anti-ISIS tweets	Sentiment analysis (polarity) based on CoreNLP [23] that uses a Recursive Deep Model over a Sentiment Tree-bank	Feature based classification using SVM, Naive Bayes and Adaboost
[24]	Detection	Twitter	Single-class SVM and KNN	Binary model to indicate if a tweet includes terms that indicate a negative emotion
[2]	Detection	pro and anti-ISIS tweets	Lexicon based Sentiment Analysis	Polarity (sentiment): neutral, positive, negative using SentiStrength [25]
[26]	Prediction	Twitter	Alerts based on user influence (followers, followed and messaged retweeted) and sentiment load of received mentions	Sentiment (polarity) using SentiStrength [25]

evaluate the intensity of some terms, such as hate or violence [9], [14]. Finally, other works [15], [20] use LIWC's categories for affective processes [17]: positive emotions, negative emotions, anxiety, anger, and sadness.

The analysis of sentiment has provided many insights. Abbasi and Chen [9] conclude that there is a linear relationship between the intensity scores for violence and hate affects, which is strong in Middle Eastern forums and weak in U.S. forums.

Another interesting aspect is the process of radicalization. Rowe and Saif [11] notice when analyzing the process of radicalization that users, before being activated, frequently discuss politics, using words like 'Syria' or 'Egypt' with negative sentiment. Once they are activated, they tend to use more religious words and some words as ISIS are used with negative sentiment since they prefer the term 'Islamic State'.

Some analyses that have also considered the temporal evolution of the text with different purposes, such as the comparison of different radical blogs [15] or the language used in Dabiq radical magazine [16].

Finally, another compelling aspect is how public opinion reacts when a terrorist attack has happened. Dewan *et al.* [20] analyzed both the sentiment of images and texts posted on Facebook. They observed that the sentiment for textual posts was negative during the first few hours but gradually moved

to the positive side over time. Contrarily, posted images reflected positive sentiment initially but moved towards negative after the first few hours.

There is not an agreement in current research regarding the importance of affect signals. Several studies [18], [22], [24] highlight their importance, while in other experiments [2] sentiment features, which consist of both, word unigrams and their sentiment have no impact on the classification performance compared to using unigrams only.

B. WORD EMBEDDINGS AND AFFECT LEXICONS

Word embedding representations encode both semantic and syntactic regularities that are extracted through training from large amounts of not annotated text [32]; such regularities are expressed in the resulting vector space as relation offsets. Commonly, word embedding approaches are trained using an unsupervised training process where word vectors are generated, forming word representations that capture relevant language knowledge. Such type of word vectors, whose training process relies on co-occurrence data, are denoted pre-trained word vectors. This work makes use of this type of word vectors for one of the proposed feature extraction method.

As traditionally done with bag-of-words features, word embeddings can be used as features for textual representation

in text categorization tasks, including sentiment analysis [32]. Several works tackle the use of word embeddings as feature extractors in text classification. The work described in [33] constitutes a study of the effectiveness of word embeddings when applied to various tasks, including sentiment analysis. This study also provides a summary of unsupervised embedding models and how they can be used to acquire text representations. In the work of [34], word embeddings are used as features for a SVM classifier in the task of polarity detection. After the evaluation, the authors conclude that word embeddings may contain semantic information among words, which can be leveraged for obtaining useful text representations. In addition, word embeddings can be used in combination with more traditional features, such as n-grams, lexicons and lexical features to improve the quality of text representations [35]. Authors also show that the use of embeddings in an ensemble model can produce higher prediction performance. Also, it is worth to consider that techniques based on word embeddings are widely used in public challenges in which competitors aspire to obtain the highest scores in a wide variety of tasks [36]–[38]. Reference [39] introduces an interesting work where word embeddings are used in a relevant analysis of computation complexity reduction. In a compelling experiment, the authors report similar performance metrics in comparison to more complex neural models in several tasks. Furthermore, [40] presents a model that uses both a measure of semantic similarity and embedding representations. For a comprehensive description of how to exploit word embeddings, we refer the reader to [41].

Considering affect lexicons, the challenge of adequately using an affect lexicon is a common issue that frequently appears in Sentiment Analysis literature. The interested reader can consult an interesting survey regarding the use of affect lexicons [42]. There are numerous affect lexicons available, and deciding which qualities drive the final performance of a machine learning system that uses them is currently an open challenge. In this line of work, [43] deals with some of these topics, analyzing a series of affect lexicons and how they can be complemented.

As done in this work, affect lexicons have been used as features in supervised machine learning scenarios. For example, in [44], lexical resources are combined with both specific micro-blogging features and n-grams. In such a way, the authors show that these features are useful in their validation. Similarly, the work described in [45] uses two sentiment lexicons and then combines such information with n-gram and Part-Of-Speech (POS) features. All of these features are used by a SVM classifier, which resulted in a state-of-the-art system in a Sentiment Analysis competition. There are cases where lexicon features have been supplemented with domain-specific features, as done in [46], where a search query settles the domain. Of course, several affect lexicons can also be aggregated, as is the case of [47]. That work integrates several lexicons by means of Markov's logic, using information about the relations between neighboring words.

III. MAGAZINES DATASET COLLECTION

The scarcity of annotated data in the radicalization detection field is notable, which difficult the advancement of research. As an additional issue, we have to consider that the majority of existing datasets are oriented to the Twitter domain [28]. Indeed, Twitter represents a valuable source of information in this area, but generalization concerns have to be taken into account when addressing radicalization detection. Thus, intending to expand the scope of both our proposal and future research, we have collected the *Magazines dataset*.

Firstly, we have considered the Dabiq [48] and Rumiyah [49] online magazines, which are distributed by Islamic State of Iraq and the Levant (ISIS) radical organization [50]. These magazines are written in the English language. Dabiq was released from July 5, 2014, to July 31, 2016, by Al Hayat Media Centre, the branch of ISIS's Ministry of Media which produces material in English. After producing fifteen issues of Dabiq, the organization released the first issue of Rumiyah on September 6, 2016. Thirteen issues of Rumiyah have been released on more or less monthly schedule until September 9, 2017. Most analysts consider that Rumiyah has replaced Dabiq [51]. The change in the name is attributed to a change in ISIS's objectives: from Dabiq, a northern Syrian city where jihadists believe they would be defeated, to Rumiyah (Rome in Arabic), a West battle space outside Syria and Iraq [50].

Since the original Dabiq and Rumiyah publications are not freely accessible, they have been collected from an online resource that tackles terrorism¹. This online site makes the magazines available for research purposes. The source data is available in PDF format, so the textual data has been extracted. Since we aim to evaluate textual data, the images have been omitted. The resulting data consists of the 15 issues of Dabiq, and the 13 issues of Rumiyah. In total, the number of articles from Dabiq and Rumiyah is 161 and 155, respectively.

As a counterpoise of the previous radical content, we have selected two online newspapers that address ISIS-related issues but are not radical sources: CNN² and The New York Times³. These sources have been selected since both represent prominent newspapers that frequently mention radical and ISIS-related topics from a neutral perspective. Also, such an addition to the data has been done in the line of previous work [28], [52] where the authors complement radical content with non-radical data. Such content is freely accessible and can be obtained through the newspapers' APIs. The data has been obtained using domain-related keywords (*ISIS*, *Daesh*, *Islamic State*, etc.) on a time frame of 10 months. During this time, articles were automatically collected. In order to enhance the quality of the collected data, manual filtering was done, removing articles that are not related to the topic

¹<http://www.jihadology.net>

²<https://cnn.com> (<http://developer.cnn.com/docs/>)

³<https://www.nytimes.com> (<https://developer.nytimes.com/apis>)

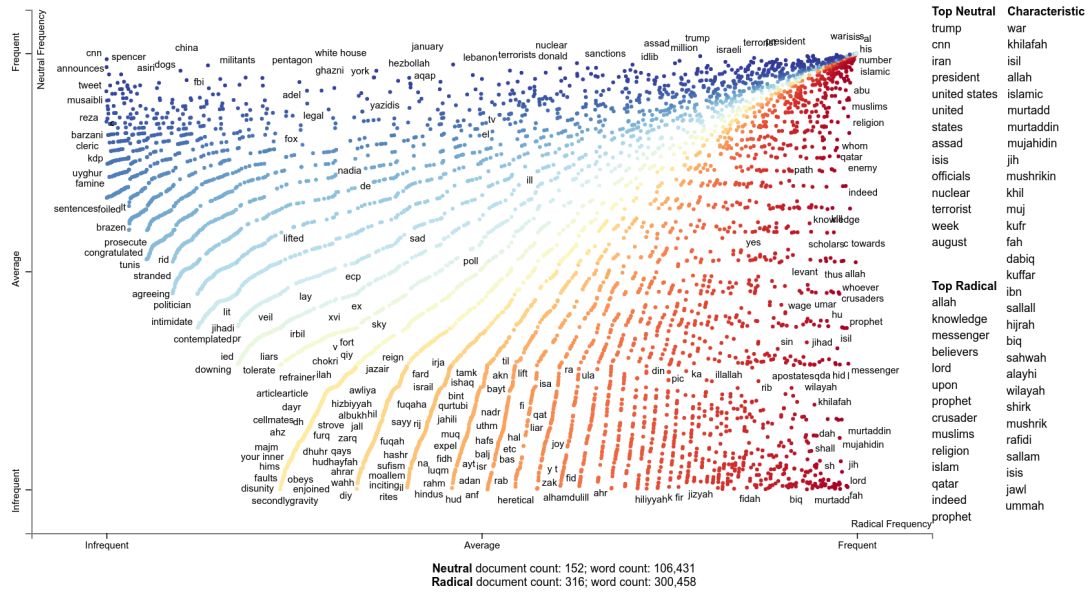


FIGURE 1. Frequency of words for both neutral and radical categories. On the right, most frequent words for neutral (Top Neutral) and radical (Top Radical), and both (Characteristic).

TABLE 2. Statistics of the collected dataset.

	CNN	NYT	Dabiq	Rumiyah
No. of articles	129	23	161	155
Avg. no. of words	860	271	1097	35
Avg. no. of sentences	35	10	35	32
Avg. no. of word appearances				
<i>Allah</i>	0.08	0.0	9.05	14.83
<i>knowledge</i>	0.09	0.08	0.55	0.90
<i>Trump</i>	1.87	0.09	0.01	0.0
<i>Iran</i>	2.46	0.30	0.22	0.07

at hand. Images, links, and other media are removed from the articles, leaving the text, both titles and bodies. In total, 129 articles have been collected from CNN, and 23 from The New York Times.

The same preprocessing has been applied to all the obtained data: normalization of numbering, capital letters, contractions (e.g., *I've*, *we'll*) is done, resulting in documents composed of lower case tokens. Table 2 presents a summary of some statistics of the resulting dataset.

Figure 1 shows a visualization of the introduced dataset, as described in [53]. This graph consists of a frequency scatter plot of the dataset words accordingly to the neutral and radical category. We consider the Dabiq and Rumiyah texts to be radical, in opposition to the CNN and New York Times articles, which we label as neutral. The color indicates the frequency in relation to each class: blue for neutral, red

for radical. For each category, the frequency of each different word to that category is computed. A subset of the word labels is placed along the figure due to space constraints. In this way, the x-axis indicates the frequency in the radical category: if a word frequently appears in radical texts, it is placed on the right area.

Similarly, the y-axis encodes the frequency in the neutral category. A word that frequently appears in neutral texts will be placed in the top area. As a consequence, it is of particular interest the areas where the more frequent words appear: top left (frequent in neutral texts), bottom right (frequent in radical texts) and top right of the figure (frequent in both neutral and radical texts). In those areas, the most characteristic words for the neutral, radical, and both categories are presented. These areas offer a view of how the words are used in these two categories. For example, common radical words are *Allah*, *knowledge*, and *messenger*, that stress the religious tone that can be found in the narratives of the Dabiq and Rumiyah magazines. In contrast, the articles found in the neutral newspapers have a more informative tone, frequently using words such as *Trump*, *CNN*, and *Iran*. Given the implication of the United States in the affairs related to ISIS, it is understandable that such words are being frequently used in the neutral class.

IV. EXPLOITING EMOTION FEATURES AND SEMANTIC SIMILARITY FOR RADICALIZATION DETECTION

This paper proposes a model that exploits several sources of information, aiming at improving the final performance in

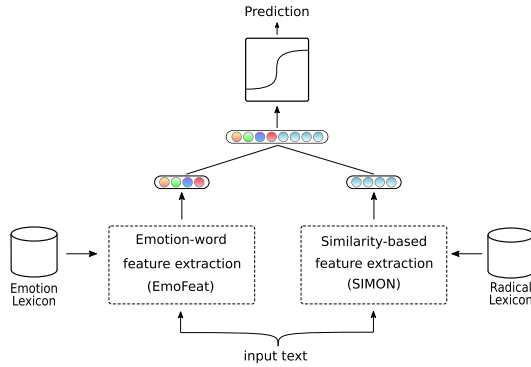


FIGURE 2. General architecture representation of the proposed model.

the task of radicalization detection. For this end, a machine learning system is introduced that consists of two submodules: (i) emotion-based (Sect. IV-A) and (ii) embedding word similarity feature extraction (Sect. IV-B).

Figure 2 shows a diagram of the proposed model. As shown, the analyzed text is processed by the two processing modules. Both of them use as input the natural language, yielding as output a feature vector that represents the input text. These feature vectors are concatenated and fed to a machine learning classifier, which outputs a prediction based on the information given by the features mentioned above. As classifiers, in this work, both Logistic Regression and Linear SVM are considered.

A. EMOTION BASED FEATURES

It has been discussed in the literature that emotions can play a role in the elicitation of radical behavior [6]. Sentiment information has been used in a machine learning system in order to distinguish between radical and non-radical content [54]. Nevertheless, to the extent of our knowledge, the effect of the emotions present in the text over a learning system has not been thoroughly studied in the task of radicalization detection.

In this manner, this work proposes the use of an emotion lexicon in order to extract emotion-driven features that are, as explained, fed to a machine learning algorithm. With this, we intend to investigate whether this kind of information is relevant for the task of radicalization detection and to which extent. So, an emotion lexicon-based representation is proposed that makes use of statistical measures to encode the emotion of the text. For simplicity, we denote this method as *EmoFeat* (Emotion Features).

Consider an emotion lexicon, composed by a vocabulary $W^{(l)} = \{w_1^{(l)}, \dots, w_i^{(l)}, \dots, w_P^{(l)}\}$ and a vector of numeric annotations $L = [l_1, \dots, l_i, \dots, l_P]$. Note that such lexicon has an annotation l_i for each word w_i , with a total of P (w_i, l_i) pairs. Note also that l_i is a vector expressing the intensity of each emotion for word $w_i^{(l)}$ in the lexicon. An emotion

vector has dimensionality $l_i \in \mathbb{R}^m$, and thus the emotion lexicon annotation matrix is $L \in \mathbb{R}^{P \times m}$, where m is the number of emotions considered in the lexicon. Following, let $W^{(i)} = \{w_1^{(i)}, \dots, w_j^{(i)}, \dots, w_I^{(i)}\}$ be the set of length I composed by the input words.

For each word w_k of the intersection $W^{(l)} \cap W^{(i)}$, the associated emotion vector is extracted from L . This process outputs a matrix that contains the emotion annotation for all the input words that appear in the lexicon. Next, several statistical measures are taken in order to represent such a matrix as a feature vector. The proposed measures are average, maximum, and median. Please note that these measures can be used independently: for example, only average and median could be used, removing maximum. As a result, a feature vector is obtained with dimension $n \cdot m$, where n is the number of statistical measures selected.

Algorithm 1 EmoFeat

Require: Emotion lexicon composed by a vocabulary $W^{(l)}$ and annotations L

Ensure: $v \in \mathbb{R}^{n \cdot m}$, the final feature vector

```

1: for all  $w_k \in W^{(l)} \cap W^{(i)}$  do
2:    $E_{k,:} \leftarrow \text{emotionAnnotation}(w_k, L)$ 
3: end for
4: for  $i \leftarrow 1, n$  do
5:   for  $j \in 1, m$  do
6:      $\text{index} \leftarrow i + n(j - 1)$ 
7:      $v_{\text{index}} \leftarrow \text{statMeasure}(i, E_{:,j})$ 
8:   end for
9: end for

```

The proposed feature extraction method can be expressed as an algorithm, as shown in Algorithm 1. The function *emotionAnnotation* extracts the emotion annotation vector corresponding to word w_k from matrix L . Also, the function *statMeasure* computes the corresponding statistical measure from the column j of matrix E . The index i indicates which statistical measure is applied (e.g., average if $i = 1$, maximum if $i = 2$, etc.).

B. EMBEDDING BASED SEMANTIC SIMILARITY

A well-known advancement of the NLP field is that distributed representation based techniques have become the state-of-the-art models on text processing problems [55]. Among the different methods for computing distributed representations, we highlight word embeddings, as they arguably represent the most popular choice of such text representations. In light of this, it is natural to presume that word embeddings can be leveraged for the task of radicalization detection effectively. Nevertheless, pre-trained word embedding models do not contain specific knowledge of the task at hand, since these models are trained from large corpora in an unsupervised manner.

One possible solution that has been studied before [56] would be to train a word embedding model using a

sufficiently large corpus with explicit radical detection annotations. Unfortunately, a data repository that fulfills these requirements is not currently available for the research community [57].

In this way, we propose the application of the SIMilarity-based sentiment projectiON (SIMON) method [7] as a feature extractor in radicalization detection. This method makes use of a word embedding model and orients the extracted features to a particular domain utilizing a domain-centered lexicon. In this work, the SIMON method is adapted to extract features for radicalization detection by using radical-oriented lexicons.

The main idea of the SIMON method is that given a domain lexicon, the input text is measured against it, computing a vector that encodes the similarity between the input text and the lexicon. With such a model, the model can exploit the knowledge contained in word embeddings, as well as the domain information that the lexicon offers. As an additional remark, this method does not need large corpora to be trained and thus can be used in problems where annotated data is scarce.

As stated, the application of SIMON towards this domain is achieved by composing domain-oriented lexicons. The generation of domain adapted lexicons is a challenge in itself [58], but in this work, we proposed a simple method for the generation of a domain-specific lexicon. Please note that, for the use of a lexicon made by the SIMON method, numeric annotations are not needed. As previously demonstrated [7], using such annotations does not necessarily lead to an improvement of results.

Thus, a collection of words is generated through filtering by the frequency of appearance in the training data. This selection of words is used as a lexicon for the SIMON method. Such a selection method is oriented to represent a simple baseline to capture the vocabulary of a specific domain. We refer to this frequency-based selection as *FreqSelect* in the rest of the paper.

Additionally, the SIMON method allows the generated collection of words to be subsequently filtered. Such filtering is done by means of an ANOVA-based features filtering, attending to the feature relevance for the classification task [7].

V. EVALUATION

The evaluation of the proposed model has been done through a text categorization task, where given a piece of text, the aim is to detect whether it does contain radicalization evidence. In this way, the evaluation is done by performing a binary classification task. As such, the proposed approaches have been validated using the datasets, lexicons, and embeddings listed in Section V-A, and following the methodology described in Section V-B. Results of such experiments are shown in V-C.

A. MATERIALS

The datasets used for the evaluation are presented in Table 3. Apart from the dataset presented in Section III, other two datasets have been used, which contain data from Twitter.

TABLE 3. Statistics of the datasets used for the evaluation: number of positive (radical), negative and total instances.

Dataset	Positive	Negative	Total
Pro-Neu [28]	112	112	224
Pro-Anti [11]	566	566	1,132
Magazines (this work)	316	152	468

TABLE 4. Statistics of the twitter datasets used in the evaluation.

Statistic	Pro-Neu [28]		Pro-Anti [11]	
	Pro	Neu	Pro	Anti
Total no. of tweets	17,350	197,743	602,511	1,368,827
Avg. no. of tweets	154	1,765	1,065	2,418
Total no. of words	271,842	3,151,107	3,945,815	9,375,841
Avg. no. of words	2,427	28,134	6,971	16,570

Pro-Neu. This dataset is composed of two different datasets, both in English, collected by [28]. The first one contains 17,350 tweets extracted from 112 distinct Twitter pro-ISIS accounts, which can be found online⁴. Through a study of three months, a set of users was identified using a selection of keywords (e.g., *Dawla, Amaq, Wilayat*), and filtered according to their use of images (e.g., radical leaders images, ISIS flags) and their network of followers. The second one, which comprises 122k tweets from more than 95k accounts, has been used as a counterexample from the pro-ISIS instances since it contains ISIS-related messages (that can be either neutral or anti-ISIS). This set of tweets was collected using ISIS-related keywords (e.g., *ISIS, ISIL, Daesh, IslamicState, Raqqa, Mosul*). From the original accounts, a selection of 112 is made, as done in [28]. This additional filtering is done filtering accounts that are not active nowadays from the original dataset, thus ensuring that the remaining accounts are not pro-ISIS. For comparison purposes, the same selection and split from [28] are performed.

Pro-Anti. formed by 1,132 Twitter accounts and their timelines, which have been collected in [11]. This dataset is in English. In the mentioned work, the authors identified users as pro-ISIS by attending to their sharing activity of incitement material from known pro-ISIS accounts, as well as the use of extreme language. Initially, the authors had 727 accounts identified, but 161 of these Twitter accounts were either suspended or hidden from public access. Such status prevents from accessing the profile information. Consequently, these 161 accounts were removed, obtaining 566 pro-ISIS users in total [11]. In order to balance the data, the authors added 566 anti-ISIS users. The annotation of anti-ISIS accounts has been done by observing the use of anti-ISIS language. This dataset has been used in [2], a fact that enables comparison with our work.

Magazines. which is the dataset presented in this work, as described in Section III.

⁴<https://www.kaggle.com/fifthtribe/how-isis-uses-twitter>

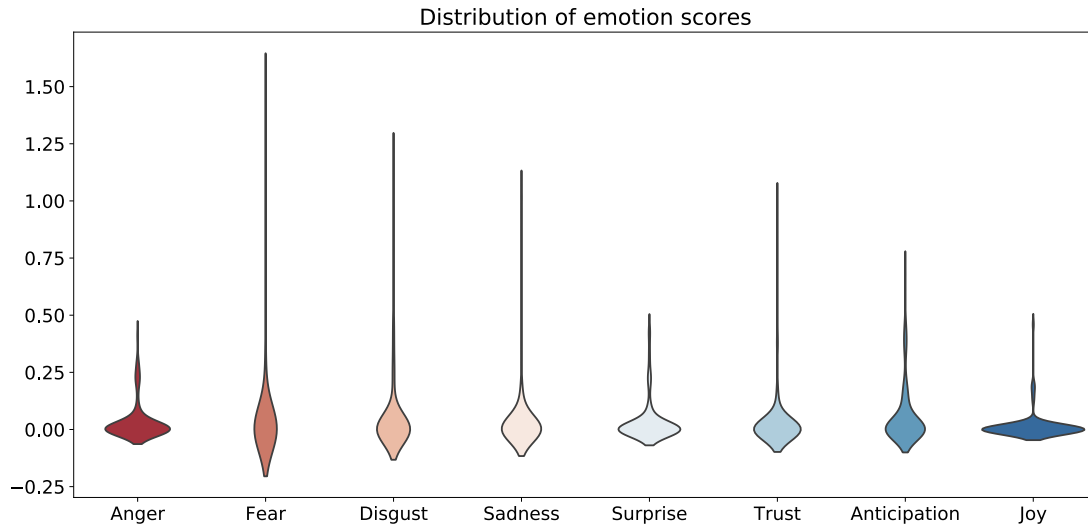


FIGURE 3. Distribution of raw emotion scores for all considered emotions.

TABLE 5. Statistics of the word vectors used for the evaluation: name of the vectors, reference, vocabulary size (no. of words), and training domain.

Name	Reference	Vocabulary size	Training domain
Word2Vec	[32]	3,000,000	News
FastText	[59]	1,999,995	Wikipedia
Glove	[60]	400,000	Wikipedia + Gigaword ⁵

Also, as described in Section IV, word embeddings are extensively used in the proposed model. In order to assess the differences in performance among various pre-trained models, we make use of three popular resources extensively used in the Natural Language Processing community. Table 5 shows the characteristics of such vectors. Please note that, in order to normalize the dimension of the vectors, in all three variants we use 300-dimensional vectors.

Another resource is the emotion lexicon used, as explained in Section IV-A. For this work, the selected lexicon is the NRC Hashtag Emotion Lexicon [61], which has been curated specifically for the Twitter domain. This resource contains 16,862 words annotated with eight emotions: anger, anticipation, disgust, fear, joy, sadness, surprise, and trust. Figure 3 shows the distribution of emotions scores along with a union of the evaluation data. Although a relevant portion of the scores is close to 0 (no emotional content) it can be observed that for all emotions, there is a long tail in the distribution. Such an elongated distribution contains a variation that is exploited by the emotion feature extraction model. We have examined the emotion distribution along to each dataset and annotation, confirming that there are no relevant differences between those. This indicates that our dataset is balanced in

terms of emotion distribution and that there are not any biases that may corrupt the evaluation.

B. EVALUATION METHODOLOGY

The proposed model (Section IV) can be implemented with a variety of resources and methods. In order to thoroughly evaluate the effectiveness of the model, an extensive experimental setup has been performed. In all experiments, the weighted average of the F1-Score is used as the performance metric. To evaluate an instance of our model, cross-validation via k-fold is used, with a $k = 10$.

Concerning the SIMON features, the experiments consider three variations, attending to which (i) word vectors and (ii) collection of words are used, as well as (iii) the percentage of the word collection retained. As explained in Section IV-B, a collection of words is used in order to compute the similarity to the analyzed text. This collection is extracted from the training dataset by measuring the appearance frequency of the dataset vocabulary: this is done by the FreqSelect method. Although, in order to compare to other word selections, we include in the experiments the use of two lexicons: Bing Liu's [62], a popular sentiment lexicon (as done in [7]); and a radical-oriented lexicon that is the result of merging different radical-related lexicons (for more information, please consult the original publication [28]). With respect to the third parameter, it expressed the percentage of the collections of words that is retained after performing an ANOVA based filtering. In this way, the extracted features can be filtered, attending to how relevant are for the classification problem [7]. A percentage of 100 would indicate that no filtering is performed.

As for the EmoFeat method, an evaluation of the number of emotions considered (parameter m) and the number of

the statistical measures selected (parameter n) is done. Thus, we use the Recursive Feature Elimination (RFE) method [63] to select among all the possible combinations of these parameters.

In order to evaluate the use of the different features by our model, the experiments explore the performance using the following selection of features: (i) EmoFeat, (ii) SIMON, and (iii) EmoFeat combined with SIMON. Also, we have tested two classifier algorithms: Logistic Regression and Linear SVM, since the primary objective is to evaluate the effectiveness of the feature extraction methods.

C. RESULTS

Firstly, in order to evaluate the influence of the number of emotions considered (parameter m) and the number of statistical measures selected (parameter n), the RFE method has been used [63]. This method outputs a feature ranking (the lower, the better) that can be used to eliminate features. Consequently, we explore such a ranking, as given by RFE, when applied to the features generated by the proposed EmoFeat approach. In this way, taking into account the selected emotion lexicon (NRC Hashtag Emotion Lexicon [61]), the maximum number of emotions considered is $m = 8$, and $n = 3$, which outputs a maximum number of features $m \cdot n = 24$. Following, the RFE strategy has been applied to these features across all considered datasets and machine learning classifiers. Attending to the parameter n , the *median* measure has a lower rank on average. Thus, we evaluate the F-Score obtained using the three measures (*average* + *maximum* + *median*), which raises to 79.01%, against the performance removing the *median* (*average* + *maximum*), that yields 79.18%. Please note that these F-Scores are averaged over all datasets and classifiers. Next, the parameter m is evaluated. The emotion that has the lower ranking, as computed by the RFE method, is *Joy*. Similarly, a comparison is made between the averaged performance including all emotions, which is again 79.18%, and the performance obtained when removing *Joy*, that outputs 78.97%. It can be seen that excluding the *Joy* emotion decreases the overall performance of the system. As a conclusion, for the rest of the experiments, we consider $m = 8$ (all emotions) and $n = 2$, selecting the average and maximum measures.

Secondly, we evaluate the three variations relevant to the SIMON approach: word vectors, collection of words, and percentage of filtering over this collection. Figure 4 shows the results. The percentage of selection (horizontal axis) presents a general trend, where the performance improves proportionally with the percentage. The best scores, with only one exception, are obtained when using a percentage of 100%; that is, with no filtering.

As for the word collections used, it can be seen the difference in performance among the three variations. As expected, the domain-oriented FreqSelect outperforms the other two collections in all cases. This improvement indicates that adapting the word collection leads SIMON to an increase in performance in the radical classification task. Nevertheless,

the radical collection used as baseline performs poorly in comparison to the sentiment-oriented collection. This accuracy difference could be explained attending to the composition of the radical lexicon, that contains a large number of Arabic terms that do not appear in the word vectors. In such cases, the similarity between the analyzed text and the out of vocabulary words cannot be computed, which leads to a loss of information that can be relevant to the problem at hand.

As for the word vector comparison, we observe that the best performance is obtained by using the FastText word vectors in most cases. Nevertheless, this difference can be seen more easily by attending to Tables 6 and 7. In these tables, the F-Score is shown for each combination of our models, and the best scores are marked in bold. Please note that the different word collections (sentiment, radical and FreqSelect) do not apply when uniquely evaluating the emotion-based features (Sect. IV-A), since this approach does not make use of such collections.

As mentioned, the best performance is obtained in almost all cases when using FastText embeddings. Also, in the cases where this is not the case, the difference in performance to that using FastText word vectors is small, being 0.45% the highest. This difference could be explained attending to training method of the embedding models: FastText does take into account subword information, while Word2Vec and Glove do not [59].

Paying attention to the extracted features, it is clear that although solely using the EmoFeat method does not yield the best results, its performance is competitive. This observation is especially true in the pro-neu dataset, where the F-Score obtained is 92.41% when using a Linear SVM classifier. This result enforces the idea that emotion analysis can play a relevant role in radical content detection. The scores obtained when using emotion-based features surpass that of specific approaches that merge social and computational concepts [28].

Following, the combination between the emotion-based and SIMON features has been evaluated. As shown, combining these two types of features does not always improve over the performances obtained by using SIMON uniquely. This is quite evident when looking at the best performances (in bold). Such an effect could be due to overfitting, since the combination augments the number of features, causing the performance score to stop improving, or even to decrease. The reduced size of the datasets further supports this hypothesis. Nevertheless, there is still a large number of cases where this combination improves over the separate methods, which indicates that a combination of the proposed features can lead to an improvement in the performance.

In order to further study the impact of the proposed features and their performance, we have performed the Friedman statistical test [64]. As a result, the Friedman test outputs a ranking of methods attending to their performance in the different datasets. The lower the ranking, the better a specific method performs in comparison to the rest. This test has been computed with $\alpha = 0.01$. For simplicity, Table 8 shows

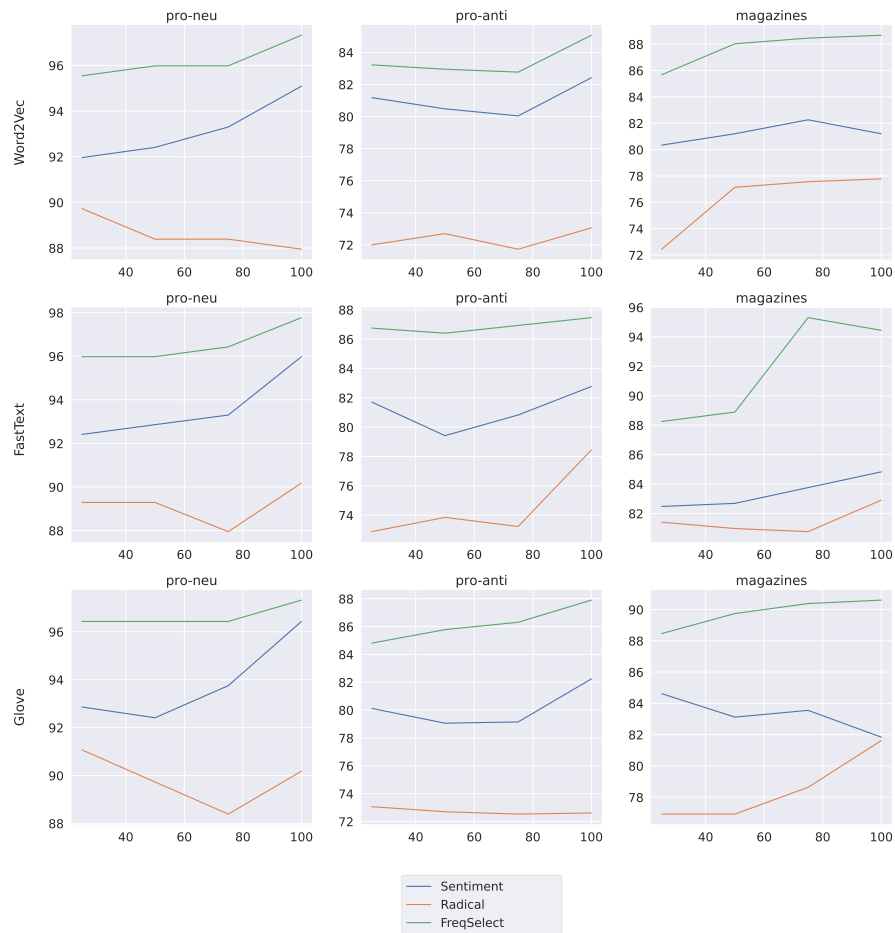


FIGURE 4. Evaluation of SIMON variations on the three datasets. Rows express the word vectors used, columns differentiates the datasets, and the word selections are shown in color. Vertical axis expresses the weighted F1-Score, while horizontal axis indicates the percentage of filtering.

TABLE 6. Averaged F1-Scores for the model using emotion, SIMON, and SIMON + emotion, using logistic regression as classification algorithm.

LOGISTIC REGRESSION		PRO-NEU			PRO-ANTI			MAGAZINES		
EmoFeat		91.07			73.95			68.59		
SIMON	Word2Vec	Sentiment	Radical	FreqSelect	Sentiment	Radical	FreqSelect	Sentiment	Radical	FreqSelect
	Facebook	95.09	87.95	97.32	82.42	73.06	85.07	81.20	77.78	88.68
	Glove	95.98	90.18	97.77	82.77	78.45	87.46	84.83	82.91	94.44
SIMON + Emotion	Word2Vec	96.43	90.18	97.32	82.24	72.61	87.90	81.84	81.62	90.60
	Facebook	95.54	88.39	97.32	82.77	75.00	85.16	81.84	80.34	89.53
	Glove	95.09	90.62	97.32	82.69	79.59	87.90	85.47	83.97	94.44
		96.43	91.07	97.77	82.77	73.67	87.10	83.33	82.26	91.03

the best six approaches, as computed by the Friedman test. As can be seen, the lower rank is obtained by two methods that make use of a Linear SVM classifier, the FastText word

vectors, as well as the FreqSelect word collection. More importantly, the features used are both SIMON, and SIMON combined with emotion. This is consistent with the current

TABLE 7. Averaged F1-Scores for the model using emotion, SIMON, and SIMON + emotion, using linear SVM as classification algorithm.

LINEAR SVM		PRO-NEU			PRO-ANTI			MAGAZINES		
EmoFeat		92.41			77.21			72.22		
		Sentiment	Radical	FreqSelect	Sentiment	Radical	FreqSelect	Sentiment	Radical	FreqSelect
SIMON	Word2Vec	95.54	80.80	97.77	80.12	65.72	83.92	77.78	75.43	88.46
	Facebook	95.09	89.29	99.11	78.71	69.52	86.22	83.12	79.49	94.02
	Glove	97.32	86.16	98.66	79.15	65.46	85.07	79.49	79.91	88.25
SIMON + Emotion	Word2Vec	95.09	88.39	97.77	79.51	64.75	83.75	79.27	79.27	87.39
	Facebook	95.09	91.07	99.11	78.80	72.53	86.57	83.12	83.12	93.80
	Glove	96.88	87.95	98.66	79.42	70.23	85.42	80.34	78.85	89.10

TABLE 8. Friedman rank for the six best methods. EF is the proposed EmoFeat method.

Classifier	Features	Embedding	Collection	Rank
Lin. SVM	SIMON	FastText	FreqSelect	3.5
Lin. SVM	SIMON + EF	FastText	FreqSelect	3.5
Log. Reg.	SIMON	FastText	FreqSelect	3.66
Log. Reg.	SIMON + EF	FastText	FreqSelect	4.67
Log. Reg.	SIMON + EF	GloVe	FreqSelect	5.17
Log. Reg.	SIMON	GloVe	FreqSelect	6.17

discussion and sheds light in the fact that the combination of features does not necessarily improve over the use of SIMON separately.

Next, a cross-dataset experiment has been performed, with the aim of studying how the idiosyncrasy of the data affect the generalization performance. Table 9 shows the results of this experiment, where the rows indicate the training dataset and the columns the test dataset.

As expected, the performance drop is relatively high. The differences in the datasets easily explain this. For example, a model trained in the pro-neu data suffers a performance decrease when predicting on pro-anti data; as seen, the non-radical class in the first leans towards the neutrality, while in the second it is composed of anti-radical messages. These differences affect cross-dataset performance. Additionally, when attending to the pro-neu/magazines experiment, we see that in general, the performance is higher than in the pro-neu/pro-anti case. Similarly, this can be explained by the observation that the magazines dataset is also composed of radical and neutral texts, which aligns with the pro-neu data. Also, it can be seen that the peculiarities of the magazines data make it difficult for the classifiers to generalize, especially to the pro-anti data.

An additional consideration can be made when comparing the SIMON + EmoFeat model against the isolated SIMON and EmoFeat counterparts. It can be seen that in many cases combining these features lead to an improvement of the F-Score (see pro-anti/pro-neu and pro-anti/magazines results). In light of this, we can assume that under certain circumstances, the feature combination can be used to augment the robustness of the proposed method. Nevertheless,

TABLE 9. Averaged F1-Scores for the model using emotion, SIMON, and SIMON + emotion, using logistic regression as classification algorithm in a cross-dataset evaluation.

		PRO-NEU	PRO-ANTI	MAGAZINES
		EmoFeat		
PRO-NEU	Sentiment	-	65.81	63.25
	Radical	73.66	-	66.24
	FreqSelect	45.54	47.53	-
		SIMON		
PRO-NEU	Sentiment	-	69.61	71.79
	Radical	-	66.25	68.80
	FreqSelect	-	61.75	70.30
PRO-ANTI	Sentiment	55.80	-	65.17
	Radical	66.07	-	67.74
	FreqSelect	54.02	-	60.90
MAGAZINES	Sentiment	66.07	63.43	-
	Radical	64.73	42.93	-
	FreqSelect	63.39	38.34	-
		SIMON + EmoFeat		
PRO-NEU	Sentiment	-	69.61	72.22
	Radical	-	66.61	67.31
	FreqSelect	-	61.75	71.79
PRO-ANTI	Sentiment	57.59	-	65.38
	Radical	71.43	-	71.79
	FreqSelect	56.70	-	68.38
MAGAZINES	Sentiment	60.27	56.63	-
	Radical	51.52	41.17	-
	FreqSelect	58.93	38.25	-

there are cases where such a combination does not improve but decreases the performance. This result can be observed in the cases where the training is done with the magazines data and the testing on the twitter data. Based on this observation, we hypothesize that the learning process done with the magazines is not applicable to the Twitter domain. This is to be expected since they represent two very different domains, and this last experiment is a cross-dataset evaluation.

VI. DISCUSSION

When comparing the presented method with previous works, we can attend to two different criteria: the method, and the affect model (Table 1). In the sense of the

method, some works do not make use of machine learning techniques to assess the radicalization of texts [11], [14], [16], [18], [19], [26].

In contrast, many of the related works include some machine learning techniques to study radicalization. In this regard, our method uses two popular machine learning algorithms, logistic regression and Support Vector Machine (SVM). Some works also use these or similar learning algorithms, such as [1], [2], [9], [15], [20], [22], [24]. The main difference with our work can be found in the feature extraction phase. The proposed method exploits a word embedding model and a domain knowledge lexicon to calculate a set of features that are fed to the learning algorithm. To our knowledge, this type of approach has not been proposed before in radicalization assessment.

When attending to the affect model described in previous work (Table 1), it can be observed that there is a trend of using lexicons. Indeed, our proposal also makes use of such kind of resource. Nevertheless, previous work exploit the lexicons by directly matching words in the analyzed text to the ones contained in the lexicon [14], [16], [18], [19]. In this way, out-of-vocabulary terms can not be properly modeled by these approaches. In contrast, our model can handle terms that are not in the lexicon by computing the similarity of the analyzed text to the lexicon, and thus taking advantage of the wide vocabulary contained in word embedding models [7].

An additional difference with the body of previous works is that our model uses three different sources of knowledge to perform the radicalization detection. Those are the emotion lexicon, the domain-adapted radicalization lexicon, and the word embeddings model. It is relevant to stress that the combination of both an affect and radicalization lexicon is novel, and has not been previously studied in related work.

In related literature authors regularly use sentiment rather than emotion signals (Sect. II). This largely limits the capacity of the models, as taking into account the emotions offers a more detailed categorization of the affect information. In addition, we implement a distributed emotion representation, which not only considers the most prominent emotion but rather exploits all emotions simultaneously.

It is also worth noting that although embeddings have been well known by the NLP community, its role in radicalization detection is not as extended as in other fields. Still, some recent works incorporate embeddings to their learning models. In [52], the authors explore the use of sentence embeddings for classifying texts into *radical* and *benign*, representing the text directly with the Doc2Vec model [65]. A more recent work [66] also makes use of word embeddings, using Word2Vec vectors to represent the text. Such representations are combined with psychological and behavioral features to perform radicalization detection. Our work computes a similarity measure of the analyzed text to a domain lexicon. For the generation of this domain-adapted lexicon, we propose a simple but effective approach, *FreqSelect*. Such a method generates a selection of words relevant to the domain that is extracted from the datasets.

Previous research has studied the Twitter datasets used in this work, which allows us to compare the performance of our approach to those that use the same datasets. The Pro-Neu was used in [28] where the authors present a two-dimensional model for classifying users radical and non-radical users, reporting 90.60% as F-Score. In comparison, the proposed emotion model reaches 91.07%, while the full model (EmoFeat and SIMON) yields 99.11%. In [28], the authors present a model that computes a two-dimensional metric indicating radicalization from individual and social influence. In contrast, the model presented in this paper uses a richer representation, combining semantic similarity and a distributed emotion representation.

When attending to the Pro-Anti dataset, the comparison is made against the work presented in [2]. This last work introduces a learning model that makes use of 5 different types of information sources, including the inferred knowledge from the semantic graph, contextual network information, and entity knowledge. When using only sentiment information through the SentiStrength lexicon [25], the authors report an F-Score of 85.40%. Our model, in comparison, outputs 87.90%, indicating the effectiveness of the proposed model. If attending to the full system, which includes semantic pattern detection and network features, their score raises to 92.30%. This is to be expected, since extracting contextual information (e.g., network information) is beneficial for classification tasks [67]. Nevertheless, exploiting the social context falls outside the scope of this paper.

The lack of resources related to radicalism has been a challenge as well as a limitation for conducting this research. Regarding social networks, we have taken advantage of public datasets provided by researchers and Kaggle, since radical accounts are banned. Nevertheless, this is a limitation of our study, since radical messages are in continuous evolution. In the same way, we have provided a cured dataset of radical texts that have been balanced with standard newspapers. Still, a broader selection of sources could be done to incorporate additional sources.

VII. CONCLUSION

This work proposes a machine learning model that exploits two kinds of information sources: (i) emotion and (ii) embedding-based semantic similarity features. The first approach has been designed to take advantage of an existing emotion dictionary in order to generate features that can be used for radicalization detection. Such an approach generates simple but effective features by computing a statistical summary of the emotion valence of the words in the text. As for the second approach, the SIMON model has been adapted to extract radicalization-oriented features. In this work, a method for generating a domain word collection is proposed (*FreqSelect*), which leads to consistent improvements in comparison with other existing collections. Also, as an additional contribution, the Magazines dataset is presented, which expands the scope of this and future research. Intending to conduct a thorough and comparable

evaluation, we use this new dataset, as well as other two existing ones that have been previously studied. As a further study, a statistical test verifies the conclusions drawn from the experiments. Besides, a cross-dataset evaluation is also done.

Previously, we presented three research questions that were inherent to the task of radicalization detection (Sect. I). In the first one, RQ1, we centered our attention on how emotion information can be used for radicalization detection. In this sense, a novel approach that exploits the presence of emotions in the text has been presented. As seen in the experiments, these emotion-based features can yield interestingly high scores, reaching an F-Score of 92.41% in a known dataset. It is reasonable to assume that these emotional features can be useful when performing radicalization detection. Also, results from the cross-validation experiments indicate that the proposed combination can help with generalization issues. As seen in Table 9, combining the emotion features in a cross-dataset setting can improve the final performance, resulting in a more robust system.

In relation to RQ2, it is asked whether semantic similarity features can be used in radicalization detection. For addressing this issue, we have applied an original feature extraction method (SIMON) to the radicalization detection domain. In order to do this, a novel method has been presented that generates a domain-oriented vocabulary, which can be used by the SIMON model. Given the experimental results, it is clear that this kind of features can accurately represent the analyzed text, allowing simple classifiers to obtain high classifications metrics: 99.11%, 86.22%, and 94.02% for the pro-neu, pro-anti and magazines datasets, respectively. Thus, given these promising results, we conclude that the application of this method can be further studied.

Following, RQ3 raises attention around the challenge of generating and using domain-oriented vocabularies. As presented, we proposed a method that effectively generates this kind of vocabularies: FreqSelect. Moreover, we have evaluated the use of this vocabulary, comparing it with similar selections. Seeing the results, we believe that such an approach performs positively.

To summarize, this paper presents a machine learning approach for categorizing radical content in both social media and magazines, exploiting an emotion dictionary, a word embeddings model, and a domain-adapted word selection. The presented advancement in automatic identification of online radicalization can be beneficial to counter-extremist agencies, since these organizations have as a key priority the use of intelligent technologies in this direction.

Although the obtained results are very positive, we believe that as future work, the effect of word embeddings can be further studied. That is, obtaining domain-oriented word vectors could largely improve the text representations. Besides, studying the effect of temporal effects on the radical narratives can be interesting, since the use of the language is prone to change as time progresses.

ACKNOWLEDGMENT

The authors would like to thank the Trivalent Team, specially M. Fernandez, H. Alani, R. Denaux, J. M. Gomez-Perez, R. Barbado, and D. Suarez for their help in this research.

REFERENCES

- [1] A. Bermingham, M. Conway, L. Mcinerney, N. O'hare, and A. F. Smeaton, "Combining social network analysis and sentiment analysis to explore the potential for online radicalisation," in *Proc. Int. Conf. Adv. Social Netw. Anal. Mining*, Jul. 2009, pp. 231–236.
- [2] H. Saif, T. Dickinson, L. Kastler, M. Fernandez, and H. Alani, "A semantic graph-based approach for radicalisation detection on social media," in *The Semantic Web (Lecture Notes in Computer Science)*, vol. 10249, E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, and O. Hartig, Eds. Cham, Switzerland: Springer, 2017.
- [3] J. Van Stekelenburg, "Radicalization and violent emotions," *APSC*, vol. 50, no. 4, pp. 936–939, Oct. 2017.
- [4] S. K. Rice, "Emotions and terrorism research: A case for a social-psychological agenda," *J. Criminal Justice*, vol. 37, no. 3, pp. 248–255, May 2009.
- [5] R. Pelzer, "Policing of terrorism using data from social media," *Eur. J. Secur. Res.*, vol. 3, no. 2, pp. 163–179, Oct. 2018.
- [6] C. Mccauley and S. Moskalenko, "Mechanisms of political radicalization: Pathways toward terrorism," *Terrorism Political Violence*, vol. 20, no. 3, pp. 415–433, Jul. 2008, doi: [10.1080/09546550802073367](https://doi.org/10.1080/09546550802073367).
- [7] O. Araque, G. Zhu, and C. A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowl.-Based Syst.*, vol. 165, pp. 346–359, Feb. 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705118305926>
- [8] G. Zhu and C. A. Iglesias, "Exploiting semantic similarity for named entity disambiguation in knowledge graphs," *Expert Syst. Appl.*, vol. 101, pp. 8–24, Jul. 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417418300897>
- [9] A. Abbasi and H. Chen, "Affect intensity analysis of dark Web forums," in *Proc. IEEE Intell. Secur. Informat.*, May 2007, pp. 282–288.
- [10] S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *Proc. Int. Conf. Lang. Resour. Eval. (LREC)*, vol. 10, 2010, pp. 2200–2204.
- [11] M. Rowe and H. Saif, "Mining pro-isis radicalisation signals from social media users," in *Proc. 10th Int. AAAI Conf. Web Social Media (ICWSM)*, 2016, pp. 329–338.
- [12] L. Deng and J. Wiebe, "MPQA 3.0: An entity/event-level sentiment corpus," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 1323–1328.
- [13] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, "A large scale arabic sentiment lexicon for arabic opinion mining," in *Proc. EMNLP Workshop Arabic Natural Lang. Process. (ANLP)*, 2014, pp. 165–173.
- [14] A. Jurek, M. D. Mulvenna, and Y. Bi, "Improved lexicon-based sentiment analysis for social media analytics," *Secur. Informat.*, vol. 4, no. 1, p. 9, 2015.
- [15] H. Chen, "Sentiment and affect analysis of dark Web forums: Measuring radicalization on the Internet," in *Proc. IEEE Int. Conf. Intell. Secur. Informat.*, Jun. 2008, pp. 104–109.
- [16] M. Vergani and A.-M. Bliuc, "The evolution of the ISIS' language: A quantitative analysis of the language of the first year of Dabiq magazine," *Sicurezza, Terrorismo e Societa Secur., Terrorism Soc.*, vol. 2, no. 2, pp. 7–20, 2015.
- [17] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic inquiry and word count: LIWC 2001," *Mahway, Lawrence Erlbaum Associates*, vol. 71, no. 2001, p. 2001. 2001.
- [18] T. Chalothorn and J. Ellman, "Affect analysis of radical contents on Web forums using sentiwordnet," *Int. J. Innovation Manage. Technol.*, vol. 4, no. 1, pp. 122–124, 2013.
- [19] S. Ghajar-Khosravi, P. Kwantes, N. Derbentseva, and L. Huey, "Quantifying salient concepts discussed in social media content: A case study using Twitter content written by radicalized youth," *J. Terrorism Res.*, vol. 7, no. 2, p. 79, May 2016.
- [20] P. Dewan, A. Suri, V. Bharadhwaj, A. Mithal, and P. Kumaraguru, "Towards understanding crisis events on online social networks through pictures," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*. New York, NY, USA: ACM, 2017, pp. 439–446.

- [21] D. Borth, T. Chen, R. Ji, and S.-F. Chang, "SentiBank: Large-scale ontology and classifiers for detecting sentiment and emotions in visual content," in *Proc. 21st ACM Int. Conf. Multimedia (MM)*, New York, NY, USA: ACM, 2013, pp. 459–460.
- [22] M. Ashcroft, A. Fisher, L. Kaati, E. Omer, and N. Prucha, "Detecting Jihadist messages on Twitter," in *Proc. Eur. Intell. Secur. Informat. Conf.*, Sep. 2015, pp. 161–164.
- [23] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.
- [24] S. Agarwal and A. Sureka, "Using KNN and SVM based one-class classifier for detecting online radicalization on Twitter," in *Distributed Computing and Internet Technology (Lecture Notes in Computer Science)*, vol. 8956, R. Natarajan, G. Barua, and M. R. Patra, Eds. Cham, Switzerland: Springer, 2015.
- [25] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment strength detection for the social web," *J. Amer. Soc. Inf. Sci.*, vol. 63, no. 1, pp. 163–173, Jan. 2012.
- [26] D. López-Sánchez, J. Revuelta, F. de la Prieta, and J. M. Corchado, "Towards the automatic identification and monitoring of radicalization activities in Twitter," in *Knowledge Management in Organizations (Communications in Computer and Information Science)*, vol. 877, L. Uden, B. Hadzima, and I. H. Ting, Eds. Cham, Switzerland: Springer, 2018.
- [27] D. Correa and A. Sureka, "Solutions to detect and analyze online radicalization: A survey," Jan. 2013, *arXiv:1301.4916*. [Online]. Available: <https://arxiv.org/abs/1301.4916>
- [28] M. Fernandez, M. Asif, and H. Alani, "Understanding the roots of radicalization on Twitter," in *Proc. 10th ACM Conf. Web Sci. (WebSci)*, New York, NY, USA: ACM, 2018, pp. 1–10, doi: [10.1145/3201064.3201082](https://doi.org/10.1145/3201064.3201082).
- [29] S. Agarwal and A. Sureka, "Topic-specific YouTube crawling to detect online radicalization," in *Databases in Networked Information Systems (Lecture Notes in Computer Science)*, vol. 8999, W. Chu, S. Kikuchi, and S. Bhalla, Eds. Cham, Switzerland: Springer, 2015.
- [30] E. Ferrara, W. Q. Wang, O. Varol, A. Flammini, and A. Galstyan, "Predicting online extremism, content adopters, and interaction reciprocity," in *Social Informatics (Lecture Notes in Computer Science)*, vol. 10047, E. Spiro and Y. Y. Ahn, Eds. Cham, Switzerland: Springer, 2016.
- [31] S. Agarwal and A. Sureka, "Applying social media intelligence for predicting and identifying on-line radicalization and civil unrest oriented threats," Nov. 2015, *arXiv:1511.06858*. [Online]. Available: <https://arxiv.org/abs/1511.06858>
- [32] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Jan. 2013, *arXiv:1301.3781*. [Online]. Available: <https://arxiv.org/abs/1301.3781>
- [33] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, "Evaluation methods for unsupervised word embeddings," in *Proc. 2015 Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 298–307.
- [34] D. Zhang, H. Xu, Z. Su, and Y. Xu, "Chinese comments sentiment classification based on word2vec and SVMperf," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1857–1863, Mar. 2015.
- [35] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417417300751>
- [36] S. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "SemEval-2018 task 1: Affect in tweets," in *Proc. The 12th Int. Workshop Semantic Eval.*, 2018, pp. 1–17.
- [37] C. Van Hee, E. Lefever, and V. Hoste, "SemEval-2018 task 3: Irony detection in English tweets," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 39–50.
- [38] K. Gábor, D. Buscaldi, A.-K. Schumann, B. Qasemizadeh, H. Zargayouna, and T. Charnois, "SemEval-2018 task 7: Semantic relation extraction and classification in scientific papers," in *Proc. 12th Int. Workshop Semantic Eval.*, 2018, pp. 679–688.
- [39] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," Jul. 2016, *arXiv:1607.01759*. [Online]. Available: <https://arxiv.org/abs/1607.01759>
- [40] O. Araque, G. Zhu, M. García-Amado, and C. A. Iglesias, "Mining the opinionated Web: Classification and detection of aspect contexts for aspect based sentiment analysis," in *Proc. IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, Dec. 2016, pp. 900–907.
- [41] S. Lai, K. Liu, S. He, and J. Zhao, "How to generate a good word embedding," *IEEE Intell. Syst.*, vol. 31, no. 6, pp. 5–14, Nov./Dec. 2016.
- [42] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowl.-Based Syst.*, vol. 89, pp. 14–46, Nov. 2015.
- [43] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Meta-level sentiment models for big social data analysis," *Knowl.-Based Syst.*, vol. 69, pp. 86–99, Oct. 2014.
- [44] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the OMG!" in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, pp. 538–541.
- [45] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the State-of-the-art in sentiment analysis of tweets," Aug. 2013, *arXiv:1308.6242*. [Online]. Available: <https://arxiv.org/abs/1308.6242>
- [46] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.-Assoc. Comput. Linguistics*, vol. 1, 2011, pp. 151–160.
- [47] C. Zirn, M. Niepert, H. Stuckenschmidt, and M. Strube, "Fine-grained sentiment analysis with structural features," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, 2011, pp. 336–344.
- [48] H. K. Gambhir, "DABIQ: The strategic messaging of the Islamic State," *Inst. Study War*, vol. 15, pp. 1–12, Nov. 2014.
- [49] R. Mahzam, "Rumiyah-jihadist propaganda & information warfare in cyberspace," *Counter Terrorist Trends Analyses*, vol. 9, no. 3, pp. 8–14, 2017.
- [50] N. A. Azman, "'Islamic State' (IS) propaganda: DabiQ and future directions of 'Islamic State,'" *Counter Terrorist Trends Analyses*, vol. 8, no. 10, pp. 3–8, 2016. [Online]. Available: <http://www.jstor.org/stable/26351457>
- [51] P. Wignell, S. Tan, K. O'Halloran, and R. Lange, "A mixed methods empirical examination of changes in emphasis and style in the extremist magazines dabiQ and rumiyah," *Perspect. Terrorism*, vol. 11, no. 2, pp. 2–20, 2017.
- [52] A. H. Johnston and G. M. Weiss, "Identifying sunni extremist propaganda with deep learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov./Dec. 2017, pp. 1–6.
- [53] J. S. Kessler, "Scattertext: A Browser-based tool for visualizing how corpora differ," Mar. 2017, *arXiv:1703.00565*. [Online]. Available: <https://arxiv.org/abs/1703.00565>
- [54] B. S. Iskandar, "Terrorism detection based on sentiment analysis using machine learning," *J. Eng. Appl. Sci.*, vol. 12, no. 3, pp. 691–698, 2017.
- [55] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing [review article]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.
- [56] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. for Comput. Linguistics (Long Papers)*, vol. 1, 2014, pp. 1555–1565.
- [57] M. Fernandez and H. Alani, "Contextual semantics for radicalisation detection on Twitter," in *Proc. Workshop Semantic Web Social Good (SW4SG), 17th Int. Semantic Web Conf. (ISWC)*, Monterey, CA, USA, Oct. 2018. [Online]. Available: <http://ceur-ws.org/Vol-2182/>
- [58] O. Araque, M. Guerini, C. Strapparava, and C. A. Iglesias, "Neural domain adaptation of sentiment lexicons," in *Proc. 7th Int. Conf. Affect. Comput. Intell. Interact. Workshops Demos (ACIIW)*, Oct. 2017, pp. 105–110.
- [59] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Jul. 2016, *arXiv:1607.04606*. [Online]. Available: <https://arxiv.org/abs/1607.04606>
- [60] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
- [61] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets," *Comput. Intell.*, vol. 31, no. 2, pp. 301–326, 2015, doi: [10.1111/coin.12024](https://doi.org/10.1111/coin.12024).
- [62] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA: ACM, 2004, pp. 168–177.
- [63] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.
- [64] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
- [65] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

- [66] M. Nouh, J. R. C. Nurse, and M. Goldsmith, "Understanding the radical mind: Identifying signals to detect extremist content on Twitter," May 2019, *arXiv:1905.08067*. [Online]. Available: <https://arxiv.org/abs/1905.08067>
- [67] J. F. Sánchez-Rada and C. A. Iglesias, "Social context in sentiment analysis: Formal definition, overview of current trends and framework for comparison," *Inf. Fusion*, vol. 52, pp. 344–356, Dec. 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1566253518308704>



OSCAR ARAQUE received the graduate and master's degrees in telecommunication engineering from the Technical University of Madrid (Universidad Politécnica de Madrid), Spain, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree. He is currently a Teaching Assistant with the Universidad Politécnica de Madrid. His research interest includes the application of machine learning techniques for natural language processing. The main topic of his thesis is the introduction of specific domain knowledge into machine learning systems in order to enhance sentiment and emotion analysis techniques.



CARLOS A. IGLESIAS received the telecommunications engineering degree and the Ph.D. degree in telecommunications from the Universidad Politécnica de Madrid (UPM), Spain, in 1993 and 1998, respectively. He is currently an Associate Professor with the Telecommunications Engineering School, UPM, where he has been leading the Intelligent Systems Group, since 2014. He has been the Principal Investigator on numerous research grants and contracts in the field of advanced social and the IoT systems, funded by the regional, national, and European bodies. His primary research interests include social computing, multiagent systems, information retrieval, sentiment and emotion analysis, linked data, and web engineering.

...

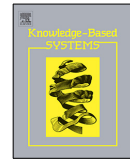
3.2.2 MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction

Title	MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction
Authors	Araque, Oscar and Gatti, Lorenzo and Kalimeri, Kyriaki
Journal	Knowledge-Based Systems
Impact factor	JCR 2019 5.921
Scopus citations	1 (02/09/2020)
Web of Science citations	0 (02/09/2020)
Google Scholar citations	4 (02/09/2020)
ISSN	0950-7051
Year	2020
Keywords	Machine learning, Natural language processing, Lexicon, Moral foundations, Moral values, Twitter data
Pages	105184
DOI	https://doi.org/10.1016/j.knosys.2019.105184
Online	http://www.sciencedirect.com/science/article/pii/S095070511930526X
Abstract	Moral rhetoric plays a fundamental role in how we perceive and interpret the information we receive, greatly influencing our decision-making process. Especially when it comes to controversial social and political issues, our opinions and attitudes are hardly ever based on evidence alone. The Moral Foundations Dictionary (MFD) was developed to operationalize moral values in the text. In this study, we present MoralStrength, a lexicon of approximately 1,000 lemmas, obtained as an extension of the Moral Foundations Dictionary, based on WordNet synsets. Moreover, for each lemma it provides with a crowdsourced numeric assessment of Moral Valence, indicating the strength with which a lemma is expressing the specific value. We evaluated the predictive potentials of this moral lexicon, defining three utilization approaches of increased complexity, ranging from lemmas' statistical properties to a deep learning approach of word embeddings based on semantic similarity. Logistic regression models trained on the features extracted from MoralStrength, significantly outperformed the current state-of-the-art, reaching an F1-score of 87.6% over the previous 62.4% (p-value ≤ 0.01), and an average F1-Score of 86.25% over six different datasets. Such findings pave the way for further research, allowing for an in-depth understanding of moral narratives in text for a wide range of social issues.



Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knossysMoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction[☆]Oscar Araque^{a,*,1}, Lorenzo Gatti^{b,2}, Kyriaki Kalimeri^{c,3}^a Intelligent Systems Group, Universidad Politécnica de Madrid, Madrid, Spain^b Human Media Interaction Lab, University of Twente, Enschede, The Netherlands^c Data Science & Digital Philanthropies Laboratory, ISI Foundation, Turin, Italy

ARTICLE INFO

Article history:

Received 17 April 2019

Received in revised form 31 October 2019

Accepted 2 November 2019

Available online 5 November 2019

Keywords:

Moral foundations

Moral values

Lexicon

Twitter data

Natural language processing

Machine learning

ABSTRACT

Moral rhetoric plays a fundamental role in how we perceive and interpret the information we receive, greatly influencing our decision-making process. Especially when it comes to controversial social and political issues, our opinions and attitudes are hardly ever based on evidence alone. The Moral Foundations Dictionary (MFD) was developed to operationalize moral values in the text. In this study, we present *MoralStrength*, a lexicon of approximately 1,000 lemmas, obtained as an extension of the Moral Foundations Dictionary, based on WordNet synsets. Moreover, for each lemma it provides with a crowdsourced numeric assessment of *Moral Valence*, indicating the strength with which a lemma is expressing the specific value. We evaluated the predictive potentials of this moral lexicon, defining three utilization approaches of increased complexity, ranging from lemmas' statistical properties to a deep learning approach of word embeddings based on semantic similarity. Logistic regression models trained on the features extracted from *MoralStrength*, significantly outperformed the current state-of-the-art, reaching an F1-score of 87.6% over the previous 62.4% (p -value < 0.01), and an average F1-Score of 86.25% over six different datasets. Such findings pave the way for further research, allowing for an in-depth understanding of moral narratives in text for a wide range of social issues.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Language usage reflects our thoughts, emotions, values, and culture, as we communicate with others. With the burst of on-line communication and social media, people are empowered to express and broadcast their opinions on contentious issues, timely, and at greater scale. This unprecedented opportunity allows scientists and policymakers to study phenomena such as opinion formation, radicalization, and polarization in society, as they happen.

In this study, we propose a lexicon for detecting and quantifying the moral rhetoric behind people's judgments, as reflected in spontaneous digital interactions. Moral values influence the way

we rationalize and take a stance upon controversial topics, like abortion [1], homosexuality [1], climate change [2], poverty [3] or health related attitudes like for instance exercising [4] and vaccine advocacy [5,6]. They are also closely related to our political views [7] and the opinion formation mechanisms regarding immigration [8], political extremism [9,10], and intimate partner violence [11]. Recently, scientists also showed that moral values could be employed to detect violent protests [12] based on user-generated text.

We operationalize morality via the Moral Foundations Theory (MFT) [13], which expresses the psychological basis of morality in terms of innate intuitions, defining the following five foundations: *care/harm*, *fairness/cheating*, *loyalty/betrayal*, *authority/subversion*, and *purity/degradation* (see [14,15]). Even if in its infancy, MFT is the most well-established theory in the psychology and social sciences. It is also broadly adopted in the computational social science field since it defines a clear taxonomy of values together with a term dictionary, the Moral Foundations Dictionary (MFD, hereafter) [13], which is an essential resource for natural language processing applications. The creators of the MFD, highlight the difficulty of creating such a resource since linguistic, cultural, and historical context reflect on language usage. Among the most significant limitations of the MFD, we have: (i) a limited amount of lemmas and stem of

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.knossys.2019.105184>.

* Corresponding author.

E-mail addresses: o.araque@upm.es (O. Araque), l.gatti@utwente.nl (L. Gatti), kkalimeri@acm.org (K. Kalimeri).

¹ <https://orcid.org/0000-0003-3224-0001>

² <https://orcid.org/0000-0003-2422-5055>

³ <https://orcid.org/0000-0001-8068-5916>

<https://doi.org/10.1016/j.knossys.2019.105184>

0950-7051/© 2019 Elsevier B.V. All rights reserved.

words; (ii) “radical” lemmas rarely used in everyday language, for instance, “homologous” and “apostasy”; and (iii) an association with a moral bipolar scale, so-called vice and virtue, but without any indication of “strength”.

Here, we address precisely these shortcomings; initially, we expanded the existing MFD using the WordNet lexical database [16] and then, we provide a set of normative ratings for empirical assessment of morality. The resulting lexicon, namely *MoralStrength*, offers approximately three times more lemmas while going beyond the binary nature of the MFD. Moreover, we present a machine learning framework exploring the potentials of *MoralStrength* in predicting the moral narratives from the user-generated text.

The suggested framework includes three models of increasing complexity; two of them are based on straightforward feature extraction from lemma frequencies and statistical properties, while the third one is based on embedding representation of semantic similarity. We thoroughly evaluate the proposed framework employing the Moral Foundations Twitter Corpus (MFTC) [17]. The MFTC corpus is a collection of seven Twitter dataset previously employed in studies related to moral detection from text. It consists of approximately 35,000 tweets along with their respective annotations according to the MFT foundations regarding critical social issues.

Importantly, the performance of our approach in predicting morality is outperforming the current state-of-the-art methods. Our results show that the pure textual representations emerged from the *MoralStrength* lexicon greatly benefit the performance of the prediction. These findings pave the way towards a more in-depth understanding of moral judgments, dispositions, and attitudes formation from spontaneous digital data.

Hence, we contribute to the research and policymaking communities with a useful resource and a concrete framework that can be employed for analyzing large scale user-generated communications, or even nowcasting people's attitudes and opinions on controversial phenomena. When it comes to critical social issues, the proposed approach can provide insights to understand personal narratives and viewpoints better, but also how people will potentially reason on the information they receive. Such knowledge is essential for policymaking specialists to design effective communication campaigns that appeal to people's values, given the ever-increasing penetration of social media to the population.

2. Related literature

Psychologists and social scientists have systematically analyzed text data to address their research questions. Back in the late '60s, the Harvard General Inquirer dictionary [18] was the precursor of sentiment analysis which was to become, together with opinion extraction, a core theme of natural language processing (NLP). Ever since scientists gradually increased complexity moving from simple techniques (e.g. the unsupervised and partially rule-based approach of [19]) to sophisticated methods that try to determine the context of words (e.g. [20]). The topics addressed also became more challenging tackling notions such as irony and sarcasm [21]. With time, not only the methods became more sophisticated but also the tasks become more ambitious. Extensive studies on linguistic markers of sentiment and affect [20,22–25] paved the way to assess more complex constructs such as personality [26,27] and human values [28,29]. Moral values are considered to be a higher level construct with respect to personality traits, determining how and when dispositions and attitudes relate with our life stories and narratives [30]. Here we provide a brief literature review of the studies regarding moral values assessment from textual data. As in sentiment and

personality analysis, also, in this case, pioneer works followed a dictionary-based approach, while the current state of the art performance is based on deep learning.

The first vocabulary developed to assess the moral values from textual data was the Moral Foundations Dictionary (MFD) [13]. It was used together with the Linguistic Inquiry and Word Count (LIWC) software [31] to estimate moral traits and to investigate differences in moral concerns between different cultural groups. Clifford et al. [32] employed the MFD for performing manual text analysis of 12 years of coverage in the New York Times focusing on political debate in the US. Teernstra et al. [33] assessed the political debate regarding the “Grexit” from approximately 8000 tweets. They compared the performance of using the raw data, bigrams, and the MFD features in employing basic machine learning models, namely, Naive Bayes (NB) and Maximum Entropy (ME). They concluded that pure machine learning is preferable to dictionary approaches since it has similar prediction accuracy while using fewer assumptions. In this study, we follow a similar approach to [33]; however, we propose an expanded version of the MFD, including also the moral valence per lemma. Moreover, we employ logistic regression models to infer moral values from uni-grams combined with lexical features.

Dehghani et al. [34] examined the differences between liberal and conservative moral value systems using a hierarchical generative topic modeling technique, based on Latent Dirichlet Allocation (LDA) [35], to enable the unsupervised detection of topics in their corpus of liberal and conservative weblogs. They used small sets of words selected from the MFD as seeds to encourage the emergence of issues related to different moral concerns and examined similarities and differences in how such matters are expressed between these groups. Consistently with findings in moral psychology, they demonstrate that there are significant differences in how liberals and conservatives construct their moral belief systems. Sagi et al. [36] employed the same framework to study moral rhetoric in text for a specific case study, the US Federal shutdown of 2013 [1], where they examined the role of morals in intra- and inter-community differences of political party retweets. In both works, they were based on the framework presented in [34], where LDA was employed to create a co-occurrence matrix on which the similarity between the texts and the vectors representing the different MFT moral traits was computed. In a similar approach, Kaur et al. [37] attempted to quantify the moral loadings of text, based on the Latent Semantic Analysis (LSA). They used a bag-of-words model, representing the entire corpus by a word-context matrix. Then they reduced its dimensionality obtaining low-dimensional word vectors, in which similar vectors represent similar meaning words. Our study is presenting a different approach since we do not use LSA representations, but rather pre-trained word embeddings models. Although pre-trained word embeddings do not contain domain-specific knowledge, they express language regularities encoded as offsets in the resulting vector space. The proposed representations based on the work of Araque et al. [38], exploit precisely the similarity between the analyzed text and a selection of words with moral content.

More recently, Garten et al. [39] employed the MFD to detect moral rhetoric in general, and more specifically, shifts in long political speeches over time. Then, based on psychological dictionaries and semantic similarity to quantify the presence of moral sentiment around a given topic, Garten et al. [40], proposed the Distributed Dictionary Representations (DDR) method. Showing promising results, DDR was also employed by Hoover et al. [41] to detect moral values in charitable giving. Later on, Garten et al. [42] extended the method, incorporating demographic embeddings into the language representations. Our approach is based on an expanded version of the MFD, with evaluated manual annotations regarding the moral valence of each lemma, that can be incorporated in computational frameworks.

In a study more similar to ours, attempting to predict moral values involved in Twitter posts, Lin et al. [43] proposed a method that automatically acquires background knowledge to improve the moral value prediction, pointing out the difficulty of the task also for human experts. Based on the work of [12,13,43] predicted the moral sentiment of the tweets. Their model consists of three layers, an embedding (lookup) layer, a recurrent neural network (RNN) with long short-term memory (LSTM) [44] and an output layer. The first layer converts words in an input tweet to a sequence of pre-trained word embeddings, the LSTM layer processes these embeddings and outputs a fixed-sized vector which encodes critical information for moral value prediction, while a vector representing the percentage of words that match each category in the Moral Foundations Dictionary [13] are concatenated with the LSTM feature vector. Our approach is differentiating to this one since we employ word embeddings to compute the similarity between words rather than directly feeding them in a neural network architecture. Along the same line, Rezapour et al. [45] investigated the relationship between moral values and stance towards a series of social issues. Their findings show that enhancing the original MFD improves the prediction accuracy of morality in text. They expanded the original MFD and employed a series of machine learning classifiers (SVM, RF and LSTM) predicting the moral traits. This study underlines the importance of expanding the MFD; we go one step further introducing the notion of moral “strength” while showing how more abundant information is improving the overall accuracy of the models.

The core contribution of this study is the extended lexicon of moral lemmas with the respective moral valence. To explore the properties and full potentials of the lexicon, we suggest three different models of increasing complexity, demonstrating the value of this resource. The proposed approaches range from feature engineering methods to a system which employs word embeddings of semantic similarity based on the work of Araque et al. [38].

3. Materials and methods

3.1. Expansion of the moral foundations dictionary (MFD)

The cornerstone of our study is the Moral Foundations Dictionary (MFD) [13] which was created to capture the moral rhetoric according to the five predefined dimensions defined by the Moral Foundations Theory (MFT) [46]. The original MFD⁴ consists of lemmas and stems divided into “virtue” and “vice” [13] for each foundation according to their moral polarity. “Virtue” words are foundation-supporting words (e.g., *safe** and *shield* for Care “virtue”), whereas “vice” words are foundation-violating words (e.g., *kill* and *ravage* for Care “vice”). MFD [13] was meant to be used together with the Linguistic Inquiry and Word Count (LIWC) program [47], and thus contains either lemmas (158 entries such as *abandon*) or stems with a wild-card sign (166 listings), that LIWC analysis uses to match with all the forms of the base word; for instance, the entry *abuse** will match “abuse”, “abuses”, “abused”, “abuser”, “abusers”, and so on. Due to the limited amount of lemmas and stems of words, often radical or rarely used in everyday language, for instance, “homologous” and “apostasy”, the expansion of the existing dictionary is of essential importance (see Fig. 1).

Since we are interested in lemmas instead of stems, we initially expanded the original dictionary using the WordNet [16] synsets, maintaining the lemmas that shared the same initial part with stems in the MFD. The result of this first expansion was to obtain for each MFD entry, for instance, *traitor**, a series

Table 1

Corpus size after employing the WordNet resource to expand the MFD according to the official “virtue” and “vice” categories. The initial number of words contained in the MFD is shown in parenthesis.

Moral Dimension	Virtue	Vice
Care/Harm	95 (16)	85 (35)
Fairness/Cheating	69 (26)	57 (18)
Loyalty/Betrayal	99 (29)	72 (23)
Authority/Subversion	160 (45)	101 (37)
Purity/Degradation	97 (35)	161 (55)
Total	520 (151)	476 (168)

of lemmas, for instance, *traitor**, *traitorous**, *traitorously**, *traitorousness**,⁵

We performed an initial preprocessing step on the obtained word corpus removing the forms that matched the search but did not relate to a moral trait. For example, the stem *caste** not only matches *caste** and *caste_systems**, but also *caster** and *caster_sugar**, which are clearly not related to any moral foundation. This procedure was carried out manually, considering both the gloss for the lemma provided by WordNet and the moral trait that should be attributed to that word (e.g., while it could be argued that a statesman’s name is an appropriate match for the Authority trait, the stem *church** relates to purity, and thus we ignored *Churchill**).

Following the original classification, we divided the obtained word corpus (1148 words) in “virtue” and “vice” lemmas resulting with 520 “virtues” and 476 “vices” while 152 were characterized as “general” morality words. These words can pertain to more than one traits; however, this is not common as the dataset consists of 442 unique “vice” words and 512 unique “virtue” words as shown in Table 1.

3.2. Moral valence annotation

Once the expanded dictionary was obtained, we used the Figure Eight⁶ crowdsourcing platform to annotate each lemma with an association strength to the related moral trait. The goal here is twofold. On the one hand, we can use these annotations to determine if the terms extracted during the expansion process are still related to a moral trait. On the other hand, a lexicon with ratings could be useful for better dictionary-based approaches and is a first step in the direction of moral detectors that can rank sentences, instead of merely classifying them with a binary vice/virtue rating.

The expanded dictionary was annotated in terms of moral valence, but we also collected ratings of valence and arousal, following the definitions employed for the ANEW resource [16]. For our purpose, moral valence can be represented by a bipolar scale that, in aggregate, defines a continuous dimension from one moral extremity of the MFT to the other, e.g., from Care to Harm. Moral valence was operationalized in a 9-point Likert Scale, wherein if a word was ranked in the middle of the scale, it was semantically neutral to the specific moral dimension. The annotators were presented with the description of the moral trait and were asked to rate the relevance of the word to the specific foundation; if relevant, they were asked to rate its emotional valence, arousal and then its moral valence. Each experiment presented 20 different words to the annotator. The first time the annotators participated in the rating of a specific moral dimension (e.g., Care/Harm), after the experiment, they were asked to

⁵ The letter after the number sign # indicates the part of speech for that word, i.e., #n for nouns, #a for adjectives, #r for adverbs, and #v for verbs.

⁶ The Figure Eight Platform is available here: <https://www.figure-eight.com/>.

⁴ Available at: <http://moralfoundations.org/othermaterials>.

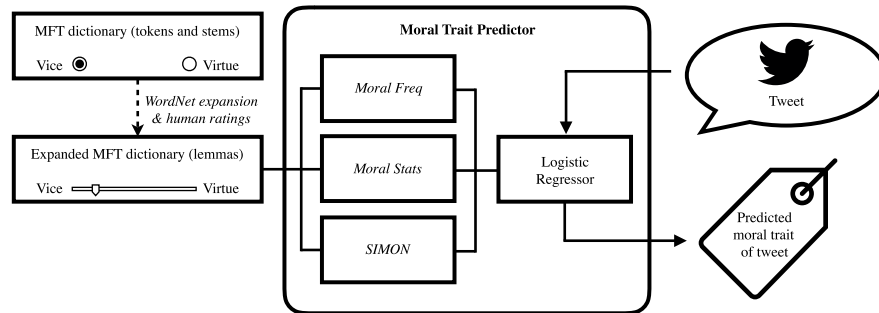


Fig. 1. Overview of the process, from dictionary expansion to moral value prediction.

fill in the Moral Foundations Questionnaire [13] for the respective dimension. At least five annotators were recruited for each lemma.

The ratings of valence and arousal were included to ensure a minimum quality of the annotation. Since no existing resource annotates moral valence on a fine scale, we used the values of valence from the subset of words that appear both in our extended dictionary and in [48]. Annotators have always been presented 4 “gold” words among the 20 words they annotate, and the annotations of those who fail more than 1 gold word are discarded. A valid answer is one that lies within 1.5 standard deviations from the valence mean of [48], for each specific gold word.

As seen, the proposed lexicon has both subjective and generative components. We need to take into consideration the subjectivity of human annotators; still, the candidate words for the annotations were chosen automatically, as a result of the expansion from a large word seed.

3.3. Moral lexicon approaches

For the generated moral lexicon, we propose the following feature extraction approaches, which can be divided into those that solely exploit the semantic information of each word, and those who exploit the moral valence associated to the word. More specifically, we propose three lexicon utilization approaches: (i) frequency counts, (ii) statistical summary, and (iii) word embedding similarity based representations. The two first approaches use both the words and their moral values, while the third one makes use solely of the selection of words, ignoring the associated numeric moral values. Fig. 2 illustrates these approaches. These three methods described above are used as feature extractors. In the next experiments, we feed these features to a logistic regression classifier. Such a simple learning algorithm is used to evaluate the performance of the proposed features, without exploiting more complex learning methods.

Moral Freq. It consists of counting the number of words that express a specific moral dimension in a binary way. To decide if a specific word expresses a moral, we apply a simple rule: if the word has its moral value lower than a certain threshold, it does not convey that moral; if higher, the word does express that moral. Given the properties of the generated moral lexicon, the threshold is set at 5. We represent a given text with a 10-dimensional vector, which contains the corresponding normalized frequency counts, each for each moral extremity; for instance, *care/harm* are represented by two dimensions, one for *care* and other for *harm*.

Moral Stats. Given a specific text, we generate a statistical summary of the moral valence distribution of the contained

words in the text. In the statistical summary, we included (i) the average, (ii) the standard deviation, (iii) the median, and (iv) the maximum value. As a result, the text is represented by a 20-dimensional vector consisting of the statistical values obtained from the lexicon annotations.

Similarity-based sentiment projection (SIMON). Finally, the third approach is known as SIMilarity-based sentiment projection (SIMON), described in [38]. This method was initially developed for sentiment analysis tasks, while here, we adapted it to moral valence assessment. *SIMON* uses a pre-trained word embedding model to compute the cosine similarity between the words of the analyzed text and a selection of domain-related words, in our case, a specific moral dimension. Projecting the analyzed text over the selection of words from *MoralStrength*, we result with a vector representation that encodes the similarity of the document to the specific moral dimension.

3.4. Data collection and preprocessing

For the evaluation of our models, we employed the Moral Foundation Twitter Corpus (MFTC) [17]. MFTC is the most extensive available corpus containing 35,108 tweets and annotations, specifically collected to assess the moral values from user-generated content. This corpus includes seven distinct datasets⁷ which were employed in scientific studies to assess the moral narratives in the user-generated text according to the moral foundations’ theory. Here we provide an overview of the datasets that are included in the MFTC and are employed in our analysis.

Hurricane Sandy (HS). The first dataset we employed is presented in [43] and originally consisted of 4191 tweets.⁸ These Tweets contain hashtags relevant to the “Hurricane Sandy”, a hurricane that caused significant damage to the Eastern seaboard of the United States in 2012. Due to Twitter regulations, the original dataset could not be fully recovered, leaving us with only 3853 messages. We further removed the retweets, keeping only the original messages, to avoid overfitting the data. In this way, the processed dataset consists of 3478 instances.

Baltimore Protest (BP). The second dataset is comprised of messages related to the 2015 Baltimore Protests, which were motivated by the death of Freddie Gray. An older version of this dataset exists which contains a more significant number of instances. Nevertheless, since the annotations of this older version are obtained by automated means [12], we have decided to use the newer version, which has manual annotations.

⁷ The full dataset is available at <https://osf.io/k5n7y/>.

⁸ This dataset can be obtained from <https://osf.io/nzx3q/>.

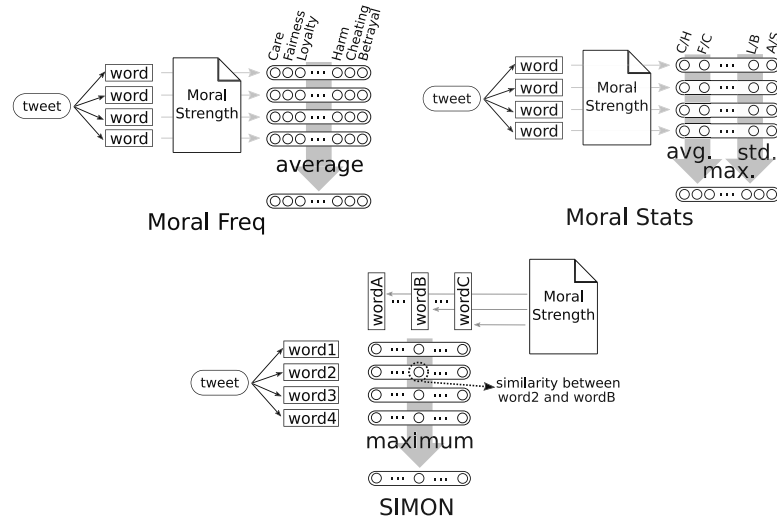


Fig. 2. Diagram of the proposed feature extraction approaches that utilize the presented lexicon.

All Lives Matter (ALM). Include #BlueLivesMatter and #AllLivesMatter hashtags and were posted between 2015–2016. These tweets were purchased from a third-party vendor.

Black Lives Matter (BLM). Posted between 2015–2016 about the Black Lives Matter Movement. Hashtags used to compile the corpus: #BLM, #BlackLivesMatter. The tweets were purchased from a third-party vendor.

2016 Presidential Election (PE). Scraped during the 2016 Presidential election season from the followers of @HillaryClinton, @realDonaldTrump, @NYTimes, @washingtonpost, and @WSJ.

Davidson (D). Taken from Davidson et al.'s [49] corpus of hate speech and offensive language.⁹

All the above datasets are annotated by experts who indicated the presence or absence of a moral foundation dimension for each tweet. Moreover, annotations include a “non-moral” label, indicating that the specific text does not reflect any moral trait. Importantly, the above datasets cover a wide variety of topics, both political and not. Topics related to politics cover left (e.g. BLM), right (e.g. ALM), and bilateral sides (e.g. Presidential Elections); while the datasets that are not unrelated to politics are expressing two controversial situations, a humanitarian call (e.g., Hurricane Sandy) and a collection of hate speech (e.g. Davidson). Such variability in the topics allows for a broader evaluation of the models’ performance, avoiding biases due to context-specific language usage.

All data were collected downloading the original tweets following the Twitter IDs provided in the MFTC [17]. Since users often delete their tweets, we only managed to recover a portion of the original datasets. More specifically, 82% of the original dataset has been recovered, and the statistics are reported in Table 2. We also report the distribution of Tweets per moral dimension per dataset. We applied some basic preprocessing to the original textual content of the tweets employing the GSITK library.¹⁰ In particular, we normalized the text converting the URLs using the special token “<url>”, usernames to “<username>”, and hashtags to the token <hashtag> and the word that is

Table 2

Statistics of the Moral Foundations Twitter Corpus employed in this study as a benchmark. All datasets were annotated by human annotators. Moreover, we note that according to the topic the distribution of the traits is varies. Last row reports the average number of training instances when using under-sampling.

Moral	Sandy	Baltimore	ALM	BLM	Elections	Davidson
Care	217	434	1314	1065	798	462
Fairness	416	292	723	940	736	133
Loyalty	410	895	408	531	286	331
Authority	155	120	274	494	177	1064
Purity	38	37	182	254	349	122
No moral	2242	2396	585	1056	2020	2846
Total	3478	4174	3486	4340	4366	4958
Under-sampling	824	1185	1162	1146	1455	1400

included in the hashtag (e.g., “#Baltimore” to “<hashtag> Baltimore”). Moreover, punctuation, symbols, and numbering were normalized.

3.5. Experimental design

To evaluate the potentials of the *MoralStrength* lexicon, we postulate the problem as a classification task. In particular, we employ the three approaches previously described, namely *Moral Freq*, *Moral Stats*, and *SIMON*, to predict the moral rhetoric in each of the aforementioned datasets (see Section 3.4).

In our experimental design, we include a basic Bag-of-Words (unigram) model providing a standardized way of obtaining a baseline in the computational linguistics field. We also report as a baseline the frequency counts employing the original MFD. As we built a series of logistic regression models; firstly, we assess the predictive power of the unigrams, *Moral Freq*, *Moral Stats*, and *SIMON* lexicon methods alone. Then, we train logistic regression models concatenating the features extracted by the above approaches.¹¹ In this way, we examine the effective performance of

⁹ The original corpus is available at <https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data>.

¹⁰ <https://github.com/gsi-upm/gsitk>.

¹¹ For replicability purposes, we have liberated the *MoralStrength* lexicon along with the implementation of the presented methods in a GitHub repository.¹²

¹² <https://github.com/oaraque/moral-foundations>.

both engineered and word embedding features in analyzing user-generated text. We also combine the unigrams to the proposed lexicon approaches described above. Hence, for each dataset and moral dimension, we train a series of logistic regression models following a 10-fold cross-validation scheme. We then report the F1-score as the evaluation metric per moral dimension since this is the once employed in the majority of the related studies.

To directly compare our proposed framework with the current state-of-the-art approach of Lin et al. [43], we replicated their same configuration. Namely, we perform over-sampling on the original dataset to overcome the highly imbalanced nature of the benchmark data (see Section 3.4). After over-sampling on the *Hurricane Sandy* data, we resulted with an average number of training examples, $N = 6128$, instead of the original dataset size, $N = 3478$ (see Table 2).

Since over-sampling implies “artificial” data samples, we propose an alternative methodology; more specifically, we performed under-sampling, which also deals with the issue of unbalanced classes, however, in doing so, it randomly excludes data points of the most populated class. In this way, for the *Hurricane Sandy* we had $N = 824$ data points (see Table 2). By reporting the score for both methods, we ensure the results are not biased by the technique used to address the class imbalance.

For all experiments, we report the performance in terms of F1-score, which is the metric also employed by Lin et al. [43], as well as the average F1-score over all moral dimensions. Moreover, to compare the improvement of the simplest model, which for this study we consider being the *Moral Freq* model, we employ the Friedman statistical test [50], which yields a ranking of the proposed method ordered by their performance. To obtain further insights on the statistical significance of our obtained results for the baseline model the Bonferroni–Dunn [50] post-hoc statistical test is performed with $\alpha = 0.05$.

4. Results and discussion

4.1. Evaluation of moral valence

After collecting the moral valence ratings, we assessed the quality of the crowdsourced data with an intrinsic evaluation. However, since the only dictionary currently available for MFT has binary annotations (vice/virtue), a direct comparison with it is not informative enough.

Hence, we evaluate the quality by (i) calculating inter-annotator agreement for the moral valence ratings, (ii) calculating the correlation between valence scores and the normative lexicon of Warriner et al. [48], and (iii) comparing binarized moral valence ratings with the gold standard given by the MFD. The results for all these tests are reported in Table 3.

To assess inter-annotator agreement we calculated Gwet’s agreement coefficient (AC2) [51]. We opted for this measure since other, more common measures (e.g., Cohen’s Kappa) require the number of annotators per element to be constant, and this is not the case for our data. Moreover, Gwet’s coefficient can be weighted, meaning that annotators expressing close ratings will positively influence the coefficient score, and negatively impacted by scores that are far apart, a sensible feature for our dataset. Results for all the traits are in the “Moderate” to “Good” range (0.4–0.8), except for fairness (which had “Poor” agreement, 0.17). While this is positive, it also indicates that the task is not trivial and that some words might be hard to rate. The lower agreement of fairness led us to inspect the agreements for all traits manually, and we discovered that some annotators were particularly inaccurate. It was thus decided to discard some annotators, despite their ability to complete the crowdsourced experiment without failing the control questions. In particular, for the Authority trait,

Table 3

Measures of the quality of the collected ratings. The first column is the inter-annotator agreement for each moral dimension via Gwet’s gamma with quadratic weighting metric. The second column is the correlation of the aggregate valence ratings and the gold standard of [48]. The last column is the agreement of the aggregate ratings (binarized) and the original Moral Foundations Dictionary, using Cohen’s Kappa.

Moral trait	Inter-annotator	Warr correlation	MFD agreement
Authority	0.42	0.84	0.78
Care	0.65	0.95	0.91
Fairness	0.34	0.88	0.84
Loyalty	0.59	0.91	0.84
Purity	0.56	0.79	0.92

the annotator with the worst agreement was removed, improving the original AC2 of 0.41 to 0.42. For loyalty, the answer of one annotator was lost due to programmatic error (the result for one word is outside the range specified by the Likert scale) and was removed from the dataset (no effect on the agreement). In the case of fairness, we intervened more drastically and removed five annotators, plus one non-valid answer. The five discarded annotators were chosen due to them having a weak agreement with other annotators, and to inconsistent ratings (i.e., they gave the same score to antonyms that have opposite traits in the MFD gold standard, such as “honest” and “dishonest”). The inter-annotator agreement for valence ranges between 0.61 and 0.72, thus falling in the “Good” category for the set of words of every moral trait. This indicates, in general, that annotating valence is easier and less controversial than rating moral traits.

We also compared the aggregated values of valence ratings (i.e., the mean of all valence annotations for a word) with the gold scores provided by [48]. In this case, we report the results of the Pearson correlation, which ranges from 0.79 to 0.95, indicating once again that the crowdsourced annotation is of good quality, and that differences between annotators are within the acceptable range.

Finally, to be able to compare with the only gold standard for moral foundations, i.e., the Moral Foundations Dictionary, we binarized the aggregated annotations and excluded those whose average is 5 (the center of the Likert scale, meaning that the word is neither positive nor negative¹³). This way, we could calculate Cohen’s kappa coefficient by comparing to the vice/virtue ratings of the MFD for the subset of words that exists in both datasets. The lowest agreement is for Authority, but also, in this case, the 0.78 value suggests that the annotations are generally reliable and entirely in line with the original MFD. It is perhaps worth noting that the agreement of fairness is quite good (0.84), despite the lower inter-annotator agreement of the collected ratings. This might indicate that, while the aggregate ratings are reliable (i.e., they fall in the correct side of the morality spectrum), there is a relatively high individual variation regarding where the words of that dimension should be placed.

4.2. Evaluation of MoralStrength lexicon

In this section, we assess the predictive power of the various approaches exploiting *MoralStrength* to analyze the moral rhetoric on the benchmark datasets described above. We confront the performance of the models against a series of baseline models. Initially, for each dataset, we report the performance of the model employing as features frequency counts from the MFD; this model shows how well the MFD alone would perform. Then,

¹³ While it would be sensible to consider neutral a range instead of a single value, e.g., excluding everything in the interval 4.5–5.5, we wanted to avoid removing more words from the comparison.

we report the performance of the predictive model employing unigrams, which provide an assessment of the difficulty of the task itself and the State-Of-The-Art (SOTA) performance for each dataset when available.

We show the performance of the logistic regression models of increasing complexity, starting from the *Moral Freq*, *Moral Stats*, and *SIMON*, followed by aggregations of the above lexicons. For all experiments, we report the Friedman statistical test [50], which yields a ranking of the proposed methods ordered by their performance. To obtain further insights on the statistical significance of our obtained results to the baseline model the Bonferroni-Dunn [50] post-hoc statistical test is performed. Note that in this study, for the statistical significance test, we employed the *Moral Freq* with MFD model as a baseline model, and not the unigram one, since it is the one that infers on the simplest generated lexicon.

For the case of Hurricane Sandy we can see that across all moral dimensions, the model inferring on the aggregated unigram and *SIMON* features emerges as the best performing approach; with a statistically significant improvement of the average F1-score – 87.6 over 62.4 reported by Lin et al. [43] (see Table 4). In this case alone, we employed over-sampling to directly compare to the previous SOTA approach on the same dataset [43]. Interestingly, the highest score is obtained for “purity”, which was reported being the most challenging moral dimension in the work of Lin et al. [43]. Examining each moral dimension separately, we note that our results are also consistently higher than the unigram model. The models that stand out are (i) unigrams + *SIMON* for fairness, loyalty, and purity, (ii) unigrams + *SIMON* + *Moral Freq* for care, purity, and neutral text, while (iii) unigrams + *SIMON* + *Moral Freq* + *Moral Stats* is the best performing models for authority and purity.

Table 5 reports the results of the evaluation for Hurricane Sandy when under-sampling is applied. Following this sampling approach, the results vary for oversampling (see Table 4), while the average overall performance slightly decreases (85.0% against 87.6% F1-score). Still, it is arguably preferable to perform under-sampling in comparison to over-sampling since in this way we avoid overfitting to the most prevalent outcome. Noteworthy is the fact that the best performing models are consistent between the over and under-sampling approaches. We also note that the importance of the statistical features regarding the moral valence of lemmas, exploited in the *Moral Freq* and *Moral Stats* lexicon methods, is more pronounced for all moral traits, for the oversampling technique. More precisely, the model inferring from unigrams together with the *Moral Stats* model, has a better performance in fairness, loyalty, and purity, while for care, the best performing model is the *SIMON* combined to the *Moral Freq* and *Moral Stats* Lexicons. Observing the obtained results, we conclude that combining lexicon-driven representations which take into consideration the moral valence, together with pure textual information (for instance, the unigrams), allows for a more robust and semantically meaningful representation. Despite the differences in the proposed approaches, comparing our approach to the study presented by Garten et al. [39], who also predicted the moral foundations on the *Hurricane Sandy* dataset, we note that their best performing model achieved 49.6% F1-Score, which is remarkably lower than the 88.2% reported here.

Next we present the performance of the models on *Baltimore Protest*, *All Lives Matter*, *Black Lives Matter*, *Davidson* and *2016 Presidential Election* datasets in Tables 4, 5, 6, 7, 8, 9 and 10, respectively. According to the previous discussion, we employ the under-sampling technique in all cases, always reporting the SOTA, as described in Hoover et al. [17].¹⁴ Carefully comparing

the experimental results, we first note that unigrams provide a reasonably good baseline for all datasets. As expected, this method is shown to be a generally stable approach, even if the training samples are few. Next, we note that when we introduce the notion of moral strength to the basic unigram approach, results are steadily improved. Table 11, provides clear evidence to this statement; there, the Friedman test indicates that the best model overall is unigrams + *Moral Freq*, followed by unigrams + *Moral Stats*. Thus, introducing knowledge about moral valence, we can better predict the moral rhetoric in text.

This result differs from the statement made by Lin et al. [43], where they show that adding the features from the original lexicon, MFD, does not improve the score. Hence, we could argue that introducing the notion of moral valence the quality of the proposed lexicon, *MoralStrength*, improves the performance in text analysis as compared to the MFD. To safely conclude to this latest argument, we present a direct comparison of all datasets in Table 12. Here, we compare the performance of the original MFD versus our *MoralStrength* for all the proposed approaches. The reported scores are averaged over all moral values per dataset. As observed, the Friedman test indicates that the *SIMON* model with the proposed lexicon outperforms the rest. Hence, it is safe to assume that the newly introduced resource offers an improvement over the previous lexicon.

Moving to the model comparison, it can be seen that combining unigrams and *SIMON* does not generally improve the results of the classification. Interestingly enough, such combination does primarily improve the metrics when done in the over-sampling case (Table 4). In light of this contrast, and considering that both the unigrams and *SIMON* approach generate a large number of features, we hypothesize that combining large feature vectors leads to overfitting. As expected, an increase of the training data quantity improves the results, leading the unigrams + *SIMON* model to obtain better results.

To conclude, we observe that the performance trends are maintained; a unigram model is a robust approach, and adding information from *MoralStrength* improves the prediction performance. Noteworthy is the fact that the expression of moral sentiment can vary substantially according to the context. Variability in the model performances may also depend on the topic of discourse; the datasets employed for the evaluation include political left (e.g., BLM), right (ALM), both ideological poles (e.g., the Presidential election). They also include topics unrelated to political discourses (e.g., Hurricane Sandy). Moreover, the variability of training samples available for each trait may explain the differences in the model performance.

We believe that exploratory analysis will be useful for the ever-increasing studies on moral foundations since it presents a variety of approaches on how the moral lexicon we propose can be employed for the prediction of moral narratives from a text.

5. Conclusions

There is an ever-increasing interest in moral values understanding since they reflect our perception, attitudes, and opinion formation on critical societal issues. Moral values are expressed in user-generated content [52], and primarily through text. With the burst of social media data, emerges a unique opportunity of observing such behaviors in scale and as they happen. Recent

¹⁴ Regarding *Baltimore Protest*, Rezapour et al. [45] reported higher accuracy with respect to Mooijman et al. [12]. Nevertheless, we cannot compare against

them for two reasons. First, the dataset is not the same; Rezapour et al. selected a subset from the original larger dataset [12] where annotations were automatically inferred by an algorithm, while our evaluation dataset originates by Hoover et al. [17], where annotations were manually assigned. Secondly, their evaluation metric is reported in terms of accuracy, while we use F1-Score as the majority of the related works. Thus, a direct comparison cannot be made.

Table 4

F1-Score of the proposed methods using over-sampling over *Hurricane Sandy* [43]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' and '†' mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	56.3	59.2	61.8	54.4	63.1	66.4	60.2	14.4
unigrams	74.0	76.9	76.5	80.7	94.1	77.2	79.9	11.9
SOTA: Lin et al [43]	82.3	70.7	50.3	69.3	37.4	64.2	62.4	12.9
<i>Moral Freq</i>	61.4	58.2	61.9	56.0	62.1	63.4	60.5	14.4
<i>Moral Stats</i>	62.8	57.2	58.8	52.7	64.1	63.3	59.8	15.1
SIMON	79.6	82.3	77.1	86.0	98.1	84.2	84.5	6.4*
SIMON + <i>Moral Freq</i>	79.2	82.5	77.2	83.8	98.2	83.9	84.1	6.8*
SIMON + <i>Moral Stats</i>	79.2	82.2	77.0	84.0	98.2	83.9	84.1	7.6
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	79.6	82.5	77.1	84.0	98.2	83.8	84.2	6.6*
unigrams + <i>Moral Freq</i>	75.3	77.7	77.2	81.2	95.5	77.8	80.8	9.7
unigrams + <i>Moral Stats</i>	73.5	77.6	76.7	81.3	95.7	77.9	80.5	10.8
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	74.0	78.2	77.1	81.7	95.9	77.9	80.8	9.4
unigrams + SIMON	84.6	85.6	81.2	90.0	98.9	85.5	87.6	2.0 †
unigrams + SIMON + <i>Moral Freq</i>	85.1	85.2	80.8	89.5	98.9	85.6	87.5	2.6*†
unigrams + SIMON + <i>Moral Stats</i>	84.9	85.4	80.4	90.0	98.8	85.2	87.5	3.3*†
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	85.0	85.4	80.8	90.2	98.9	85.2	87.6	2.1*†

Table 5

F1-Score of the proposed methods using under-sampling over *Hurricane Sandy* [43]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' marks that the approach statistically outperforms the baseline. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	56.1	54.6	61.3	50.4	59.4	66.2	58.0	15.1
unigrams	78.1	88.8	85.7	90.1	66.5	92.1	83.5	5.3*†
SOTA: Hoover et al [17]	55.0	58.0	44.0	44.0	56.0	–	51.4	13.4
<i>Moral Freq</i>	65.2	56.4	61.6	51.7	54.6	69.3	59.8	14.6
<i>Moral Stats</i>	74.1	60.8	73.4	63.3	60.4	79.6	68.6	12.6
SIMON	76.9	76.3	77.2	75.0	73.5	72.8	75.3	10.7
SIMON + <i>Moral Freq</i>	77.2	79.6	79.4	75.6	70.7	73.4	76.0	10.6
SIMON + <i>Moral Stats</i>	77.3	80.8	80.2	74.6	72.2	76.0	76.8	10.1
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	77.5	80.8	80.7	75.1	72.4	77.1	77.3	9.1
unigrams + <i>Moral Freq</i>	81.2	88.4	86.8	89.9	67.4	91.9	84.3	4.4*†
unigrams + <i>Moral Stats</i>	80.2	87.3	85.5	89.0	67.8	90.4	83.4	6.4*
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	80.0	87.3	85.6	89.1	68.2	90.2	83.4	6.2*†
unigrams + SIMON	83.8	92.0	85.2	89.1	74.7	85.0	85.0	3.3 †
unigrams + SIMON + <i>Moral Freq</i>	83.1	91.5	86.0	88.1	73.2	85.7	84.6	3.6*†
unigrams + SIMON + <i>Moral Stats</i>	82.5	90.6	84.8	86.0	72.8	88.0	84.1	5.4*†
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	82.7	90.4	85.2	85.6	72.6	88.1	84.1	5.3*†

Table 6

F1-Score of the proposed methods using under-sampling over *Baltimore Protest* [17]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' and '†' mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	59.3	64.4	64.3	50.2	54.7	64.1	59.5	14.9
unigrams	87.3	84.0	85.6	83.7	93.2	82.8	86.1	4.3*†
SOTA: Hoover et al [17]	33	47	25	25	15	–	29	13.9
<i>Moral Freq</i>	57.6	67.9	64.5	54.0	60.7	63.6	61.4	14.1
<i>Moral Stats</i>	66.5	72.5	71.0	63.7	65.6	68.1	67.9	12.3
SIMON	79.3	73.6	74.5	87.1	75.5	81.9	78.7	9.9
SIMON + <i>Moral Freq</i>	80.0	66.4	77.7	87.5	86.3	82.0	80.0	9.4
SIMON + <i>Moral Stats</i>	81.5	71.7	77.6	86.7	84.9	82.1	80.7	9.3
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	81.5	73.8	77.6	85.8	84.9	82.1	81.0	8.7
unigrams + <i>Moral Freq</i>	88.1	81.4	85.2	85.4	91.9	84.0	86.0	4.1*†
unigrams + <i>Moral Stats</i>	88.0	85.0	85.8	86.7	90.5	83.5	86.6	3.4*†
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	88.5	85.0	85.5	86.7	90.5	83.8	86.7	3.1 †
unigrams + SIMON	86.0	68.5	84.8	81.7	87.8	85.1	82.3	7.6*
unigrams + SIMON + <i>Moral Freq</i>	86.4	65.4	84.2	81.2	90.5	85.1	82.1	8.2*
unigrams + SIMON + <i>Moral Stats</i>	86.6	72.1	84.7	82.9	86.4	85.7	83.1	6.8*†
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	87.1	72.1	84.8	83.3	86.4	85.7	83.3	6.1*†

developments in the computational linguistics domain, allow us to analyze automatically such data obtaining useful insights.

Operationalizing morality via the Moral Foundations Theory (MFT) [13], we propose a linguistic resource, *MoralStrength*, that aims at improving the only currently available dictionary, i.e., the

MFD. More specifically, we contribute with a moral lexicon containing (i) a large number of lemmas, (ii) less radical and more frequently used lemmas, hence, improving its usability, and (iii) finally, containing a metric of moral valence for each lemma. *MoralStrength* contains approximately five times more lemmas

Table 7

F1-Score of the proposed methods using under-sampling over *ALM* [17]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' and '†' mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	65.9	77.7	63.8	82.8	57.5	65	68.8	14.9
unigrams	74.0	91.9	88.1	90.7	92.0	79.7	86.1	3.6 †
SOTA: Hoover et al [17]	67.0	76.0	62.0	63.0	39.0	–	61.4	14.3
<i>Moral Freq</i>	64.6	76.7	67.6	85.6	58.3	57.4	68.4	14.9
<i>Moral Stats</i>	64.4	85.4	85.3	93.8	70.6	60.3	76.6	11.6
SIMON	75.0	83.6	76.2	80.3	86.8	73.6	79.3	11.4
SIMON + <i>Moral Freq</i>	72.4	86.1	78.1	78.6	87.9	71.7	79.2	12.4
SIMON + <i>Moral Stats</i>	72.7	86.7	78.0	82.3	89.6	68.8	79.7	11.7
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	72.3	87.7	79.9	81.7	90.1	69.2	80.2	11.1
unigrams + SIMON + <i>Moral Stats</i>	74.9	90.1	84.9	85.2	93.1	74.4	83.8	6.1*
unigrams + <i>Moral Freq</i>	74.5	90.4	88.3	90.7	89.8	78.2	85.3	4.6†
unigrams + <i>Moral Stats</i>	70.8	90.9	89.1	90.7	92.8	73.2	84.6	5.9†
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	72.2	91.3	89.3	90.7	92.8	73.6	85.0	4.9†
unigrams + SIMON	75.6	88.5	86.1	87.2	90.6	73.3	83.6	6.5†
unigrams + SIMON + <i>Moral Freq</i>	74.8	88.4	83.9	86.5	90.4	73.7	82.9	8.1†
unigrams + SIMON + <i>Moral Stats</i>	74.9	90.1	84.9	85.2	93.1	74.4	83.8	6.1†
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	75.6	90.2	84.9	86.5	93.1	74.2	84.1	5.0†

Table 8

F1-Score of the proposed methods using under-sampling over *BLM* [17]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' and '†' mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	68.0	84.3	89	89.3	82.1	70.6	80.5	15.4
unigrams	85.8	93.1	90.5	93.3	93.3	81.6	89.6	3.0 †
SOTA: Hoover et al [17]	74.0	87.0	83.0	25.0	57.0	–	65.2	14.0
<i>Moral Freq</i>	66.2	86.2	88.8	92.3	80.8	70.9	80.9	13.9
<i>Moral Stats</i>	77.3	93.4	92.8	96.6	90.5	51.6	83.7	7.6*
SIMON	81.7	85.2	88.7	86.9	84.8	75.8	83.9	12.5
SIMON + <i>Moral Freq</i>	81.0	88.9	89.6	89.0	85.4	71.7	84.3	12.2
SIMON + <i>Moral Stats</i>	80.8	89.9	90.6	89.8	84.4	71.8	84.6	10.6
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	80.9	89.8	90.6	90.1	87.2	72.5	85.2	9.4
unigrams + SIMON + <i>Moral Stats</i>	83.2	90.1	90.5	90.9	87.2	75.1	86.2	7.9*
unigrams + <i>Moral Freq</i>	86.1	91.6	89.6	93.1	92.3	79.9	88.8	4.6*
unigrams + <i>Moral Stats</i>	79.3	92.0	89.8	92.2	93.7	73.3	86.7	7.4†
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	80.3	92.3	89.8	92.2	94.3	73.3	87.0	6.8*
unigrams + SIMON	88.2	91.1	90.9	89.6	84.8	79.4	87.3	6.1†
unigrams + SIMON + <i>Moral Freq</i>	87.7	88.7	90.5	90.1	90.3	76.4	87.3	6.7†
unigrams + SIMON + <i>Moral Stats</i>	83.2	90.1	90.5	90.9	87.2	75.1	86.2	7.9*
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	84.0	91.4	90.5	90.7	87.6	75.5	86.6	6.9*

Table 9

F1-Score of the proposed methods using under-sampling over *Davidson* [17]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. '*' and '†' mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	36.5	33.3	39.4	36.1	38.7	38.0	37.0	14.8
unigrams	84.6	91.3	86.6	77.3	92.3	56.5	81.4	4.4 †
SOTA: Hoover et al [17]	7.0	5.0	2.0	2.0	5.0	–	4.2	13.9
<i>Moral Freq</i>	39.6	33.3	39.9	39.1	37.9	39.6	38.2	14.3
<i>Moral Stats</i>	42.4	39.0	43.0	43.1	37.9	42.7	41.3	13.4
SIMON	84.8	85.8	87.2	72.0	89.3	53.8	78.8	11.2
SIMON + <i>Moral Freq</i>	85.4	85.8	87.7	71.1	89.8	54.3	79.0	9.4
SIMON + <i>Moral Stats</i>	85.7	85.8	88.4	72.4	89.8	54.5	79.4	7.6*
SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	85.7	85.8	88.4	72.4	89.8	54.4	79.4	7.8
unigrams + <i>Moral Freq</i>	84.6	90.6	87.6	76.2	92.3	56.0	81.2	4.9†
unigrams + <i>Moral Stats</i>	85.4	91.3	87.4	77.0	90.6	55.8	81.2	4.9†
unigrams + <i>Moral Freq</i> + <i>Moral Stats</i>	85.4	91.3	87.4	77.0	90.6	55.7	81.2	5.1†
unigrams + SIMON	83.7	86.5	88.0	74.8	91.5	54.2	79.8	8.1
unigrams + SIMON + <i>Moral Freq</i>	85.3	86.5	88.0	75.2	92.8	54.9	80.5	6.0†
unigrams + SIMON + <i>Moral Stats</i>	86.3	87.2	87.6	75.4	91.9	55.2	80.6	5.2†
unigrams + SIMON + <i>Moral Freq</i> + <i>Moral Stats</i>	86.4	87.2	87.6	75.5	91.9	55.0	80.6	5.1†

than the MFD, while at the same time providing with a moral valence, i.e., a quantitative assessment to characterize the lemmas' relationship with each moral dimension.

To explore the potentials of the moral lexicon in predicting the moral narrative in an unseen text, we generated three representations employing a series of feature extraction

Table 10

F1-Score of the proposed methods using under-sampling over *Election* [17]. C/H: Care/Harm, F/C: Fairness/Cheating, L/B: Loyalty/Betrayal, A/S: Authority/Subversion, P/D: Purity/Degradation, NM: Non-moral, Avg.: Average. "*" and "†" mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	C/H	F/C	L/B	A/S	P/D	NM	Avg.	Rank
Baseline: Frequency MFD	69.1	76.8	63.6	80.4	59.7	63.3	68.8	13.9
unigrams	86.6	96.2	90.7	87.0	93.1	63.2	86.1	6.1*†
SOTA	64.0	79.0	41.0	41.0	49.0	–	54.8	13.4
<i>Moral Freq</i>	68.2	70.7	66.1	83.6	59.8	59.0	67.9	14.1
<i>Moral Stats</i>	79.9	80.1	78.1	92.6	74.0	57.7	77.1	10.9
<i>SIMON</i>	82.2	78.7	83.0	72.5	64.8	69.0	75.0	11.9
<i>SIMON + Moral Freq</i>	81.5	82.6	80.9	74.0	68.0	69.6	76.1	10.9
<i>SIMON + Moral Stats</i>	81.5	83.4	83.7	74.2	64.9	68.9	76.1	11.0
<i>SIMON + Moral Freq + Moral Stats</i>	81.9	83.5	84.4	77.9	66.2	69.2	77.2	9.6
unigrams + <i>Moral Freq</i>	88.4	95.8	92.3	91.8	91.1	67.3	87.8	4.0*
unigrams + <i>Moral Stats</i>	88.1	94.4	92.6	93.2	92.7	65.8	87.8	4.1*
unigrams + <i>Moral Freq + Moral Stats</i>	88.5	95.1	92.5	93.2	93.0	66.4	88.1	3.1 *†
unigrams + <i>SIMON</i>	88.1	93.6	92.3	85.9	81.4	70.9	85.4	5.6*†
unigrams + <i>SIMON + Moral Freq</i>	88.2	91.6	91.6	86.4	80.7	71.6	85.0	6.3*†
unigrams + <i>SIMON + Moral Stats</i>	87.4	93.1	92.1	88.1	82.7	70.1	85.6	5.9*†
unigrams + <i>SIMON + Moral Freq + Moral Stats</i>	87.4	93.1	91.8	88.4	83.1	70.2	85.7	5.4*†

Table 11

F1-Score of the proposed methods using under-sampling over all datasets. HS: Hurricane Sandy, BP: Baltimore Protest, ALM: All Lives Matter, BLM: Black Lives Matter, D: Davidson, PE: Presidential Election. "*" and "†" mark that the approach significantly outperforms the MFD baseline and the SOTA, respectively. The model with the lowest rank is the one that outperforms the rest.

Approach	HS	BP	ALM	BLM	D	PE	Rank
Baseline: Frequency MFD	58.0	86.1	80.5	80.5	37.0	68.8	13
unigrams	83.5	67.9	89.6	89.6	81.4	86.1	4.8*†
SOTA	51.4	86.6	65.2	65.2	4.2	54.8	13.8
<i>Moral Freq</i>	59.8	78.7	80.9	80.9	38.2	67.9	14.3
<i>Moral Stats</i>	68.6	80.0	83.7	83.7	41.3	77.1	12.7
<i>SIMON</i>	75.3	80.7	83.9	83.9	78.8	75	12.3
<i>SIMON + Moral Freq</i>	76.0	81.0	84.3	84.3	79.0	76.1	11.3
<i>SIMON + Moral Stats</i>	76.8	83.1	84.6	84.6	79.4	76.1	9.9
<i>SIMON + Moral Freq + Moral Stats</i>	77.3	86.0	85.2	85.2	79.4	77.2	8.4
unigrams + <i>Moral Freq</i>	84.3	86.6	88.8	88.8	81.2	87.8	2.5 *†
unigrams + <i>Moral Stats</i>	83.4	86.7	86.7	86.7	81.2	87.8	4.3*†
unigrams + <i>Moral Freq + Moral Stats</i>	83.4	82.3	87	87	81.2	88.1	5.3*†
unigrams + <i>SIMON</i>	85.0	82.1	87.3	87.3	79.8	85.4	5.7†
unigrams + <i>SIMON + Moral Freq</i>	84.6	83.1	87.3	87.3	80.5	85	5.4†
unigrams + <i>SIMON + Moral Stats</i>	84.1	83.3	86.2	86.2	80.6	85.6	6.4
unigrams + <i>SIMON + Moral Freq + Moral Stats</i>	84.1	83.3	86.6	86.6	80.6	85.7	5.9†

Table 12

Average F1-Score of the proposed baselines using under-sampling over all datasets. "*" marks that the approach statistically outperforms the *Moral Freq* with the MFD lexicon baseline. The model with the lowest rank is the one that outperforms the rest.

	Approach	HS	BP	ALM	BLM	D	PE	Rank
MFD	<i>Moral Freq</i>	58.0	59.5	68.8	80.5	37.0	68.8	5.5
	<i>Moral Stats</i>	64.8	67.3	74.1	80.2	39.2	75.6	4.0
	<i>SIMON</i>	74.0	79.6	79.1	82.9	78.4	74.0	2.3*
MoralStrength	<i>Moral Freq</i>	59.8	61.4	68.4	80.9	38.2	67.9	5.2
	<i>Moral Stats</i>	68.6	67.9	76.6	83.7	41.3	77.1	2.5*
	<i>SIMON</i>	75.3	78.7	79.3	83.9	78.8	75.0	1.5 *

techniques, including normalized lemmas frequencies, statistical features, and finally, semantic similarity based on word embeddings. We evaluated the machine learning framework on six benchmark datasets from the Twitter platform, the only available resources of linguistic data explicitly annotated for their moral content.

Interestingly, all our models improve the prediction performance with respect to the current state-of-the-art for all moral dimensions. The most prominent approaches – as indicated by the Friedman ranking – combine pure textual (e.g., unigrams) with lexicon-based representations (e.g., the *Moral Freq*, the *Moral Stats*, and the *SIMON*). Hence, we argue that moral lexicon can be

successfully employed for moral values classification from a given text since when this information is considered, the models yield higher performance.

This study paves the way for further advancements in the moral text analysis, which is indeed an exciting field of study, both from the computational linguistics and the social sciences points of view. From a linguistic perspective, it would be interesting to explore how specific knowledge could be encoded in domain-oriented word vectors, allowing for the development of complex learning methods. Moreover, the word embedding representation based on moral similarity could be enhanced with the obtained assessments of moral valence, or even combined with sentiment features from the analyzed text. As for the social sciences, there are numerous issues where detecting the morals narrative can significantly improve our understanding of the peoples' dispositions in, for instance, controversial social phenomena as well as the evolution of opinions over time.

Acknowledgments

Oscar Araque has been partially funded by the Spanish Ministry of Economy through the project Semola (TEC2015-68284-R). Kyriaki Kalimeri acknowledges support from the "Lagrange Project" of the ISI Foundation funded by the Fondazione CRT.

References

- [1] E. Sagi, M. Dehghani, Moral rhetoric in Twitter: A case study of the US Federal Shutdown of 2013, in: *Proceedings of the 35th Annual Meeting of the Cognitive Science Society (CogSci)*, vol. 36, 2014, pp. 1347–1352.
- [2] C. Wolsko, H. Aniceaga, J. Seiden, Red, white, and blue enough to be green: Effects of moral framing on climate change attitudes and conservation behaviors, *J. Exp. Soc. Psychol.* 65 (2016) 7–19.
- [3] M. Low, M.G.L. Wui, Moral foundations and attitudes towards the poor, *Current Psychol.* 35 (2016) 650–656.
- [4] Y. Mejova, K. Kalimeri, Effect of values and technology use on exercise: Implications for personalized behavior change interventions, in: *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, ACM, 2019, pp. 36–45.
- [5] A.B. Amin, R.A. Bednarczyk, C.E. Ray, K.J. Melchiori, J. Graham, J.R. Huntsinger, S.B. Omer, Association of moral values with vaccine hesitancy, *Nat. Hum. Behav.* 1 (2017) 873–880.
- [6] K. Kalimeri, M.G. Beiró, A. Urbinati, A. Bonanomi, A. Rosina, C. Cattuto, Human values and attitudes towards vaccination in social media, in: *Companion Proceedings of the 2019 World Wide Web Conference*, ACM, 2019, pp. 248–254.
- [7] A. Miles, S. Vaisey, Morality and politics: Comparing alternate theories, *Soc. Sci. Res.* 53 (2015) 252–269.
- [8] T. Grover, E. Bayraktaroglu, G. Mark, E.H.R. Rho, Moral and affective differences in us immigration policy debate on twitter, in: *Computer Supported Cooperative Work, CSCW*, 2019, pp. 1–39.
- [9] M. Alizadeh, I. Weber, C. Cioffi-Revilla, S. Fortunato, M. Macy, Psychology and morality of political extremists: evidence from twitter language analysis of alt-right and antifa, *EPJ Data Sci.* 8 (1) (2019) 17.
- [10] L.J. Skitka, G.S. Morgan, The social and political implications of moral conviction, *Political Psychol.* 35 (2014) 95–110.
- [11] M.L. Vecina, D. Marzana, M. Paruzel-Czachura, Connections between moral psychology and intimate partner violence: Can ipv be read through moral psychology?, *Aggress. Violent Behav.* 22 (2015) 120–127.
- [12] M. Mooijman, J. Hoover, Y. Lin, H. Ji, M. Dehghani, Moralization in social networks and the emergence of violence during protests, *Nat. Hum. Behav.* 2 (6) (2018) 389–396.
- [13] J. Graham, J. Haidt, B.A. Nosek, Liberals and conservatives rely on different sets of moral foundations, *J. Personal. Soc. Psychol.* 96 (5) (2009) 1029.
- [14] J. Haidt, J. Graham, When morality opposes justice: Conservatives have moral intuitions that liberals may not recognize, *Soc. Justice Res.* 20 (1) (2007) 98–116.
- [15] J. Haidt, C. Joseph, Intuitive ethics: How innately prepared intuitions generate culturally variable virtues, *Daedalus* 133 (4) (2004) 55–66.
- [16] G.A. Miller, Wordnet: A lexical database for english, *Commun. ACM* 38 (11) (1995) 39–41.
- [17] J. Hoover, G. Portillo-Wightman, L. Yeh, S. Havaladar, A.M. Davani, Y. Lin, B. Kennedy, M. Atari, Z. Kamel, M. Mendlen, et al., Moral foundations twitter corpus: A collection of 35k tweets annotated for moral sentiment, 2019.
- [18] P. Stone, D. Dunphy, M. Smith, *The General Inquirer: A Computer Approach To Content Analysis*, MIT Press, 1966.
- [19] P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2002, pp. 417–424.
- [20] A. Muhammad, N. Wiratunga, R. Lothian, Contextual sentiment analysis for social media genres, *Knowl.-Based Syst.* 108 (2016) 92–101.
- [21] E. Sulis, D.I.H. Farias, P. Rosso, V. Patti, G. Ruffo, Figurative messages and affect in twitter: Differences between# irony, # sarcasm and# not, *Knowledge-Based Syst.* 108 (2016) 132–143.
- [22] C. Strapparava, R. Mihalcea, Learning to identify emotions in text, in: *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC)*, Fortaleza, Ceara, Brazil, 2008, pp. 1556–1560.
- [23] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, Cambridge, UK, 2015.
- [24] S. Poria, A. Gelbukh, E. Cambria, A. Hussain, G.-B. Huang, Emotionspace: A novel framework for affective common-sense reasoning, *Knowl.-Based Syst.* 69 (2014) 108–123.
- [25] O. Araque, L. Gatti, J. Staiano, M. Guerini, Depechemood++: a bilingual emotion lexicon built through simple yet powerful techniques, *IEEE Trans. Affect. Comput.* (2019) 1, <http://dx.doi.org/10.1109/TAFFC.2019.2934444>.
- [26] H.A. Schwartz, J.C. Eichstaedt, M.L. Kern, L. Dziurzynski, S.M. Ramones, M. Agrawal, A. Shah, M. Kosinski, D. Stillwell, M.E. Seligman, et al., Personality, gender, and age in the language of social media: the open-vocabulary approach, *PLoS One* 8 (9) (2013) e73791.
- [27] T. Yarkoni, Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers, *J. Res. Pers.* 44 (3) (2010) 363–373.
- [28] R.L. Boyd, S.R. Wilson, J.W. Pennebaker, M. Kosinski, D.J. Stillwell, R. Mihalcea, Values in words: Using language to evaluate and understand personal values, in: *Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM)*, Oxford, UK, 2015, pp. 31–40.
- [29] J. Chen, G. Hsieh, J.U. Mahmud, J. Nichols, Understanding individuals' personal values from social media word use, in: *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, ACM, 2014, pp. 405–414.
- [30] D. McAdams, J. Pals, A new big five: Fundamental principles for an integrative science of personality, *Am. Psychol.* 61 (3) (2006) 204–217.
- [31] Y.R. Tausczik, J.W. Pennebaker, The psychological meaning of words: LIWC and computerized text analysis methods, *J. Lang. Soc. Psychol.* 29 (1) (2010) 24–54.
- [32] S. Clifford, J. Jerit, How words do the work of politics: Moral foundations theory and the debate over stem cell research, *J. Politics* 75 (3) (2013) 659–671.
- [33] L. Teernstra, P. van der Putten, L. Noordegraaf-Eelens, F. Verbeek, The morality machine: tracking moral values in tweets, in: *Proceedings of the 15th International Symposium on Intelligent Data Analysis (IDA)*, Stockholm, Sweden, 2016, pp. 26–37.
- [34] M. Dehghani, K. Sagae, S. Sachdeva, J. Gratch, Analyzing political rhetoric in conservative and liberal weblogs related to the construction of the “Ground Zero Mosque”, *J. Inf. Technol. Politics* 11 (1) (2014) 1–14.
- [35] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, *J. Mach. Learn. Res.* 3 (2003) 993–1022.
- [36] E. Sagi, M. Dehghani, Measuring moral rhetoric in text, *Soc. Sci. Comput. Rev.* 32 (2014) 132–144.
- [37] R. Kaur, K. Sasahara, Quantifying moral foundations from various topics on Twitter conversations, in: *Proceedings of the IEEE International Conference on Big Data (BigData)*, Washington D.C. USA, 2016, pp. 2505–2512.
- [38] O. Araque, G. Zhu, C.A. Iglesias, A semantic similarity-based perspective of affect lexicons for sentiment analysis, *Knowl.-Based Syst.* 165 (2018) 346–359.
- [39] J. Garten, R. Boghrati, J. Hoover, K.M. Johnson, M. Dehghani, Morality between the lines: Detecting moral sentiment in text, in: *Proceedings of the IJCAI 2016 Workshop on Computational Modeling of Attitudes (WCMA)*, New York, NY, USA, 2016.
- [40] J. Garten, J. Hoover, K.M. Johnson, R. Boghrati, C. Iskiwitsch, M. Dehghani, Dictionaries and distributions: Combining expert knowledge and large scale textual data content analysis, *Behav. Res. Methods* 50 (1) (2018) 344–361.
- [41] J. Hoover, K. Johnson, R. Boghrati, J. Graham, M. Dehghani, Moral framing and charitable donation: Integrating exploratory social media analyses and confirmatory experimentation, *Collabra: Psychol.* 4 (1) (2018).
- [42] J. Garten, B. Kennedy, J. Hoover, K. Sagae, M. Dehghani, Incorporating demographic embeddings into language understanding, *Cogn. Sci.* 43 (1) (2019).
- [43] Y. Lin, J. Hoover, G. Portillo-Wightman, C. Park, M. Dehghani, H. Ji, Acquiring background knowledge to improve moral value prediction, in: *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Barcelona, Spain, 2018, pp. 552–559.
- [44] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [45] R. Rezapour, S.H. Shah, J. Diesner, Enhancing the measurement of social effects by capturing morality, in: *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2019, pp. 35–45.
- [46] J. Graham, B.A. Nosek, J. Haidt, R. Iyer, S. Koleva, P.H. Ditto, Mapping the moral domain, *J. Personal. Soc. Psychol.* 101 (2) (2011) 366–385.
- [47] J.W. Pennebaker, The secret life of pronouns, *New Scientist* 211 (2828) (2011) 42–45.
- [48] A.B. Warriner, V. Kuperman, M. Brysbaert, Norms of valence, arousal dominance for 13,915 English lemmas, *Behav. Res. Methods* 45 (4) (2013) 1191–1207.
- [49] T. Davidson, D. Warnsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: *Eleventh International AAAI Conference on Web and Social Media*, 2017.
- [50] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (Jan) (2006) 1–30.
- [51] K.L. Gwet, *Handbook of Inter-rater Reliability: the Definitive Guide to Measuring the Extent of Agreement Among Raters*, Advanced Analytics, LLC, Gaithersburg, MD, USA, 2014.
- [52] K. Kalimeri, M.G. Beiró, M. Delfino, R. Raleigh, C. Cattuto, Predicting demographics, moral foundations, and human values from digital behaviours, *Comput. Hum. Behav.* 92 (2019) 428–445.

3.2.3 A Semantic Similarity-based Perspective of Affect Lexicons for Sentiment Analysis

Title	A Semantic Similarity-based Perspective of Affect Lexicons for Sentiment Analysis
Authors	Araque, Oscar and Zhu, Ganggao and Iglesias, Carlos A.
Journal	Knowledge-Based Systems
Impact factor	JCR 2019 5.921
Scopus citations	25 (02/09/2020)
Web of Science citations	14 (02/09/2020)
Google Scholar citations	34 (02/09/2020)
ISSN	0950-7051
Volume	165
Year	2019
Keywords	Sentiment analysis, Semantic similarity, Sentiment lexicon, Word embeddings
Pages	346–359
DOI	https://doi.org/10.1016/j.knosys.2018.12.005
Online	http://www.sciencedirect.com/science/article/pii/S0950705118305926
Abstract	Lexical resources are widely popular in the field of Sentiment Analysis, as they represent a resource that directly encodes sentimental knowledge. Usually sentiment lexica are used for polarity estimation through the matching of words contained in a text and their associated lexicon sentiment polarities. Nevertheless, such resources have limitations in vocabulary coverage and domain adaptation. Besides, many recent techniques exploit the concept of distributed semantics, normally through word embeddings. In this work, a semantic similarity metric is computed between text words and lexica vocabulary. Using this metric, this paper proposes a sentiment classification model that uses the semantic similarity measure in combination with embedding representations. In order to assess the effectiveness of this model, we perform an extensive evaluation. Experiments show that the proposed method can improve Sentiment Analysis performance over a strong baseline, being this improvement statistically significant. Finally, some characteristics of the proposed technique are studied, showing that the selection of lexicon words has an effect in cross-dataset performance.



A semantic similarity-based perspective of affect lexicons for sentiment analysis

Oscar Araque*, Ganggao Zhu, Carlos A. Iglesias

Intelligent Systems Group, Universidad Politécnica de Madrid, Avenida Complutense, 30, Madrid, Spain

ARTICLE INFO

Article history:

Received 11 July 2018

Received in revised form 3 December 2018

Accepted 4 December 2018

Available online 8 December 2018

Keywords:

Sentiment analysis

Sentiment lexicon

Semantic similarity

Word embeddings

ABSTRACT

Lexical resources are widely popular in the field of Sentiment Analysis, as they represent a resource that directly encodes sentimental knowledge. Usually sentiment lexica are used for polarity estimation through the matching of words contained in a text and their associated lexicon sentiment polarities. Nevertheless, such resources have limitations in vocabulary coverage and domain adaptation. Besides, many recent techniques exploit the concept of distributed semantics, normally through word embeddings. In this work, a semantic similarity metric is computed between text words and lexica vocabulary. Using this metric, this paper proposes a sentiment classification model that uses the semantic similarity measure in combination with embedding representations. In order to assess the effectiveness of this model, we perform an extensive evaluation. Experiments show that the proposed method can improve Sentiment Analysis performance over a strong baseline, being this improvement statistically significant. Finally, some characteristics of the proposed technique are studied, showing that the selection of lexicon words has an effect in cross-dataset performance.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The consistent growth of users and user-generated content in many web sites, social networks, and online consumer platforms such as Twitter, Amazon, and Yelp has incremented the quantity of opinionated information available in the internet. Such content has inherent value for a number of online and offline businesses, since online opinions can directly affect future sales in a wide range of fields such as hotels, e-commerce, restaurants, and electronics [1, 2]. Consequently, interest towards opinion mining techniques as a way of automatically extracting and analyzing user-generated opinions has risen [3]. In this context, Sentiment Analysis (SA) plays a key role.

Sentiment Analysis centers around the classification of sentiments, opinions or attitudes expressed in human-generated texts. For this end, text can be labeled into several categories, being *positive* and *negative* the most common. Since the study of opinions in text comprises a broad range of possibilities, SA can be classified into three categories, attending to the granularity level [3]. Literature considers the following categorization, in increasing order of specificity: *document level*, as in a movie review that can span one or more paragraphs; *sentence level*, where sentiment is bounded to a singular sentence of text; and *aspect level*, in which

case sentiment is associated with a word or small group of words (e.g., *good food*).

Due to the subjective nature of Sentiment Analysis, it is of no surprise that a key indicator of sentiment polarity are sentiment words. *Good*, *great* and *brilliant* can be considered positive words while *bad*, *worse* and *disastrous* can express negative attitudes. Consequently, sentiment lexicons, which gather sentiment words, are used extensively by the research community [4]. Sentiment lexicons can be organized in three types, attending to which information is contained in them [5,6]: (i) those who contain only sentiment words (a list of words), (ii) the ones that are formed by both sentiment words and polarity orientations (a list of words with only positive and negative annotations), and (iii) the lexicons that offer sentiment words with orientation and intensity [7] (a list of words with scalar numerical values).

The most popular approach that makes use of sentiment lexicons is keyword matching, also called keyword spotting [8]. Basically, this technique consists in detecting the presence of certain sentiment bearer words, thus obtaining the sentiment estimation as an aggregate of the associated sentiment values. Although this method is certainly simple and computationally cheap, it is also limited, as it happens in the case of domain adaptation. Sentiment words can display variations in their polarity values depending on the contextual domain [9], language [10] or even context [11], causing lexicon-based approaches to decrease their classification performance.

An additional challenge that arises in relation to the use of sentiment lexicons is the selection between the variety of these

* Corresponding author.

E-mail addresses: o.araque@upm.es (O. Araque), gzh@dit.upm.es (G. Zhu), carlosangel.iglesias@upm.es (C.A. Iglesias).

<https://doi.org/10.1016/j.knosys.2018.12.005>

0950-7051/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

resources. It is still an open question how to select and process sentiment lexicons [12], and how this selection affects the performance of sentiment classification tasks.

Furthermore, distributed representations, or word embeddings [13,14], are considered a key characteristic of state-of-the-art Sentiment Analysis systems. These techniques encode text into fixed-length vectors that can be used directly by machine-learning methods, being neural networks the most used recently [15]. The use of (deep) neural networks enables the system to learn complex features automatically extracted from data, minimizing manual domain-oriented efforts [16,17]. Nevertheless, a requirement of these approaches is the use of large amounts of data, normally annotated [13]. Efficiently training neural networks in small datasets is still an open challenge, whose resolutions could improve SA systems.

In light of such trends, it is reasonable to think that in order to improve sentiment analysis systems, lexicon exploitation methods should be improved, and that embedding models can be used to overcome lexicon limitations. This work addresses these challenges by proposing a method that exploits the similarity of sentiment lexicon words to input text. Particularly, the aforementioned approach makes use of the lexicon as a space to which analyzed text is projected, effectively representing text by how similar its component words are to lexicon words. Also, given that this method is based on the similarity measure, the proposed approach is not limited to the use of embedding models, but it can incorporate word to word similarity from additional sources, such as taxonomy sources like WordNet [18].

In this paper, we propose a semantic similarity based, sentiment classification model, which makes use of both sentiment lexicons and semantic models (e.g., embedding models, WordNet). Instead of keyword matching, features from text are extracted by computing the semantic similarity between input words and lexicon words. In this way, a lexicon is considered as a set of target words that can be used as a projective space, computing its similarity (semantic distance) with the input text. Besides these similarity-based features, our proposal is complemented with word embedding representations of the text. Both semantic distance and embeddings-based features are used in conjunction to feed a machine-learning algorithm with the objective of sentiment classification.

With the aim of studying the effectiveness of the proposed system and features, the evaluation makes use of seven public datasets, from both Twitter¹ and movie review domains. Moreover, several statistical studies are performed with the aim of characterizing the improvement of using semantic similarity features. Finally, we study how some characteristics of sentiment lexicons affect the system's performance.

With these proposals, we seek answer to the following research questions:

- Q1. Are the proposed semantic distance features more effective than word-matching approaches?
- Q2. How do embedding and taxonomy based similarity measures compare in terms of performance?
- Q3. How does lexicon characteristics affect the proposed feature extraction process?

The rest of the paper is structured as follows. Section 2 summarizes the previous work regarding semantic similarity and sentiment lexica. In Section 3, the proposed sentiment classification model is described. Following, in Section 4, we describe the experimental setup, aimed at empirically evaluating the proposed model, as well as these experimental results and discussion. Finally, the paper concludes with Section 5, that depicts the conclusions drawn from the evaluation and outlines the possible future lines of work.

2. Related work

2.1. Overview of semantic similarity

Semantic similarity methods give numerical similarity scores to words in order to represent their semantic distance. In computational linguistics, semantic relatedness is inverse of semantic distance and assumes that two objects are semantically related if they have any kind of semantic relation [19]. Semantic similarity is a special metric that represents the commonality of two concepts relying on their hierarchical relations [20]. In general, semantic similarity is a special case of semantic relatedness [20] which is a more general concept and does not necessarily rely on hierarchical relations. This work covers both semantic similarity and semantic relatedness. For convenience we call them semantic similarity interchangeably in the following sections and categorize them into corpus-based methods and knowledge-based methods [21]. Corpus-based methods mainly rely on contextual information of words appearing in the corpus, thus they mainly measure general semantic relatedness between words. Knowledge-based methods derive semantic similarity of words based on hierarchical relations encoded in WordNet. Corpus-based methods have wider computational applications because they consider all kinds of semantic relations between words, while knowledge-based methods would be more useful when applications need to encode hierarchical relations between words. We review both types of methods in the following sections.

2.1.1. Corpus-based methods

Corpus-based semantic similarity methods are based on word associations learned from large text collections following the distributional hypothesis [20]. Two words are assumed to be more similar if their surrounding contexts are more similar or they appear together more frequently. The computation of corpus-based methods is based on statistics of word distributions or word co-occurrences. According to different computational models, there are count-based methods, e.g. Pointwise Mutual Information [22] or Normalized Google Distance [23], and predictive methods, e.g. Word2Vec [13]. Count-based methods count word co-occurrences and construct a word-word matrix, in which those co-occurrence statistics are directly applied with probabilistic models [22], matrix factorization [24] and dimension reduction [25]. Predictive-based methods directly learn dense vectors through predicting a word from its surrounding context. We use the predictive-based word embedding tool Word2Vec [13] to learn dense vector representation of words, because it has been reported to have good performance in many applications [26]. As suggested by the Word2Vec authors [13], the Continuous Bag of Words (CBOW) model is more computationally efficient and suitable for larger corpus than the skip-gram model. Thus, the CBOW model is used to train word vectors in a neural network architecture which consists of an input layer, a projection layer, and an output layer to predict a word given its surrounding words with a certain context window size. Formally, given a sequence of training words $\{w_1, w_2, \dots, w_T\}$, each word vector is trained to maximize the average log probability:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq k \leq c, k \neq 0} \log p(w_t | w_{t-k}, \dots, w_{t+k}) \quad (1)$$

where k is the context window size and $p(w_t | w_{t-k}, \dots, w_{t+k})$ is the hierarchical softmax of the word vectors [13]. Having the trained word vectors (the dimension is predefined empirically and we set 300 in our experiments), word similarity is computed using standard cosine similarity. Although the training process relies on a neural network based supervised prediction model, the real

¹ <https://twitter.com/>. Accessed 13 June 2018.

training results are the vector representation of words instead of the neural network prediction model. Because of such idea, the training of word embedding is unsupervised and can be applied in various textual corpus without labeled datasets. Furthermore, due to the simple neural network architecture and the use of hierarchical softmax, Word2Vec is able to address large corpus and the training is very efficient. However, since the training of word vectors only use word sequences, a wide variety of word relations are considered as equally related according to their co-occurrences, which makes the similarity between trained word vectors coarse and unable to address synonymous words and hierarchical relations accurately. In consequence, knowledge-based semantic similarity methods are considered to enrich some commonsense knowledge of words.

2.1.2. Knowledge-based methods

Knowledge-based semantic similarity methods measure the semantic similarity between words based on an ontology. Two words are considered to be more similar if they are located closer in the given ontology. The lexical database WordNet [18] is used as background ontology, which is organized through synsets, being each synset a set of words sharing one common sense (synonyms). The hierarchical relations between synsets (i.e. hypernym and hyponymy), organize WordNet into a concept taxonomy. Having synonymous words in synsets and human defined hierarchical relations, knowledge-based semantic similarity methods are designed to encode this information to improve semantic similarity between words. Those semantic similarity methods are directly used for synsets rather than words, therefore, they need to be converted into word similarity by taking the maximal similarity score over all the synsets which are the senses of the words [27,28]. This is based on the intuition that human would pay more attention to word similarities (i.e., most related senses) rather than their differences [28], which has been demonstrated in psychological studies [29]. As polysemous words can be mapped to a set of synsets, let $s(w)$ denote a set of synsets that are senses of word w , then word similarity is defined as:

$$sim_{word}(w_i, w_j) = \max_{c_i \in s(w_i), c_j \in s(w_j)} sim_{synset}(c_i, c_j) \quad (2)$$

where sim_{synset} can be any semantic similarity methods used for WordNet synsets which are presented in the following of this section.

Many knowledge-based methods have been proposed in the literature [30] for measuring similarity in WordNet exploiting various information such as shortest path length, depth, and Information Content (IC). The basic idea is counting the number of nodes or edges (shortest path) between two concepts (synsets) in WordNet. Two concepts are assumed to be more similar if they are closer to each other in WordNet. Let $path(c_i, c_j)$ be the shortest path length between c_i and c_j , the Path [31] method defines semantic similarity method as:

$$sim_{Path}(c_i, c_j) = \frac{1}{1 + path(c_i, c_j)} \quad (3)$$

Another common information used to compute semantic similarity is depth, which is defined as shortest path length between root concept and a given concept through hierarchical relations. The intuition behind depth is that the upper-level concepts in a taxonomy are supposed to be more general. Thus, the similarity between lower-level concepts should be considered more similar than those concepts between upper-level concepts. This method described in [32] measures the semantic similarity between concepts based on concept depth in a taxonomy, and a special concept Least Common Subsume (LCS), which is the most specific ancestor

concept shared by two concepts. Let $LCS(c_i, c_j)$ be the LCS of concepts c_i and c_j , the method of [32] measures semantic similarity of given concepts using the following formula:

$$sim_{Wu\&Palmer}(c_i, c_j) = \frac{2depth(LCS(c_i, c_j))}{depth(c_i) + depth(c_j)} \quad (4)$$

where $depth(c_i)$ computes the $path(c_{root}, c_i)$ given c_{root} is the root concept of the taxonomy. The two similarity methods described above consider the structural knowledge of a taxonomy which have a common drawback of uniform distance between concepts. Some methods consider IC to overcome the uniform distance drawback. The IC is defined as the probability of encountering the concept in a corpus $IC(c_i) = -\log Prob(c_i)$. Note that Brown Corpus [33] is used to compute IC because words in BC are annotated with WordNet concepts. The method described in [27] only considers the IC of LCS concept, while the consequent works by [34] and [35] extend the IC-based method by including the IC of concepts.

$$sim_{Resnik}(c_i, c_j) = IC(LCS(c_i, c_j)) \quad (5)$$

$$sim_{Lin}(w_i, w_j) = \frac{2IC(LCS(c_i, c_j))}{IC(c_i) + IC(c_j)} \quad (6)$$

$$sim_{Jiang\&Conrad}(w_i, w_j) = \frac{1}{1 + IC(c_i) + IC(c_j) - 2IC(LCS(c_i, c_j))} \quad (7)$$

Note that Eq. (7) transforms original semantic distance into similarity and solves the division by zero problem. As IC-based methods lack the important information of path and depth, they are not able to represent concept's distance and specificity accurately. WPath [21] combines structural knowledge and statistical IC to have hybrid semantic representation between concepts.

$$sim_{WPath}(c_i, c_j) = \frac{1}{1 + path(c_i, c_j) * k^{LCS(c_i, c_j)}} \quad (8)$$

where $k \in (0, 1]$ and $k = 1$ means that IC has no contribution in shortest path length. The parameter k ($k = 0.8$ is used as the original proposal) represents the contribution of the LCS's IC which indicates the common information shared by two concepts. WPath aims to give different weights of the shortest path length between concepts based on their shared information, where the path length is viewed as difference and the IC is viewed as commonality. For identical concepts, their path length is 0 so their semantic similarity reaches the maximum similarity 1.

2.2. Word embeddings

Continuous word vector representations contain syntactic and semantic regularities present in the language, expressed as relation offsets in the vector space [13]. Most common approaches train unsupervised word embeddings models without a specific objective, but rather with the aim of capturing language knowledge. This type of word vectors, that are trained using co-occurrence information, are normally called generic or pre-trained word vectors. In this work, pre-trained word vectors are used for all types of proposed feature extraction methods.

In the same way that bag-of-words features are exploited for textual representation in sentiment analysis, word embeddings can be similarly used [13]. Some straight-forward approaches that use embeddings as features for classification have been studied. The work contained in [36] studies the effectiveness of using word embeddings applied to several tasks, one of them being sentiment analysis. This study offers an overview of unsupervised embedding techniques, and how they can obtain meaningful text representations. In the work by [37] an SVM classifier is trained over embedding representations to predict sentiment polarity. Through

their results, authors argue that embeddings contain deep semantic features between words, which results beneficial in sentiment analysis. Also, as shown in [38], word embeddings can be combined with surface features, such as n-grams, sentiment lexicons and lexical features in order to improve sentiment analysis performance. Authors also demonstrate that using embeddings in a schema of model ensembling can yield higher accuracy in the predictions. On top of this, word embeddings based techniques are extensively used in public challenges where competitors aim at obtaining the highest scores in the task of sentiment analysis [39,40]. An interesting work that uses embedding with a special consideration to the reduction the computational complexity is shown in [41]; curiously enough, the authors report a similar accuracy to that of more complex neural models in several tasks, including sentiment analysis. In addition, a model that uses both a semantic similarity measure and embedding representations is presented in [42]. This work addresses the task of aspect-based sentiment analysis by using the similarity information for detecting the aspect of a certain text, while exploiting embedding representations for performing the sentiment classification. For a more detailed description on how to train and use word embeddings, we refer to [43].

In the line of using embedding-based approaches, many authors have proposed complex neural architectures that are able to successfully exploit word vectors. One architecture family is Recurrent Neural Networks, which are able to adapt to different sequence sizes; this results specially useful in text processing, as a document can be modeled as a sequence of sentences, words or even characters. As shown by [44], the study of compositionality in sentiment classification tasks has proven to be relevant. This work proposes the Recursive Neural Tensor Network (RNTN) model, and it also shows how RNTN outperforms previous models on both fine-grained and binary sentiment analysis tasks. The RNTN model represents a phrase using word vectors composed from the structure given by a parse tree, computing vectors for higher nodes in the tree using a tensor-based composition function. Another interesting work [45] tackles the use of Long Short-Term Memory (LSTM) networks, which read the input sequence into a vector. These vectors are used then to predict the sentiment label. A similar approach [46] leverages the use of both LSTM and parse trees, outperforming previous systems and strong baselines in sentence relatedness and sentiment classification tasks. As an additional technique, attention mechanism are normally used in recurrent architectures [47], which allows models to search for parts of the input that are relevant to the problem. Building on top of this, the work in [48] incorporates knowledge of document structure, developing a attention-based approach under the assumption that not all parts of a document are equally important. In this way, the model is allowed to distinguish the importance of the different parts that compose a document, resulting in benefits in sentiment analysis.

Another studied family of neural networks is Convolutional Neural Network (CNN) architectures. CNN models were initially proposed in the field of computer vision [49], although they have been successfully applied to a range of NLP tasks [15]. A representative study of the use of CNN models in sentiment analysis is described in [50]. This work uses pre-trained vectors as well as embeddings that are allowed to be fine-tuned during training, which improves the model. Similarly, the work in [51] shows that the parameter initialization technique in the embeddings of a CNN model highly affects the final result. Also, this work incorporated noisy annotations from data to further refine the weights of the model, optimizing it to perform better in the task at hand. In [52], the authors propose a character-based CNN model that can be trained for several languages, without requiring any machine processes, comparing these results with the more traditional word-based encoding.

Taking this into consideration, it seems natural that some authors have studied the applicability of hybrid architectures, with the aim of leveraging both RNN and CNN models. In [53], the proposed model aims at combining these two types of architectures into a unified approach that firstly classifies documents according to the number of opinion targets with a recurrent model, and later applies a convolutional network to perform sentiment analysis. The work in [54] tackles the encoding of semantic relations between sentences in a document by combining, on one hand, a convolutional network that learns sentence representations, and on the other hand, a recurrent network that encodes the sentence relations.

Although the described neural architectures have a high representational power, there have been efforts towards adapting embeddings models to the task of sentiment analysis. A pioneer work that uses embeddings applied to sentiment analysis is that of [55]. In this work, both semantic and sentiment knowledge is captured in the embeddings. In a similar way, the work in [56] proposes a model that generates semantic sentiment embeddings, capturing context of sentiment together with co-occurrence information. In this way, generated sentiment vectors can be directly used for sentiment analysis tasks without feature engineering. Sentiment signals that are incorporated to word embeddings can come from different sources, as in the case of the work in [57], where the model includes sentiment information from both word and document level. An alternative method for generating sentiment embeddings is possible: refining a pre-trained embeddings model [58]. This work adjusts the original word embeddings, allowing them to be closer to both semantically and sentimentally similar words; while at the same time moving them further away to dissimilar words also in the semantic and sentiment plane.

2.3. Sentiment lexicons

How to use a sentiment lexicon is a common problem that appears in the vast majority of Sentiment Analysis works. An interesting survey that tackles the use of sentiment lexicons for Sentiment Analysis is done in [4]. There are numerous sentiment lexicons that can be used, and determining which characteristics determine the final performance of a system that uses them is an open challenge. In this sense, the work in [12] treats some of these issues, analyzing a number of sentiment lexicons and how they can be complemented in a sentiment analysis system.

One relevant attribute of sentiment lexicons is the creation methodology, which can be divided in three main categories [59, 60]: (i) manual approach, (ii) dictionary-based approach and (iii) corpus-based approach. The manual approach is done by humans, resulting in a very time-consuming process; thus, lexicons generated in such a manner are normally combined with the other two, that make use of automatic strategies. Many dictionary-based methods involve bootstrapping from a small set of opinionated words, and using them as seeds in a process of searching related words in a known dictionary such as WordNet [18] or SentiWordNet [61]. Traditionally, dictionary-based approaches are not able to obtain domain specific orientations [59]. Although, in order to generate domain-adapted dictionaries, some recent works use unsupervised pattern-based approach [62], as well as morphosyntactic information [63]. In spite of this, corpus-based techniques can be used to alleviate the aforementioned issue. This last type of lexicon generation method relies on co-occurrence patterns detected on an unsupervised corpus, as well as a seed set of opinion words to locate opinionated words in the corpus [64,65].

As done in this paper, sentiment lexicons have been used as features in supervised machine learning scenarios. In [66], lexical resources are complemented with n-gram, Part-of-Speech (PoS) and

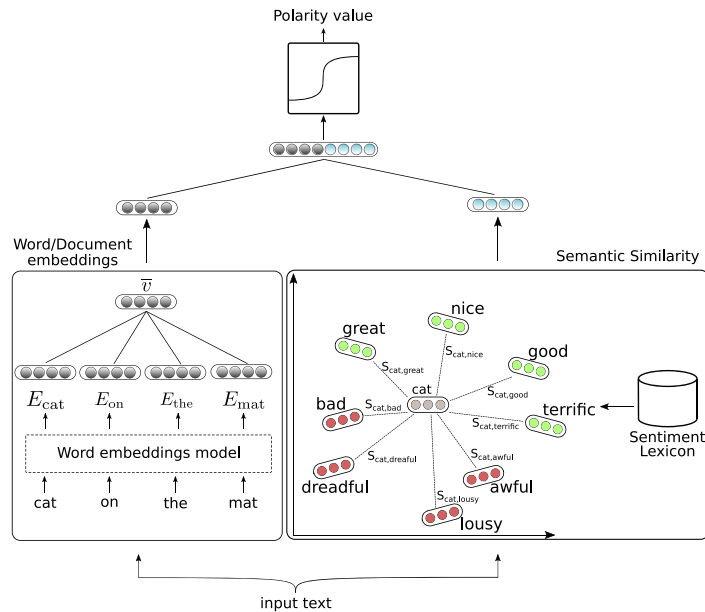


Fig. 1. System architecture diagram.

micro-blogging specific features, such as the presence of emoticons. In this way, authors show that both lexicon and micro-blogging features result of utility in their validation. Similarly, the work in [67] makes use of two sentiment lexicons, combining this information with n-grams, PoS. All these features are used by a SVM-based classifier, resulting in a state-of-the-art system in a SA competition. In some cases, lexicon features have been complemented with domain specific features, as in [68], where the domain is determined by a search query. Also, several lexicons can be used at the same time, as in the case of [69]. This work integrates several lexicons using Markov logic with information about relations between neighboring words.

3. Proposed model

This section introduces the proposed sentiment analysis model, describing its submodules. The semantic similarity feature extraction method is detailed in-depth; as well as its integration with additional embedding-based representations. These two processing steps constitute a full machine learning sentiment analysis system.

Fig. 1 shows a diagram of the proposed model. As shown, the text is processed by two different submodules: (i) word/document embeddings and (ii) semantic similarity. Both use the natural language as input, outputting a feature vector. Feature vectors from both submodules are concatenated and then fed to a machine learning algorithm, which is trained with the sentiment annotations.

3.1. Semantic similarity features

As stated, in general word embeddings contain semantic and syntactic information. Arguably, it is accepted that pre-trained word vectors do not enclose specific sentiment information, as no sentiment-related signal has been included in the training process. In order to include subjective sentiment information into the proposed model, additional information sources must be included

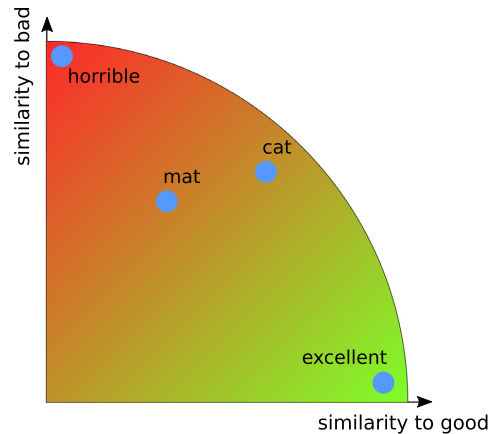


Fig. 2. Conceptual diagram of word projection over a lexicon formed only by the set of words {good, bad}.

in the feature extraction process. In this way, semantic similarity features exploit the aforementioned word embeddings regularities using a selection of sentiment words, namely, a sentiment lexicon vocabulary.

This work proposes the representation of a certain word, that may be outside of the lexicon vocabulary, by a projection to a set of sentiment words extracted from a sentiment lexicon. Such projection is computed using the semantic similarity between words, which can be computed by means of a word embedding model or a word taxonomy. In this way, a certain word is represented by its similarities to the selection of lexicon words.

To illustrate the concept of projection, Fig. 2 shows a conceptual case where only two words (*good* and *bad*) are selected from a sentiment lexicon; it can be seen that several words are projected in then in a two-dimensional space. As described, the values of that space correspond to the semantic similarity of each word (e.g., *horrible*, *cat*, *mat*, *excellent*) to the lexicon words.

More formally, let

$$W^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \dots, w_i^{(i)}, \dots, w_l^{(i)}\} \quad (9)$$

be the set of length l , formed by input tokens that constitute the text to be analyzed. This text can be a sentence, a paragraph or a whole document. Also, we consider that a lexicon is formed by tuples of sentiment word and polarity value: $(w_j^{(s)}, s_j)$. In this way, we define the selection of target sentiment words,

$$W^{(s)} = \{w_1^{(s)}, w_2^{(s)}, \dots, w_i^{(s)}, \dots, w_l^{(s)}\} \quad (10)$$

which is extracted from a sentiment lexicon. Similarly, the vector $\mathbf{l} = [l_1, l_2, \dots, l_l]$ comprises the numerical sentiment values of the word in $W^{(s)}$, as given by a sentiment lexicon. In this work, the selection process that generates the set of words $W^{(s)}$ is done in two steps, following different criteria: (i) frequency of appearance in the training data, and (ii) informativeness of each word towards the training annotation.

The process to generate the features is as follows. For each word $w_i^{(i)} \in W^{(i)}$ and for each sentiment word $w_j^{(s)} \in W^{(s)}$, a similarity value is computed so that

$$S_{i,j} = \text{sim}(w_i^{(i)}, w_j^{(s)}), \quad (11)$$

with $S_{i,j} \in [0, 1]$. This value represents that $w_i^{(i)}$ and $w_j^{(s)}$ are no similar at all if the result is 0, and completely similar if the result is exactly 1. After iterating over all the input words $W^{(i)}$ and all sentiment words $W^{(s)}$, a matrix $S \in \mathbb{R}^{l \times L}$ can be constructed, containing all similarity values.

Following, a pooling function (maximum) is applied column-wise over S , obtaining the semantic similarity feature vector $\mathbf{p}$ of L -length:

$$p_j = \max S_{:,j} = \max \text{sim}(w_k^{(i)}, w_j^{(s)}) \quad \text{for } k \in \{1, 2, \dots, l\} \quad (12)$$

Additionally, we consider feature weighting via the sentiment words associated polarity values through a simple element-wise product, so that $\mathbf{l} \circ \mathbf{p}$ contains the vector of length L , with the weighted features extracted from the input text.

An illustration of how the similarity scores are computed and passed through the pooling function is shown in Fig. 3. It can be seen that the pooling function transforms the dimension of the matrix (that is dependent of the number of words of the input text) to a fixed-dimension vector, defined by the number of selected lexicon words.

Regarding the similarity metric function, two variants are proposed: (i) WordNet semantic similarity [70] and (ii) embedding-based word vector similarity.

The first type of similarity makes use of WordNet taxonomy. In this sense, any of the metrics described in Section 2.1 can be used in this approach. Still, WPath yields better results than the rest of metrics, as detailed in Section 5

Regarding the embedding-based measure, a previously trained word embedding model is used. In this paper, we used the pre-trained word vectors of Word2Vec approach.² Nevertheless, similarity measures can be computed using any word embedding model and are not dependent on embedding dimension either. The embedding similarity is implemented using the dot product between an input word w_i and a sentiment word l_j :

$$\text{sim}(w_i^{(i)}, w_j^{(s)}) = E_{w_i^{(i)}}^T E_{w_j^{(s)}} \quad (13)$$

² <https://code.google.com/archive/p/word2vec/>. Accessed 23 June 2018.

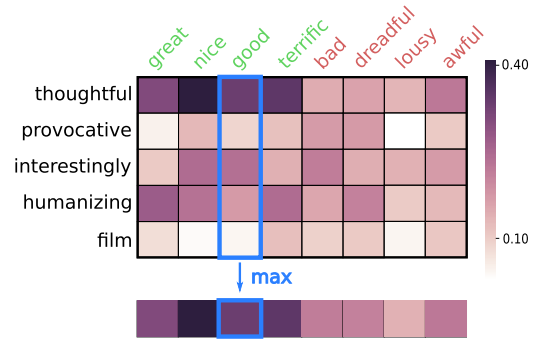


Fig. 3. SIMON similarity computation: similarities are computed against a selection of lexicon words (in green and red), and a *max* function is applied column-wise, obtaining a feature vector. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where $E \in \mathbb{R}^{V \times d}$ is the embedding matrix, and $E_{w_i^{(i)}} \in \mathbb{R}^d$ and $E_{w_j^{(s)}} \in \mathbb{R}^d$ are the word vectors associated with words w_i and l_j respectively.

Algorithm description. The proposed method for extracting similarity features can be expressed as an algorithm, as shown below (Algorithm 1). We call this method SIMilarity-based sentiment projection (SIMON).

Result: Weighted feature vector $\mathbf{v}$

Let $W^{(i)} = \{w_1^{(i)}, \dots, w_l^{(i)}, \dots, w_l^{(i)}\}$ be the set of instance input words

Let SL be a sentiment lexicon

$W^{(s)}, \mathbf{l} \leftarrow \text{selection}(SL)$, being

$W^{(s)} = \{w_1^{(s)}, \dots, w_j^{(s)}, \dots, w_l^{(s)}\}$ the word selection and

$\mathbf{l} = [l_1, l_2, \dots, l_l]$ the sentiment scores of the word selection

foreach $w_i^{(i)} \in W^{(i)}$ **do**

foreach $w_j^{(s)} \in W^{(s)}$ **do**

 compute similarity: $S_{i,j} = \text{sim}(w_i^{(i)}, w_j^{(s)})$,

end

end

for $k \in \{1, 2, \dots, l\}$ **do**

 compute feature vector $\mathbf{p}$: $p_j = \max S_{:,j} = \max \text{sim}(w_k^{(i)}, w_j^{(s)})$

end

compute sentiment weighting: $\mathbf{v} = \mathbf{l} \circ \mathbf{p}$

Algorithm 1: Similarity-based feature extraction SIMON algorithm.

The function “selection” is implemented as described in Algorithm 2. Such algorithm is done in two steps. In the first step, words are filtered by frequency of appearance in a certain dataset, being the frequency cutoff a parameter to adjust. The second step makes use of an ANOVA statistical test between features (which correspond to selected words) and labels. In this way, the F-value is computed for each feature, and the features are selected based on their informativeness regarding the classification task of a certain dataset.

3.2. Embedding text representation

The proposed model uses an embedding-based textual representation that transforms the input text into a fixed-length feature

Result: Selection of words from a sentiment lexicon SL : $W^{(s)}$ and I
 Let $dataset$ be the training dataset
 $W^{(s)}, I := selection(SL, dataset)$
 $tmp \leftarrow frequencyFiltering(SL, dataset)$
 $W^{(s)}, I \leftarrow ANOVAFiltering(tmp, dataset)$

Algorithm 2: Selection over a lexicon method implementation.

vector. As previously studied [38], the number of words in a text directly affects the effectiveness of embedding-based representations. Accordingly, this work makes use of two variations of the same feature extraction method.

The first one is aimed at short texts, as the ones found in the Twitter platform. In this variation, word vectors are extracted for each word in the input text. Following the study in [38], the average pooling operation is performed on all word vectors, resulting in a vector of the same dimension as the original word vectors that is then fed to a logistic regressor. As for the second variation, the representation of a text uses Paragraph Vector [14], and it is used in long texts. This distinction between short texts (typical of online sources) and long texts (more commonly found in review sites) has shown an improvement in the performance of the presented sentiment analysis models [38].

This approach can be used either independently, or in combination with semantic similarity features. Also, the embedding-based text representation method serves as a comparison baseline for the rest of the proposed techniques.

4. Experimental evaluation and discussion

The proposed approaches have been validated against the datasets listed in Section 4.1, and the lexicons described in Section 4.2. Using these datasets and the training evaluation strategy described in Section 4.3, these first experiments are aimed at the characterization of the proposed methods for SA classification.

In order to evaluate the proposed model performance, an extensive experimental evaluation has been made. For this end, eight public datasets have been selected that are widely used in the SA community. Throughout all experiments, results are expressed using the F1 score metric. Also, in order to facilitate replication of this work and to foster research in the field, we make public the implementation of the SIMON method.³

4.1. Datasets

Following, the datasets used for the evaluation are presented. Moreover, Table 1 shows some of the datasets' statistics.

- *Sentiment140* [71], that contains 1,600,000 messages extracted from Twitter using a distant supervision approach. For this, the authors collected tweets that were later filtered using emoticons expressing positive and negative sentiments as noisy labels.
- *SemEval 2013* [72] and *SemEval 2014* [73]. Both datasets are composed of English comments extracted from Twitter and describing a range of topics: entities, products and several entities. Also, these datasets are not directly accessible, as it must be downloaded firstly from the source. Since a number of users have deleted their original comments from the platform, we have not been able to recover the whole dataset, but a subset of it. Obtained sizes are detailed in Table 1.

³ <https://github.com/gsi-upm/simon-paper>.

Table 1

Statistics of the datasets used for the evaluation: number of positive, negative and total instances and average number of words per instance.

Dataset	Positive	Negative	Total	Average no. words
Sentiment140	800,000	800,000	1,600,000	15
SemEval2013	2,315	861	3,176	23
SemEval2014	2,509	932	3,441	22
Vader	2,901	1,299	4,200	16
STS-Gold	632	1,402	2,034	16
IMDB	25,000	25,000	50,000	255
PL04	1,000	1,000	2,000	723
PL05	5,346	5,349	10,695	20

Table 2

Statistics of the lexicons used for the evaluation: number of positive, negative and total instances and total number of words.

Lexicon	No. positive words	No. negative	Total no. words
Liu's	2006	4783	6789
SentiWordNet	2236	3732	5968
ANEW	576	454	1030
AFINN	878	1598	2476

- *Vader* [74]. This dataset contains 4200 tweet-like messages that are originally inspired by real Twitter texts. A subset of the dataset instances are intentionally designed to capture a number of syntactical and grammatical attributes that appear in natural language.
- *STS-Gold* [75], which has been generated as a complement for sentiment analysis evaluation in the Twitter domain. In this way, through a different annotation strategy, this dataset considers the presence of individual entities in tweet labeling process.
- *IMDB* [55]. As of the time of writing, this dataset is widely used in machine learning evaluations. It contains 50,000 annotated reviews from the review site⁴ that gives the name, as well as 50,000 unlabeled reviews. The object of the reviews are movies in the online platform.
- *PL04* [76] and *PL05* [77]. Similarly to the previous one, these datasets contain labeled movie reviews. While in the *PL04* dataset the instances included whole reviews, for the *PL05* these reviews have been split into sentences. Consequently, the average number of words per instance is greatly reduced from one version to another, as it can be seen in Table 1.

When considering the selected datasets, a qualitative distinction that can be made is the domain attribute. For the evaluation, we consider the division of the datasets into two groups, Twitter-related (*Sentiment140*, *SemEval 2014*, *SemEval 2014*, *Vader* and *STS-Gold*); and movie reviews (*IMDB*, *PL04*, *PL05*). As described in Section 3.2, the embedding text representation is made differently for each dataset group. For the Twitter-related, the representation is made through average pooling of the word vectors, while Paragraph Vector is used for movie review datasets.

4.2. Lexicons

As previously described in Section 3.1, the extraction of semantic similarity features requires the use of a lexicon vocabulary and, optionally, the associated sentiment values. In order to extensively study the effect of different sentiment lexica, for this evaluation we have selected four of them. For this work the positive and negative sentiment words and associated scalar values have been selected, discarding the neutral values (see Table 2).

⁴ <http://www.imdb.com/>. Accessed 1 June 2018.

- Bing Liu's [78]. Formed by positive and negative words, it can be found online.⁵ This lexicon contains a number of frequent sentimental words, as well as misspelled words, slang words and common variants. It is worth to highlight that this lexicon has no range in its polarity values, since its two possibilities are either positive (+1) and negative (−1) terms.
- SentiWordNet [61], which is a lexical resource specifically designed for sentiment and opinion mining. The version used in this work (SentiWordNet 3.0)⁶ is an improvement over an earlier version (SentiWordNet 1.0) [79]. SentiWordNet extends WordNet [18], a well-known English lexical database where words are organized into a tree-like structure. Consequently, in SentiWordNet each word is automatically annotated in the range [0, 1] according to its positivity, negativity and neutrality. For the experiments, we compute the aggregated polarity value by subtracting the negative value from the positive value of a word. For example, the word *dangerous* has a positive value of 0.0 and a negative value of 0.75, resulting in −0.75 of aggregated polarity value.⁷ Due to the fact that SentiWordNet annotations have a value for both positive and negative polarities (e.g., the word *easy* has a positive score of 0.625 and a negative score of 0.25), this operation is done to aggregate the overall polarity value.
- Affective Norms for English Words (ANEW) [80] provides emotional ratings for a large number of English words. Said ratings have been calculated by means of measuring the psychological reaction of a person to an specific word. From these, we select the *valence* rating as the most useful for sentiment analysis, with the scale ranging from pleasant to unpleasant.
- AFINN [81]. Since the ANEW lexicon does not contain specific microblogging words, we also consider for the evaluation the AFINN lexicon, that is more focused in this type of language. AFINN word list comprises a number of slang and obscene words, and typical web acronyms. Positive words are scored from 1 to 5, while negative ones have a sentiment score ranging from −5 to −1.

4.3. Evaluation methodology

Given the different characteristics of the evaluation datasets, for this evaluation we have designed a evaluation strategy dataset-wise. In this way, we try to evaluate each dataset so that the best results can be obtained. Consequently, the training and test procedure is done as follows.

Firstly, the *sentiment140* dataset is not used for testing, but only for training and development. The training and development split is done randomly with a 70/30 distribution. In particular, the method implemented for the twitter-domain datasets is to trained the logistic regressor with the extracted features from *sentiment140* dataset, and test the performance of the obtained classifier with the *SemEval2013*, *SemEval2014*, *Vader* and *STS* datasets. Note that these last datasets are not used for training or development, only for testing.

As for the movie review datasets, two different strategies have been used. The authors of the *IMDB* dataset have defined training and testing splits, so those have been used for the experiments. Besides, the *PL04* has associated cross-validation splits, and we have

followed them. Lastly, the *PL05* dataset has no pre-defined splits so, inspired by the *PL04* dataset, we have used cross-validation with random splits.

Finally, the embedding-based text representations have been trained in the following manner. The word vector model used is Skip-gram [13], and it has been trained with the *sentiment140* dataset. For the Paragraph Vector model, we have used the unsupervised split of the *IMDB* dataset.

In reference to the implementation of the word-matching method, it has been done as explained in [3]. Given a text and a certain lexicon, the words in the text are selected if they appear in the lexicon, and their opinion score summed along all the text. Resulting polarity is then normalized to match that of the dataset annotations.

4.3.1. Semantic similarity feature extraction

As explained in Section 3.1, a selection over the vocabulary of a sentiment lexicon is done. Such selection is recommended in order to: (i) reduce the output dimensionality in an attempt to avoid overfitting; and (ii) increase overall performance, as it has been seen that in the experiments the performance in sentiment classification improves if this selection is implemented. As described, this selection is done in two steps. For the first step, which select words by frequency of appearance, we have experimentally set the cutoff to 200 words, distributing equally along polarities, resulting in a selection of 100 positive words and 100 negative words.

As for the second selection step, Fig. 4 illustrates the shape of the importance curves of the 200 most common words for each lexicon and dataset, as computed by the ANOVA test. As seen, the most informative features are concentrated in a low percentile. Based on these results and on a cross-validation exploration of this parameter, we have adjusted the percentile of highest feature scores, setting it to 25.

4.4. Semantic similarity evaluation

Firstly, we evaluate the approach that uses embedding-based similarity. In Table 3, the results are shown. We include the performance of the logistic regression learner trained with different features: embedding-based text representations (W2V/D2V), lexicon-based similarity (Liu, SentiWordNet, ANEW, AFINN), and the combination of both embedding representations and lexicon-based similarity features (as in Liu + W2V/D2V). It can be seen that the proposed approaches surpass the baseline in all datasets excepting STS. This is indicative of the usefulness of these features for sentiment analysis, as the defined baseline constitutes a fairly strong method [38]. Also, results indicate that the combination of the features extracted using the embedding distances with the Liu lexicon and embedding representations yield, in the majority of the datasets, the best performances. In order to further support this result, we have performed the Friedman test [38,82]. The "Rank" columns in the result tables express the ranking of methods as computed by the Friedman test. In this test, less ranking means a method results better in comparison to the rest. All Friedman tests in this work have been performed with ($\alpha = 0.01$). Regarding Table 3, the Friedman test points the combination of distance features using Liu lexicon and embedding representations as the best ones for our experiments.

Following, in order to compare the embedding-based and the Wordnet-based features, the performance of this last type of methods is computed, as shown in Table 4. With respect to the choice of similarity metric, as already explained, there are several metrics that can be used. For this, we have run an extensive set of experiments, computing the performance on all datasets for each of the similarity metrics, and we have seen that WPath yields better results than the rest of metrics; we use a k value of 0.8, as indicated

⁵ <http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html>. Accessed 4 June 2018.

⁶ <http://sentiwordnet.isti.cnr.it/>. Accessed 4 June 2018.

⁷ More specifically, annotations are done over WordNet synsets. For this reason, for a certain word we first extract its most common synset, and from that synset the sentiment annotations. Most common synsets are the first ones; e.g., from word *dog* the most common synset is *dog.n.01*.

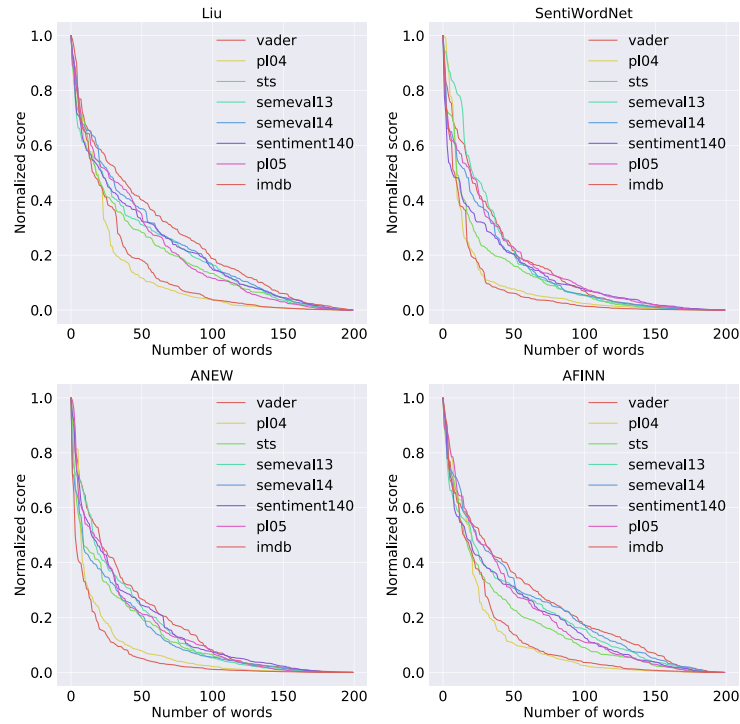


Fig. 4. Sentiment words importance curves for all the lexicons and datasets.

Table 3

F1-scores for the embedding semantic similarity method (SIMON), without sentiment scores. In bold, best score for each dataset.

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PL05	Rank
W2V/D2V	84.54	84.14	88.02	83.75	88.53	88.65	76.43	3.7
Liu	79.61	78.75	85.48	78.69	82.13	84.02	74.15	7.1
Liu + W2V/D2V	87.09	86.48	90.39	82.60	88.99	89.45	78.25	1.6
SentiWordNet	76.62	74.31	84.77	79.15	81.66	80.11	73.95	8.3
SentiWordNet + W2V/D2V	82.45	81.29	87.66	81.72	88.82	88.03	78.26	4.3
ANEW	79.39	78.75	86.91	76.60	79.42	76.66	74.21	7.8
ANEW + W2V/D2V	86.30	85.65	90.08	77.54	88.88	88.09	78.29	3.6
AFINN	81.53	79.17	86.13	80.60	81.99	82.27	74.16	6.4
AFINN + W2V/D2V	86.68	85.92	90.26	83.29	88.97	88.84	78.09	2.3

in [83]. This observation is consistent with previous results tackling WPath metric [83]. The results obtained with the other WordNet-based can be found online,⁸ and have been omitted for extension reasons.

When comparing the WPath similarities with the embedding-based ones, one can see that WPath performances are surpassed by that of embedding-based. This can be explained by attending at the difference in the vocabulary coverage. While the WordNet-based approach benefits from the WordNet vocabulary, embedding-based methods use a much more extensive vocabulary. In this particular case, WordNet has a vocabulary of 155,327 words, while the word embedding model used constitutes a vocabulary of 3 million words. A plausible explanation is that the more extended vocabulary allows the embedding-based method

to capture more word variations, and thus capturing more relevant information.

In addition to this, we have observed that further combining WPath and embedding-based distance features does not improve the sentiment classification performance. Table 5 shows the results of this experiment. One possible explanation for this decrease in performance is the effect of overfitting. When performing the combination, more features are being added, and this could cause the machine learning algorithm that learns from these features to overfit.

Given that the embedding-based distance yields better results, a natural extension of this approach tackles the use of the sentiment scores that are included in each lexicon. That is, the numerical value that a lexicon associates with each word. While in Table 3 no sentiment scores are used, Table 6 shows the performance obtained by the methods that use these scores, as explained in Section 3.1. With the aim of optimizing the results, a normalization

⁸ <https://github.com/gsi-upm/simon-paper>.

Table 4

F1-scores on all datasets for the WPath SIMON semantic similarity metric. In bold, best score for each dataset.

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PI05	Rank
W2V/D2V	84.64	84.11	88.19	83.75	88.55	88.75	76.25	2.7
Liu	56.45	51.55	72.10	62.79	72.90	68.31	57.47	7.7
Liu + W2V/D2V	84.31	83.00	88.44	83.07	88.51	88.87	75.98	4.1
SentiWordNet	66.49	61.18	73.23	62.68	71.47	72.23	56.19	7.4
SentiWordNet + W2V/D2V	85.06	83.75	88.22	83.29	88.52	88.78	76.39	2.7
ANEW	63.11	54.84	71.88	59.75	71.50	68.50	56.61	8.3
ANEW + W2V/D2V	84.31	83.08	87.58	83.58	88.58	88.85	75.99	3.5
AFINN	66.97	58.56	75.80	66.28	72.61	69.05	56.76	6.6
AFINN + W2V/D2V	84.83	83.25	88.95	83.64	88.54	89.23	76.26	2

Table 5

F1-scores on all datasets for the combination of WPath and embedding similarity, and embedding-based representations features. In bold, best score for each dataset.

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PI05
Liu_WPath + Liu_Embedding + W2V/D2V	86.20	85.49	90.28	82.95	89.06	89.17	78.19
SentiWordNet_WPath + SentiWordNet_Embedding + W2V/D2V	86.70	86.23	89.85	82.01	88.80	88.28	78.08
ANEW_WPath + ANEW_Embedding + W2V/D2V	87.34	85.87	86.91	79.91	88.85	88.21	78.03
AFINN_WPath + AFINN_Embedding + W2V/D2V	86.26	84.81	90.41	83.39	89.00	87.97	78.29

in the range $[-1, 1]$ is done to the semantic distance features, as the insertion of the lexicon sentiment scores can augment the feature range, worsening the results.

We observe that, although the sentiment information from the lexicon is added to the feature extraction process, it does not necessarily improve the performance result. As can be seen, when using the Liu lexicon, adding the sentiment scores does not improve, but worse the results. This is to be expected, as Liu lexicon only contains 1 and -1 as polarity values, with no variation; thus, this new information does not affect the obtained features. The reason why the performance decreases in the case of Liu lexicon is that the normalization operation affects the process, decreasing the resulting performance.

Nevertheless, in the case of the SentiWordNet lexicon the addition of sentiment scores and normalization effectively improves the final performance metrics. This, again, can be explained by attending to the granularity of the sentiment annotations in SentiWordNet, that vary in the range of $[-1, 1]$. In the case of ANEW, no relevant effects in performance have been found. Moreover, in AFINN lexicon, the classifier trained with only the distance features improves when using sentiment scores, but it does not surpass the better metric in any dataset.

4.5. Comparison to word-matching

In order to compare the proposed use of the sentiment lexicon against a word-matching baseline, we perform additional experiments that tackle this issue. Table 7 shows the results for the word-matching approach (Section 4.5). As expected, this type of use of a sentiment lexicon does not surpass our proposal in any of the evaluation datasets. Nevertheless, the word-matching methods constitute an interesting baseline, as it represents an approach that is commonly taken.

4.6. Effect of lexicons

Aiming at characterizing the effect of different lexicons on the feature extraction process and the final sentiment classification performance, additional experiments have been performed. With this study, we intend to deep the understanding of the proposed methods, as well as define some mechanisms that can be useful for the performance estimation of these approaches when facing new datasets.

4.6.1. Vocabulary selection

Attending at the process of selection of lexicon words, we firstly study the frequency-based filtering and the assigned informativeness score distribution. In this step of the process, the most common words from a lexicon are selected using this statistical data from the dataset. This selection of words forms the target set to which the features are extracted. Consequently, we have experimentally discovered that latent information regarding dataset characteristics can be inferred. For this, we have used two metrics that are computed from the set of lexicon target words in two datasets. These metrics are: number of common words in the two set of words (n_{cw}), and (ii) the Jensen–Shannon divergence between the informativeness scores of these words (div_{js}).

Following, a cross-dataset evaluation has been done. In this experiment, a learning algorithm trained in a certain dataset is evaluated in all datasets. In order to evaluate the cross-dataset performance variation, the performance difference is defined as:

$$d = \frac{m_c}{m_o} - 1 \quad (14)$$

where m_o is the metric obtained in the original dataset, and m_c is the metric value obtained in the crossed dataset.

Following, we see if from variables n_{cw} and div_{js} , the difference of performance d can be estimated. To assess the effect of the cross-dataset evaluation, we have performed a Least Squares study over variables d (difference of performance), the number of common words in the two selections (variable n_{cw}) and Jensen–Shannon divergence (variable div_{js}). As expected, the number of common words in the selections has been found to be significative. Nevertheless, the Least Squares study shows that Jensen–Shannon divergence has no significance. In this way, the R^2 values of the Least Squares for each lexicon are the following:

- Liu: 0.93 ($p < 0.01$)
- SentiWordNet: 0.94 ($p < 0.01$)
- ANEW: 0.92 ($p < 0.01$)
- AFINN: 0.89 ($p < 0.01$)

These results indicate that, if performing inference over a new dataset, the number of common words of the selection of our method highly indicates the change in performance metrics of the subsequent learning algorithm. This can be useful when applying sentiment classifiers trained with different datasets to a new domain where a dataset is not annotated. The method can guide

Table 6

F1-scores for the embedding semantic distance method, with sentiment scores. In bold, best score for each dataset.

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PL05	Rank
W2V/D2V	84.64	84.11	88.19	83.75	88.54	88.71	76.32	4.3
Liu	80.80	79.61	86.18	78.75	82.11	83.79	74.15	7.1
Liu + W2V/D2V	85.42	85.06	89.21	85.05	89.01	86.61	78.27	2.6
SentiWordNet	79.44	78.71	86.40	79.24	81.65	79.48	74.04	8
SentiWordNet + W2V/D2V	85.49	84.11	89.29	84.46	88.86	88.38	78.33	3.1
ANEW	78.74	78.59	86.70	76.52	79.43	76.63	74.24	8.3
ANEW + W2V/D2V	85.66	84.95	89.55	83.44	88.88	88.21	78.38	2.6
AFINN	83.13	81.28	87.71	80.02	81.99	82.08	74.14	6.6
AFINN + W2V/D2V	85.81	84.62	89.11	84.78	88.99	88.74	78.11	2.4

Table 7

F1-scores for the word-matching method. In bold, best score for each dataset.

Dataset	SemEval13	SemEval14	Vader	STS	IMDB	PL04	PL05
Liu	76.53	73.36	80.25	67.78	73.49	68.33	61.98
SentiWordNet	69.88	68.36	67.21	50.00	66.24	64.73	55.64
ANEW	71.75	69.26	66.43	54.17	66.41	65.96	54.24
AFINN	80.55	78.76	87.22	67.10	73.58	68.90	60.94

the selection of the most suitable trained classifier to this new domain, leading to a reduction in the cross-dataset error. That is, attending to the number of common words between the new and the training datasets, as selected by our approach, such metric offers an estimation of the cross-dataset performance decrement.

4.6.2. Vocabulary similarity

In order to further study the relations between the proposed method and the different lexicons used, and inspired by [12], we have defined a number of metrics that measure the similarity between target words extracted from different lexicons. In order to evaluate the informativeness of these metrics, we perform a cross-domain evaluation.

As explained, for a dataset the SIMON method extracts different target words if using different initial lexicons. Let L_1 be the set of words from the first lexicon to be compared, and L_2 be the set of words of the second studied lexicon. Also, let $t(\cdot)$ be the selection operation over a set of words. Given this, the metrics we use to outline the lexicon similarities are as follows.

- **Total overlap** $Overlap^{total}$, which is simply the number of words that the two lexicons originally share, as defined in [12]. More formally, this metric is defined as $|L_1 \cap L_2|$.
- **Target overlap** $Overlap^{target}$. Similarly, this metric defines the number of words that are common for the target sets from both lexicons. That is, the set $|t(L_1) \cap t(L_2)|$.
- **Total distance** $Distance^{total}$. This metric defines the distance, as defined by a word embedding model, between all the words from the two lexicons. We can write this metric as $dist(L_1, L_2)$. To the extent of our knowledge, this distance measurement of two set of words is a novel approach.
- **Target distance** $Distance^{target}$, which defines the distance between the sets of target words from both lexicons: $dist(t(L_1), t(L_2))$.

Moreover, we define the distance between two sets of words in a word vector space as:

$$dist(L_1, L_2) = \frac{1}{|L_1 \cup L_2|} \sum_{w_i \in L_1} \sum_{w_j \in L_2} sim(w_i, w_j) \quad (15)$$

where the similarity function $sim(\cdot, \cdot)$ is embedding-based, as defined in Section 3.1 (Eq. (13)).

The result of the computation of these metrics is shown in Table 8. Due to the reason that the comparison is between lexicons,

Table 8

Defined metrics results for all the lexicon pairs.

Lexicon	Total overlap	Target overlap	Total distance	Target distance
Liu - SentiWordNet	1483	0.41	0.076	0.151
SentiWordNet - AFINN	583	0.39	0.070	0.150
AFINN - ANEW	298	0.28	0.083	0.131
ANEW - Liu	425	0.24	0.081	0.128
Liu - AFINN	1314	0.65	0.095	0.165
SentiWordNet - ANEW	242	0.21	0.065	0.123

Table 9

Correlation between defined metrics over all datasets.

	Total overlap	Target overlap	Total distance	Target distance
Total overlap	1.00			
Target overlap	0.79	1.00		
Total distance	0.43	0.66	1.00	
Target distance	0.84	0.96	0.52	1.00

the metrics are compared over lexicon pairs (as in Liu - SentiWordNet). Also, when the metrics are dataset-dependent, the average over the datasets is done. We have verified that this simplification does not affect the results, since the metric values practically do not change for the same lexicon pair.

Table 9 shows the correlation between all four defined metrics, as obtained from the values of Table 8. As it stands, we can observe that the *total distance* does not highly correlate with the rest of metrics. This is probably due to that computing embedding similarity over a great number of words (as in the case of the entire lexicon vocabulary) loses, at least in part, the information contained in the word vectors, as explained in [38].

Following, the last step of the experiment is oriented to assess the performance change in a cross-lexicon evaluation. In this way, a learning algorithm is trained using the proposed method in a certain lexicon, and then used for prediction in a different lexicon. Please note that in this experiment, datasets are not changed in an iteration of the experiment. With this, we intend to gain insight of how the performance is affected when interchanging the sentiment lexicon used for the proposed sentiment analysis method. To measure the difference in two performance metrics, we use the metric defined in Section 4.6.1.

The idea is to check if similar lexicons yield similar sentiment analysis performances. For this, we have correlated the values of the *target distance* metric and the difference of performance cross-lexicon. As a result, the obtained value is -0.71 ($p < 0.01$). This result highly indicates that similar lexicons yield similar performances in sentiment analysis. Consequently, when confronted with a new lexicon, one could have a sense of the efficiency of the new lexicon by comparing it with already studied lexicons via the defined lexicon similarity metrics. In this sense, we emphasize the

utility of the *target distance*, as it highly correlates with almost all the rest of metrics, and it makes use of the information contained in a word embedding model.

5. Conclusions

This paper proposes a novel method of utilizing sentiment lexicons, which is based on a semantic similarity metric between text words and lexicon vocabulary. An additional proposal consists of a sentiment analysis that uses this lexicon-based semantic similarity as features, as well as embedding-based representations. In order to evaluate the effectiveness of the model, an extensive experimental evaluation is performed. With the intention of conducting a comparable study, seven public datasets are used, as well as four sentiment lexicons. Also, several statistical tests empirically verify the effectiveness of the proposed feature extraction and its combination to embedding representations. With the aim of further characterizing the feature extraction method, the effect of the lexicon characteristics on the extracted features is studied in depth by means of cross-dataset and cross-lexicon evaluations.

There were three main research questions that drove this work. The first question was whether the proposed semantic distance features are more effective than word-matching methods. Experimental results show that the proposed feature extraction method yields fairly good performances when used with a simple classifier. We consider this result a promising one, since more complex learning architectures can probably boost the overall performance. In addition, attending to both Tables 6 and 7, it can be seen that the word-matching does not improve over the performance of semantic distance features. This difference in performance is specially relevant for the movie review datasets, where word-matching yields much lower scores. This indicates that the use of semantic distance method is able to better extract subjective sentiment information from a lexicon.

The second question tackled the comparison between embedding and taxonomy based semantic similarity. In this way, we have evaluated the performance of both WordNet and embedding based features. As shown, embedding features yield better results in our evaluation. We argue that this contrast is due to the difference in vocabulary coverage, which is intimately related to the generation process of both WordNet and embedding models resources. While WordNet is a manually-generated lexical resource, thus it is limited in coverage; embedding models are automatically inferred, which leads to more coverage in terms of vocabulary.

Lastly, we raised the concern of how the lexicon characteristics affect the proposed feature extraction process. This question is oriented to predict the performance of the model when confronted with a new dataset. For this, several similarity metrics between vocabularies have been defined, determining the correlation between these metrics and the difference of performance in cross-dataset and cross-lexicon experiments. In this regard, the experiments point that the lexicon words largely determines the resulting sentiment analysis performance. That is, similar lexicon word selections yield similar sentiment analysis performances.

To summarize, this work proposes a semantic distance feature extraction method that is combined with embedding-based representations. Nevertheless, we believe that a possible line of future work lies in extending this model to emotion analysis, studying the effects of using emotion lexicons. Furthermore, we intend to extend the domain of this work to a multilingual environment. Finally, we believe that this method can be easily adapted to any domain by adapting the word selection mechanism with domain-oriented data.

Acknowledgments

This research work is partially supported by the Spanish Ministry of Economy, Spain through the project EmoSpaces (RTC-2016-5053-7) and the European Union with Trivalent (H2020 Action Grant No. 740934, SEC-06-FCT-2016), and the project Somedi (ITEA 15011).

References

- [1] J.A. Chevalier, D. Mayzlin, The effect of word of mouth on sales: Online book reviews, *J. Mark. Res.* 43 (3) (2006) 345–354, <http://dx.doi.org/10.1509/jmkr.43.3.345>.
- [2] F. Zhu, X. Zhang, Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics, *J. Mark.* 74 (2) (2010) 133–148, <http://dx.doi.org/10.1509/jmkg.74.2.133>.
- [3] B. Liu, *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*, Cambridge University Press, 2015, <http://dx.doi.org/10.1017/CBO9781139084789>.
- [4] K. Ravi, V. Ravi, A survey on opinion mining and sentiment analysis: Tasks, approaches and applications, *Knowl.-Based Syst.* 89 (2015) 14–46, <http://dx.doi.org/10.1016/j.knosys.2015.06.015>.
- [5] H.-H. Wu, A.C.-R. Tsai, R.T.-H. Tsai, J.Y.-J. Hsu, Building a graded chinese sentiment dictionary based on commonsense knowledge for sentiment analysis of song lyrics, *J. Inf. Sci. Eng.* 29 (4) (2013) 647–662.
- [6] H. Peng, E. Cambria, Csentinet: a concept-level resource for sentiment analysis in chinese language, in: *Computational Linguistics and Intelligent Text Processing. Proceedings of CICLing: International Conference on Computational Linguistics and Intelligent Text Processing*, LNCS, Springer-Verlag, 2018, p. in press, http://dx.doi.org/10.1007/978-3-319-77116-8_7.
- [7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, Lexicon-based methods for sentiment analysis, *Comput. Linguist.* 37 (2) (2011) 267–307, http://dx.doi.org/10.1162/COLL_a_00049.
- [8] E. Cambria, B. Schuller, Y. Xia, C. Havasi, New avenues in opinion mining and sentiment analysis, *IEEE Intell. Syst.* 28 (2) (2013) 15–21, <http://dx.doi.org/10.1109/MIS.2013.30>.
- [9] P.D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2002, pp. 417–424, <http://dx.doi.org/10.3115/1073083.1073153>.
- [10] V. Perez-Rosas, C. Banea, R. Mihalcea, Learning sentiment lexicons in spanish, in: *LREC*, Vol. 12, 2012, p. 73, <http://dx.doi.org/10.18180/teccienca.2017.22.5>.
- [11] X. Ding, B. Liu, P.S. Yu, A holistic lexicon-based approach to opinion mining, in: *Proceedings of the 2008 international conference on web search and data mining*, ACM, 2008, pp. 231–240, <http://dx.doi.org/10.1145/1341531.1341561>.
- [12] F. Bravo-Marquez, M. Mendoza, B. Poblete, Meta-level sentiment models for big social data analysis, *Knowl.-Based Syst.* 69 (2014) 86–99, <http://dx.doi.org/10.1016/j.knosys.2014.05.016>.
- [13] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*.
- [14] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 1188–1196.
- [15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, *J. Mach. Learn. Res.* 12 (Aug) (2011) 2493–2537.
- [16] Y. Bengio, Learning deep architectures for ai, *Found. Trends Mach. Learn.* 2 (1) (2009) 1–127, <http://dx.doi.org/10.1561/2200000006>.
- [17] E. Alpaydin, *Introduction to Machine Learning*, MIT press, 2014.
- [18] G.A. Miller, Wordnet: a lexical database for english, *Commun. ACM* 38 (11) (1995) 39–41, <http://dx.doi.org/10.1145/219717.219748>.
- [19] A. Budanitsky, G. Hirst, Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures, in: *Workshop on WordNet and other Lexical Resources*, Vol. 2, 2001, pp. 2–2.
- [20] P.D. Turney, P. Pantel, et al., From frequency to meaning: Vector space models of semantics, *J. Artificial Intelligence Res.* 37 (1) (2010) 141–188, <http://dx.doi.org/10.1613/jair.2934>.
- [21] G. Zhu, C.A. Iglesias, Computing semantic similarity of concepts in knowledge graphs, *IEEE Trans. Knowl. Data Eng.* 29 (1) (2017) 72–85, <http://dx.doi.org/10.1109/TKDE.2016.2610428>.
- [22] K.W. Church, P. Hanks, Word association norms, mutual information, and lexicography, *Comput. Linguist.* 16 (1) (1990) 22–29.
- [23] R. Gligori, W. ten Kate, Z. Aleksowski, F. van Harmelen, Using google distance to weight approximate ontology matches, in: *Proceedings of the 16th International Conference on World Wide Web*, ACM, New York, NY, USA, 2007, pp. 767–776, <http://dx.doi.org/10.1145/1242572.1242676>.

- [24] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, Vol. 12, 2014, pp. 1532–1543.
- [25] O. Levy, Y. Goldberg, I. Dagan, Improving distributional similarity with lessons learned from word embeddings, *Trans. Assoc. Comput. Linguist.* 3 (2015) 211–225.
- [26] M. Baroni, G. Dinu, G. Kruszewski, Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors, in: *ACL (1)*, 2014, pp. 238–247.
- [27] P. Resnik, Using information content to evaluate semantic similarity in a taxonomy, in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1, IJCAI'95*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 448–453, [arXiv:cmp-lg/9511007](https://arxiv.org/abs/cmp-lg/9511007).
- [28] D. Sánchez, M. Batet, D. Isern, A. Valls, Ontology-based semantic similarity: A new feature-based approach, *Expert Syst. Appl.* 39 (9) (2012) 7718–7728, [http://dx.doi.org/10.1016/j.eswa.2012.01.082](https://doi.org/10.1016/j.eswa.2012.01.082).
- [29] A. Tversky, Features of similarity, *Psychological Rev.* 84 (1977) 327–352, [http://dx.doi.org/10.1037/0033-295X.84.4.327](https://doi.org/10.1037/0033-295X.84.4.327).
- [30] A. Budanitsky, G. Hirst, Evaluating wordnet-based measures of lexical semantic relatedness, *Comput. Linguist.* 32 (1) (2006) 13–47, [http://dx.doi.org/10.1162/coli.2006.32.1.13](https://doi.org/10.1162/coli.2006.32.1.13).
- [31] R. Rada, H. Mili, E. Bicknell, M. Blettner, Development and application of a metric on semantic nets, *IEEE Trans. Syst. Man Cybern.* 19 (1) (1989) 17–30, [http://dx.doi.org/10.1109/21.24528](https://doi.org/10.1109/21.24528).
- [32] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, in: *ACL '94*, Association for Computational Linguistics, Stroudsburg, PA, USA, 1994, pp. 133–138, [http://dx.doi.org/10.3115/981732.981751](https://doi.org/10.3115/981732.981751).
- [33] W.N. Francis, H. Kucera, Brown corpus manual, Brown University.
- [34] D. Lin, An information-theoretic definition of similarity, in: *Proceedings of the Fifteenth International Conference on Machine Learning*, in: *ICML '98*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998, pp. 296–304.
- [35] J.J. Jiang, D.W. Conrath, Semantic similarity based on corpus statistics and lexical taxonomy, *Comput. Linguist.* (Rocling X) (1997) 15, [arXiv:cmp-lg/9709008](https://arxiv.org/abs/cmp-lg/9709008).
- [36] T. Schnabel, I. Labutov, D. Mimno, T. Joachims, Evaluation methods for unsupervised word embeddings, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 298–307.
- [37] D. Zhang, H. Xu, Z. Su, Y. Xu, Chinese comments sentiment classification based on word2vec and svmperf, *Expert Syst. Appl.* 42 (4) (2015) 1857–1863, [http://dx.doi.org/10.1016/j.eswa.2014.09.011](https://doi.org/10.1016/j.eswa.2014.09.011).
- [38] O. Araque, I. Corcuera-Platas, J.F. Sánchez-Rada, C.A. Iglesias, Enhancing deep learning sentiment analysis with ensemble techniques in social applications, *Expert Syst. Appl.* 77 (2017) 236–246, [http://dx.doi.org/10.1016/j.eswa.2017.02.002](https://doi.org/10.1016/j.eswa.2017.02.002).
- [39] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, V. Stoyanov, Semeval-2016 task 4: Sentiment analysis in twitter, in: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 1–18.
- [40] S. Rosenthal, N. Farra, P. Nakov, Semeval-2017 task 4: Sentiment analysis in twitter, in: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 502–518.
- [41] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, *CoRR arXiv:1607.01759*.
- [42] O. Araque, G. Zhu, M. García-Amado, C.A. Iglesias, Mining the opinionated web: Classification and detection of aspect contexts for aspect based sentiment analysis, in: *Data Mining Workshops (ICDMW)*, 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 900–907, [http://dx.doi.org/10.1109/ICDMW.2016.0132](https://doi.org/10.1109/ICDMW.2016.0132).
- [43] S. Lai, K. Liu, S. He, J. Zhao, How to generate a good word embedding, *IEEE Intell. Syst.* 31 (6) (2016) 5–14, [http://dx.doi.org/10.1109/MIS.2016.45](https://doi.org/10.1109/MIS.2016.45).
- [44] R. Socher, A. Perelygin, J. Wu, J. Chuang, C.D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.
- [45] A.M. Dai, Q.V. Le, Semi-supervised sequence learning, in: C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett (Eds.), *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., 2015, pp. 3079–3087.
- [46] K.S. Tai, R. Socher, C.D. Manning, Improved semantic representations from tree-structured long short-term memory networks, *CoRR arXiv:1503.00075*.
- [47] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, *CoRR arXiv:1409.0473*.
- [48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489, [http://dx.doi.org/10.18653/v1/N16-1174](https://doi.org/10.18653/v1/N16-1174).
- [49] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324, [http://dx.doi.org/10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [50] Y. Kim, Convolutional neural networks for sentence classification, *CoRR arXiv:1408.5882*.
- [51] A. Severyn, A. Moschitti, Twitter sentiment analysis with deep convolutional neural networks, in: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, in: *SIGIR '15*, ACM, New York, NY, USA, 2015, pp. 959–962, [http://dx.doi.org/10.1145/2766462.2767830](https://doi.org/10.1145/2766462.2767830).
- [52] J. Wehrmann, W. Becker, H.E.L. Cagnini, R.C. Barros, A character-based convolutional neural network for language-agnostic twitter sentiment analysis, in: *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2384–2391, [http://dx.doi.org/10.1109/IJCNN.2017.7966145](https://doi.org/10.1109/IJCNN.2017.7966145).
- [53] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using bilstm-crf and cnn, *Expert Syst. Appl.* 72 (2017) 221–230, [http://dx.doi.org/10.1016/j.eswa.2016.10.065](https://doi.org/10.1016/j.eswa.2016.10.065).
- [54] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432, [http://dx.doi.org/10.18653/v1/D15-1167](https://doi.org/10.18653/v1/D15-1167).
- [55] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Association for Computational Linguistics, 2011, pp. 142–150.
- [56] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, M. Zhou, Sentiment embeddings with applications to sentiment analysis, *IEEE Trans. Knowl. Data Eng.* 28 (2) (2016) 496–509.
- [57] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, E. Cambria, Learning word representations for sentiment analysis, *Cogn. Comput.* 9 (6) (2017) 843–851, [http://dx.doi.org/10.1007/s12559-017-9492-2](https://doi.org/10.1007/s12559-017-9492-2).
- [58] L.-C. Yu, J. Wang, K.R. Lai, X. Zhang, Refining word embeddings for sentiment analysis, in: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 534–539, [http://dx.doi.org/10.18653/v1/D17-1056](https://doi.org/10.18653/v1/D17-1056).
- [59] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: *Mining Text Data*, Springer, 2012, pp. 415–463, [http://dx.doi.org/10.1007/978-1-4614-3223-4_13](https://doi.org/10.1007/978-1-4614-3223-4_13).
- [60] O. Araque, M. Guerini, C. Strapparava, C.A. Iglesias, Neural domain adaptation of sentiment lexicons, in: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, 2017, pp. 105–110, [http://dx.doi.org/10.1109/ACIIW.2017.8272598](https://doi.org/10.1109/ACIIW.2017.8272598).
- [61] S. Baccianella, A. Esuli, F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining, in: *LREC*, Vol. 10, 2010, pp. 2200–2204.
- [62] P. Agathangelou, I. Katakis, I. Koutoulakis, F. Kokkoras, D. Gunopulos, Learning patterns for discovering domain-oriented opinion words, *Knowl. Inf. Syst.* 55 (1) (2018) 45–77, [http://dx.doi.org/10.1007/s10115-017-1072-y](https://doi.org/10.1007/s10115-017-1072-y).
- [63] G. Vulcu, P. Buitelaar, S. Negi, B. Pereira, M. Arcan, B. Coughland, J.F. Sánchez Rada, C.A. Iglesias Fernandez, Generating linked-data based domain-specific sentiment lexicons from legacy language and semantic resources, in: *5th International Workshop on EMOTION, SOCIAL SIGNALS, SENTIMENT and 2 LINKED OPEN DATA*, 2014.
- [64] A. Yadollahi, A.G. Shahraki, O.R. Zaiane, Current state of text sentiment analysis from opinion to emotion mining, *ACM Comput. Surv.* 50 (2) (2017) 25:1–25:33, [http://dx.doi.org/10.1145/3057270](https://doi.org/10.1145/3057270).
- [65] W.L. Hamilton, K. Clark, J. Leskovec, D. Jurafsky, Inducing domain-specific sentiment lexicons from unlabeled corpora, in: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Conference on Empirical Methods in Natural Language Processing, NIH Public Access, 2016, p. 595, [http://dx.doi.org/10.18653/v1/D16-1057](https://doi.org/10.18653/v1/D16-1057).
- [66] E. Kouloumpis, T. Wilson, J.D. Moore, Twitter sentiment analysis: The good the bad and the omg!, *ICWSM* 11 (538–541) (2011) 164.
- [67] S. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets, in: *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013, pp. 321–327.
- [68] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, Target-dependent twitter sentiment classification, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, Association for Computational Linguistics, 2011, pp. 151–160.
- [69] C. Zirn, M. Niepert, H. Stuckenschmidt, M. Strube, Fine-grained sentiment analysis with structural features, in: *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, pp. 336–344.
- [70] R. Mihalcea, C. Corley, C. Strapparava, et al., Corpus-based and knowledge-based measures of text semantic similarity, in: *AAAI*, Vol. 6, 2006, pp. 775–780.
- [71] A. Go, R. Bhayani, L. Huang, Twitter sentiment classification using distant supervision, *CS224N Project Report*, Stanford (2009) (2009) 12.
- [72] P. Nakov, S. Rosenthal, Z. Kozareva, V. Stoyanov, A. Ritter, T. Wilson, Semeval-2013 task 2: Sentiment analysis in twitter, in: *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), Vol. 2, 2013, pp. 312–320.

- [73] S. Rosenthal, A. Ritter, P. Nakov, V. Stoyanov, Semeval-2014 task 9: Sentiment analysis in twitter, in: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 73–80.
- [74] C.J. Hutto, E. Gilbert, Vader: A parsimonious rule-based model for sentiment analysis of social media text, in: *Eighth International AAAI Conference on Weblogs and Social Media*, 2014, pp. 216–225.
- [75] H. Saif, M. Fernandez, Y. He, H. Alani, Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold, *Inf. Process. Manage.* 1 (2013) 19–29.
- [76] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts, in: *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2004, p. 271, <http://dx.doi.org/10.3115/1218955.1218990>.
- [77] B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in: *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2005, pp. 115–124, <http://dx.doi.org/10.3115/1219840.1219855>.
- [78] M. Hu, B. Liu, Mining and summarizing customer reviews, in: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2004, pp. 168–177, <http://dx.doi.org/10.1145/1014052.1014073>.
- [79] A. Esuli, F. Sebastiani, Sentiwordnet: a high-coverage lexical resource for opinion mining, *Evaluation (2007)* 1–26.
- [80] M.M. Bradley, P.J. Lang, Affective norms for english words (anew): Instruction manual and affective ratings, in: *Tech. Rep.*, Citeseer, 1999.
- [81] F.A. Nielsen, A new anew: Evaluation of a word list for sentiment analysis in microblogs, *arXiv preprint arXiv:1103.2903*.
- [82] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (Jan) (2006) 1–30.
- [83] G. Zhu, C.A. Iglesias, Computing semantic similarity of concepts in knowledge graphs, *IEEE Trans. Knowl. Data Eng.* 29 (1) (2017) 72–85, <http://dx.doi.org/10.1109/TKDE.2016.2610428>.

3.2.4 DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques

Title	DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques
Authors	Araque, Oscar and Gatti, Lorenzo and Staiano, Jacopo and Guerini, Marco
Journal	IEEE Transactions on Affective Computing
Impact factor	JCR 2019 7.512
Scopus citations	5 (02/09/2020)
Web of Science citations	Not indexed (02/09/2020)
Google Scholar citations	11 (02/09/2020)
ISSN	2371-9850
Year	2019
Keywords	natural language processing, Analytical models, Computational modeling, emotion analysis, emotion lexicon, machine learning, Manuals, Mood, Tagging, Task analysis, Twitter, word embeddings
Pages	1-1
DOI	https://doi.org/10.1109/TAFFC.2019.2934444
Online	https://ieeexplore.ieee.org/document/8798675
Abstract	Several lexica for sentiment analysis have been developed; while most of these come with word polarity annotations (e.g., positive/negative), attempts at building lexica for finer-grained emotion analysis (e.g., happiness, sadness) have recently attracted significant attention. They are often exploited as a building block for developing emotion recognition learning models, and/or used as baselines to which the performance of the models can be compared. In this work, we contribute two new resources, that we call DepecheMood++ (DM++): a) an extension of an existing and widely used emotion lexicon for English; and b) a novel version of the lexicon, targeting Italian. Furthermore, we show how simple techniques can be used, both in supervised and unsupervised experimental settings, to boost performance on datasets and tasks of varying degree of domain-specificity. Also, we report an extensive comparative analysis against other available emotion lexica and state-of-the-art supervised approaches, showing that DepecheMood++ emerges as the best-performing non-domain-specific lexicon in unsupervised settings. We also observe that simple learning models on top of DM++ can provide more challenging baselines. We finally introduce embedding-based methodologies to perform a) vocabulary expansion to address data scarcity and b) vocabulary porting to new languages in case training data is not available.

DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques

Oscar Araque, Lorenzo Gatti, Jacopo Staiano, and Marco Guerini

Abstract—Several lexica for sentiment analysis have been developed; while most of these come with word polarity annotations (e.g., positive/negative), attempts at building lexica for finer-grained emotion analysis (e.g., happiness, sadness) have recently attracted significant attention. They are often exploited as a building block for developing emotion recognition learning models, and/or used as baselines to which the performance of the models can be compared. In this work, we contribute two new resources, that we call DepecheMood++ (DM++): a) an extension of an existing and widely used emotion lexicon for English; and b) a novel version of the lexicon, targeting Italian. Furthermore, we show how simple techniques can be used, both in supervised and unsupervised experimental settings, to boost performance on datasets and tasks of varying degree of domain-specificity. Also, we report an extensive comparative analysis against other available emotion lexica and state-of-the-art supervised approaches, showing that DepecheMood++ emerges as the best-performing non-domain-specific lexicon in unsupervised settings. We also observe that simple learning models on top of DM++ can provide more challenging baselines. We finally introduce embedding-based methodologies to perform a) vocabulary expansion to address data scarcity and b) vocabulary porting to new languages in case training data is not available.

Index Terms—emotion analysis, emotion lexicon, natural language processing, machine learning, word embeddings.

1 INTRODUCTION

Obtaining high-quality and high-coverage lexica is an active subject of research [1]. Traditionally, lexicon acquisition has been performed in two distinct ways: either by manual creation (e.g., crowdsourcing annotations) or by automatic derivation from already annotated corpora. While the former approach provides more precise lexica, the latter usually grants a higher coverage. Regardless of the approach chosen, when used as baselines or as additional features for learning models, lexica are often “taken for granted”, meaning that the performance against which a proposed model is evaluated are rather weak, a fact that could be arguably seen to slow down progress in the field.

Thus, this paper firstly investigates whether simple and computationally cheap techniques (e.g., document filtering, text pre-processing, frequency cut-off) can be used to improve both *precision* and *coverage* of a state-of-the-art lexicon that has been automatically inferred from a dataset of emotionally tagged news. Then, we try to answer the following research questions: (i) Can simple machine learning techniques, that only rely on lexicon scores, provide even more challenging baselines for complex emotion analysis models, under the constraint of keeping the required pre-processing at a minimum? (ii) Are such techniques portable across languages? (iii) Can the coverage of a given lexicon be significantly

increased using a simple yet effective methodology?

To this end, we build upon the publicly available DepecheMood lexicon [2], which is generated from news sources distantly annotated with emotional scores. We evaluate and release to the community an extension of the original lexicon, built using a larger dataset, as well as a novel emotion lexicon targeting the Italian language and built with the same methodology. We perform experiments on six datasets/tasks exhibiting a wide diversity in terms of *domain* (namely: news, blog posts, mental health forum posts, Twitter), *languages* (English and Italian), *settings* (both supervised and unsupervised), and *task* (regression and classification).

Results show that: (i) training simple ML models from a high-coverage/high-precision lexicon, derived from general news data, allows obtaining a good performance also on domain-specific tasks, and provides more challenging baselines for complex task-specific models; (ii) depending on the characteristics of the target language, specific preprocessing steps (e.g. lemmatization in case of morphologically-rich languages) can be beneficial; (iii) coverage of the original lexicon can be extended using embeddings, and such technique can provide performance improvements.

2 RELATED WORK

We hereby provide a short review of efforts towards building emotion lexica. The interested reader can find a more thorough overview of sentiment and emotion analysis related fields, as well as lexicon generation, in [3], [4], [5], [6].

2.1 Emotion Lexica

While many sentiment lexica have been produced, fewer linguistic resources for emotion research are described in

- Oscar Araque is with the Grupo de Sistemas Inteligentes of the Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, E.T.S.I. de Telecomunicación, Madrid, Spain. E-mail: o.araque@upm.es
- Lorenzo Gatti is with the Human Media Interaction lab at the University of Twente, Enschede, The Netherlands. E-mail: l.gatti@utwente.nl
- Jacopo Staiano is with reciTAL, 1 rue d’Hauteville, 75010 Paris, France. E-mail: jacopo@recital.ai
- Marco Guerini is with Fondazione Bruno Kessler, Via Sommarive 18, Povo di Trento, Italy and is an AdeptMind scholar. E-mail: guerini@fbk.eu

Manuscript received April 19, 2005; revised August 26, 2015.

Name	Size	Emotions	Type	Annot.	Ref.
NRC VAD	20k	3 (VAD)	Strength	Manual	[7]
ANEW	1k	3 (VAD)	Strength	Manual	[8]
-	14k	3 (VAD)	Strength	Manual	[9]
WordNet-Affect	1k	6	Binary	Manual	[10]
Affect database	2k	9	Strength	Manual	[11]
AffectNet	10k	24	Binary	Auto	[12]
EmoLex	14k	8	Binary	Manual	[13]
NRC Hashtag	16k	8	Strength	Auto	[14]
NRC Affect	6k	4	Strength	Manual	[15]
DepecheMood	37k	8	Strength	Auto	[2]

TABLE 1: Overview of related emotion lexica.

the literature. One of the first is *WordNet-Affect* [10], a manually-built extension of *WordNet*, in which about 1k lemmas are assigned with a label taken from a hierarchy of 311 affective labels, including Ekman’s six emotions [16]. *AffectNet* [12] is a semantic network containing about 10k items; emotions are represented in a vector space where the dimensions are “pleasantness”, “attention”, “sensitivity” and “aptitude”. Each of these dimensions can have a value between -3 and +3, for a total of 24 distinct emotions that can be combined.

The *Affect database* [11] contains 2.5k lemmas taken from *WordNet-Affect*, and has been manually enriched by adding the strength of association with Izard’s basic emotions [17]. *EmoLex* [13] is a crowdsourced lexicon containing 14k lemmas, each annotated with binary associations to Plutchik’s eight emotions [18].

A recent resource is the *NRC Affect Intensity Lexicon* [15], which includes 6k entries manually annotated with joy, fear, anger, and sadness. In a similar line of work, the *NRC Hashtag Emotion Lexicon* [14] has a coverage of over 16k unigrams. This resource has been automatically inferred from microblogging messages distantly annotated by emotional hashtags. As such, this lexicon is particularly useful when applied to the Twitter domain.

In [19], a generative unigram mixture model is proposed as a solution for creating domain-specific emotion lexica by jointly modeling “emotionality and neutrality of words”, starting from a corpus of labeled documents.

Apart from theories of basic emotions, the literature is also proposing a number of non-categorical models, where emotions are represented as points in a vector space. The *Valence-Arousal-Dominance (VAD)* model, in particular, is worth mentioning since multiple resources are based on it. *ANEW* [8] is a set of 1k words manually annotated. A crowdsourced extension of it [9] provides fine-grained VAD ratings for 14k frequent-usage words. A recent work, the *NRC Valence, Arousal, Dominance Lexicon* [7], contains 20k terms manually annotated using Best-Worst Scaling [20]. While many of these resources were used for sentiment analysis, considering only the valence scores (which roughly corresponds to the “sentiment” or “polarity” of a word), the VAD model actually represents emotions. In fact, recent computational works have demonstrated the possibility of seamlessly converting VAD annotations between Ekman’s categorical emotions, and vice versa [21]. Recently, a multi-lingual study using the *GRID* dictionary extended the VAD model to include a new dimension, *Novelty*, but no emotion lexicon is using this model yet [22].

DepecheMood, the lexicon that is the focus of this work, which its extension is described in Section 3, is the largest among these emotion lexica. It is automatically derived from a corpus of news articles with annotated emotions, and thus it can be assumed to be general-purpose. Its set of emotions is also peculiar: while most of other lexica are related or easily mapped to Ekman’s, Plutchik’s or Izard’s sets, the categories “annoyed” and “amused” and are unique for this resource.

Table 1 shows a comparison of these emotion lexica, “type” column indicates whether a lexicon reports binary values for an emotion, or a number indicating its strength.

2.2 Emotions and Deep Learning models

Most of the latest works in sentiment and emotion analysis are based on deep learning architectures, with LSTM-based networks being the prevalent method for detecting sentiment [23], but with a number of successful attempts based on other types of RNNs, CNNs and, recently, the Transformer model (e.g., [24], [25], [26]) or combinations of these.

These architectures represent the meaning of words with dense low-dimensional vectors (embeddings) that are pre-trained or learned from a corpus. In either case, embeddings usually capture the concept of similarity, but do not explicitly model sentiment or emotion, meaning that the vectors of “good” and “bad” will be similar, despite these words having opposite polarity. Intuitively, this is not ideal for sentiment and emotion analysis, and recent works have been trying to address this. For example, [27] learns sentiment-specific word embeddings using tweets with emoticons as indicators of sentiment. [28] shows a “refinement model” to adjust the position of existing embeddings in the vector space, so that words with a different sentiment are further away, using a “gold” emotion lexicon. [29] and [30] integrate prior sentiment knowledge, from the *Stanford Sentiment Treebank* and *Urban Dictionary* respectively, into word embeddings. A similar approach is presented in [31], [32], [33], where the emotion scores from emotion lexica are concatenated to embeddings.

Conversely, embeddings can also be used to generate new lexica; [34] use a feed-forward neural network to predict the emotional intensity of a word, given its vector representation.

In many of these cases, the availability of good lexica is still relevant. Even in the recent *WASSA-2017 Shared Task* on emotion intensity detection, the 8 top-ranking systems used emotion lexica as features [35].

3 DEPECHEMOOD++

DepecheMood++ (DM++) is an extension of the *DepecheMood* lexicon (*DepecheMood*₂₀₁₄, or *DM*₂₀₁₄ henceforth). The original lexicon [2], built in a completely automated and domain-agnostic fashion, has been extensively used by the research community and has demonstrated high performance even in domain-specific tasks, often outperformed only by domain-specific lexica/systems; see for instance [36].

The new version we release in this work is made available for both English and Italian. While the English version of DM++ is an improved version of *DM*₂₀₁₄ built using a larger dataset, the Italian one is entirely new and, to the best of our knowledge, is the first publicly-available large-scale emotion lexicon for this language.

3.1 Data Used

DepecheMood₂₀₁₄ was built using a dataset of 25.3k documents and 13.5M words (530 words per document on average). In this paper, we use an expanded dataset in order (i) to re-build the English lexicon on a larger corpus, and (ii) to build a novel lexicon targeting the Italian language. To this end, we used an extended corpus which has been harvested for a subsequent study on emotions and virality [37] – besides the English articles from *rappler.com* on a more extended period, such corpus includes crowd-annotated news articles in Italian from *corriere.it*.

The *rappler.com* platform is a “social-news” website that embeds an interface, called *Mood Meter*¹, allowing its readers to express with a click their emotional reaction to the story they are reading. The percentages of votes obtained by each emotion are also visualized by the interface, which is depicted in Figure 1. Similarly, *corriere.it*, the online version of a popular Italian newspaper called *Corriere della Sera*, uses *emoticons* to sense the emotional reactions of its readers, as shown in Figure 2². We note that the latter has discontinued its “emotional” widgets, removing them also from the archived articles, a fact that contributes to the relevance of our effort to release the Italian DM++ lexicon.

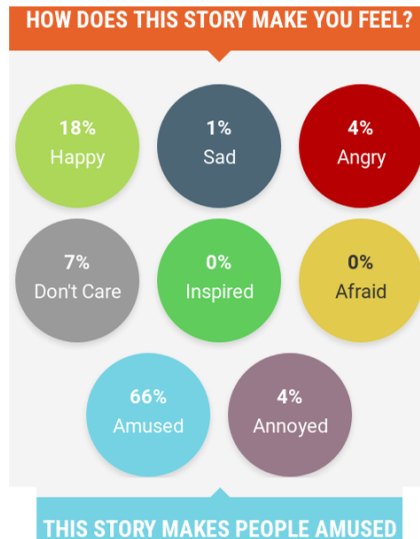


Fig. 1: Rappler’s Mood Meter.

The collected articles span over roughly one year for *Rappler*, and nine months for *Corriere della Sera*, for a total of more than 65,000 news articles and more than 1.5 million annotations. In Table 2 we report a quantitative description of

1. In French, ‘depeche’ means dispatch/news, hence *DepecheMood*.

2. As we are forced to adopt a crawling approach, we have no control on the layout of the widgets shown above. Thus, we cannot rule out possible biases arising from the (fixed) order used to present readers with affective labels and emoticons. Still, the results reported in [2] – obtained thanks to data crawled with the same strategy, indicate that such bias, if present, is negligible.

DOPO AVER LETTO QUESTO ARTICOLO MI SENTO...



Fig. 2: The emoticon-based interface on *corriere.it* articles. The sentence translates to “After reading this article, I feel...”

the data collected from *rappler.com* and *corriere.it*, for more details we refer the reader to the original work [37].

	Rappler	Corriere
Documents	53,226	12,437
Tot. words	18.17 M	4.04 M
Words per Document	341	325
No. of annotations	1,145,543	320,697

TABLE 2: Corpus statistics

In Table 3 we also report the mean percentage of votes for each emotional dimension on the two corpora: it can be seen that HAPPINESS has the highest percentage of votes in comparison to the other dimensions in both *rappler.com* and *corriere.it*, and it is closely followed by ANGRY.

Notice that, as *corriere.it* labels are derived from emoticons rather than words, mapping the two resources is to some extent discretionary: it seems reasonable, though, to compare frequency statistics of 😡 to the union of ANGRY and ANNOYED.

	Rappler _μ	Corriere _μ	
AFRAID	.05	.09	😨
AMUSED	.11	.11	😄
HAPPY	.31	.29	😁
SAD	.12	.10	😞
ANNOYED	.06	.25	😡
ANGRY	.11	-	
DONT_CARE	.05	-	
INSPIRED	.11	-	

TABLE 3: Mean vote percentages obtained by emotions in Rappler and Corriere.

The statistics on both sources confirm the trend noted in [2] about HAPPINESS predominance. Studies on other English datasets, as done in [38] on LiveJournal posts, have in the past noted predominance of the happy mood as well.

Finally, it should be noted that while previous research efforts have exploited documents with emotional annotations on various affect-related tasks [38], [39], [40], [41], those works share the limitation of only providing discrete labels, rather than a continuous score for each emotional dimension. Moreover, these annotations were performed by the document author alone rather than the readers. Conversely, this work leverages the fact that *rappler.com* and *corriere.it* readers can select as many emotions as desired so that the resulting annotations represent a distribution of emotional scores provided by multiple annotators.

3.2 Emotion Lexica Creation

Consistently with [2], the lexica creation is based on a distributional assumption for which the emotions evoked by a word are the emotions that are associated to the documents in which such words appear more often. This also means that we can ‘project’ the emotions of a document to its words. In particular, the methodology for building the lexica consists of the following steps:

- 1) First, a document-by-emotion matrix (M_{DE}) per language is produced, containing the voting percentages for each document in the eight affective dimensions available in *rappler.com* for English and the six available in *corriere.it* for Italian.
- 2) Then, the word-by-document matrices using normalized frequencies (M_{WD}) are computed. That is, the number of occurrences of the word in a given document divided by the total number of words in that document. Previous experiments have shown this to be the best normalization option.
- 3) After that, matrix multiplication are applied between the document-by-emotion and word-by-document matrices ($M_{DE} \cdot M_{WD}$) to obtain a (raw) word-by-emotion matrix M_{WE} . This method allows us to ‘merge’ words with emotions by summing the products of the weight of a word with the weight of the emotions in each document.

Finally, a transformation is done over M_{WE} by first applying normalization column-wise (so to eliminate the over-representation for happiness as discussed previously) and then scaling the data row-wise so to sum up to one.

Two excerpts of the final Matrices M_{WE} both for English and Italian are presented in Tables 4 and 5: they can be interpreted as a list of words with scores that represent how much weight a given word has in each affective dimension. These matrices, that we call *DepecheMood++*, represents our emotion lexica for English and Italian, and are freely available for research purposes at <https://git.io/fxGAP>.

3.3 Validation and Optimization

This section describes several configurations that were used to generate the *DepecheMood++* lexicon. To fairly assess the performance of each configuration, we employ randomly selected validation sets, consisting of 25% of the articles in our data for both *rappler.com* and *corriere.it*. For all the following evaluation experiments, we use the titles of such left-out sets.

Also, in order to facilitate comparisons with previous works, we used the simple approach adopted both by [2] and [38]. On a given headline, a single value for each affective dimension is computed by simply averaging the *DepecheMood++* affective scores of all the words contained in the headline; Pearson’s correlation is then measured by comparing this averaged value to the annotation for the headline.

Word Representations. Throughout the following experiments, three word representations are considered, corresponding to three different pre-processing levels: (i) tokenization, (ii) lemmatization, and (iii) lemmatization combined with Part of Speech (PoS) tagging (the only representation

used in DM_{2014}). We denote these variations as *token*, *lemma* and *lemma#PoS* respectively.

Untagged Document Filtering. First, we noted that a significant percentage of the set of training documents has no emotional annotations – such figure compounds to 8% and 16% for *rappler.com* and *corriere.it*, respectively. Therefore, we compare two versions of the proposed lexica built using two different sets of documents: (i) all documents, as done in the original DM_{2014} ; and (ii), using only documents with a non-zero emotion annotation vector. The results of this experiment are shown in Table 6.

It is evident that training with emotion-annotated documents and removing non-annotated ones leads to an improvement of results, which indicates that untagged documents add noise to the lexicon generation process. Consequently, we use this improved variant in the following experiments.

Frequency Cutoff. Different word frequency cutoff values are also explored to find a threshold that would remove noisy items without eliminating informative ones (in DM_{2014} no cutoff was used; hence hapax were also included in the vocabulary). The performance of *DepecheMood++* under different cutoff values is reported in Table 7: the best performance was obtained using a cutoff value of 10 for both *Rappler* and *Corriere*. Therefore, we use this value on the following experiments. In Table 8 we also report the vocabulary size as a function of cutoff values.

Term Weighting. As previously stated, the text representations are rather simple: an average of the affective scores of the words in the given text. With the aim of improving this, we propose an importance term weighting that makes use of the TF-IDF values to multiply the affect scores. More formally, the representations r for a given text is computed as follows: $r = \frac{1}{T} \sum_t \alpha_t \cdot \text{lex}(t)$, where T is the number of terms in the text, α_t is the weight for term t , and $\text{lex}(t)$ is the affect value as obtained from the emotion lexicon. In this case, α_t weights are computed using the TF-IDF, while in the standard approach $\alpha_t = 1$.

As shown in Table 9, applying importance weighting through TF-IDF decreases the overall performance. This observation indicates that the affective scores contained in the proposed lexicon have sufficient information that allow, through a simple computation, to obtain sufficiently robust representations.

Learning Curves. Next, we aim at understanding if there is a limit, in terms of training dataset size, after which the performance saturates (indicating that further expansions of the corpus would not result beneficial). To this end, we vary the number of documents used to build the lexica using the three different text pre-processing strategies (tokens, lemmatization, and lemmatization with PoS) and evaluate their performance.

Figure 3 shows the correlation values on the left-out sets, yielded by lexica built upon training subsets of increasing size – documents included at each subsequent step have been randomly selected from the original training sets.

The results show that, for the *Rappler* dataset, tokenization and lemmatization approaches consistently achieve the highest performance across various dataset sizes; conversely, on the *Corriere* dataset the lemmatization-based strategies yield the best performance, with a significant improvement

Word	AFRAID	AMUSED	ANGRY	ANNOYED	DON'T CARE	HAPPY	INSPIRED	SAD
awe	0.04	0.18	0.04	0.12	0.14	0.12	0.32	0.04
criminal	0.08	0.11	0.27	0.16	0.11	0.09	0.07	0.10
dead	0.17	0.07	0.18	0.08	0.07	0.05	0.08	0.29
funny	0.04	0.31	0.04	0.13	0.17	0.09	0.16	0.06
warning	0.30	0.07	0.13	0.12	0.08	0.07	0.06	0.16
rapist	0.09	0.08	0.37	0.08	0.18	0.08	0.06	0.07
virtuosity	0.00	0.24	0.00	0.01	0.00	0.41	0.33	0.01

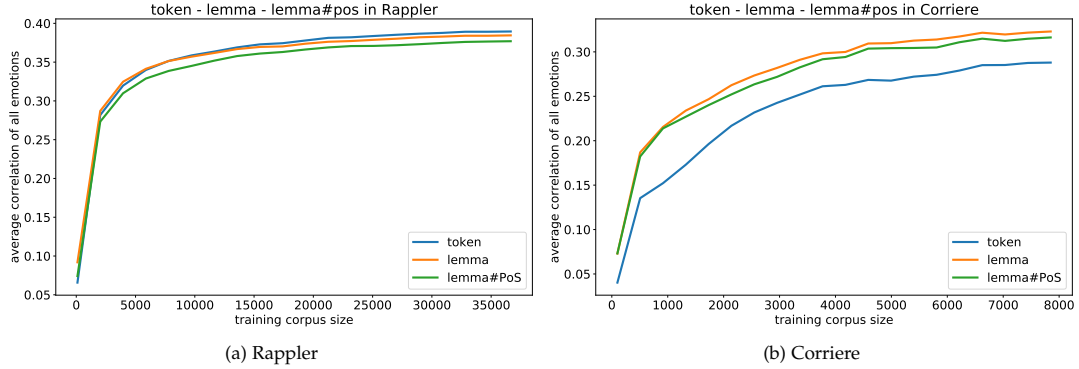
TABLE 4: An excerpt of M_{WE} for English. Dominant Emotion in a word is highlighted for readability.

Fig. 3: Token - lemma - lemma#PoS comparison in both Rappler and Corriere.

	ANNOYED	AFRAID	SAD	AMUSED	HAPPY
stupore	0.18	0.16	0.16	0.28	0.22
criminale	0.21	0.28	0.16	0.20	0.15
morto	0.19	0.19	0.39	0.12	0.11
divertente	0.10	0.13	0.15	0.32	0.30
allarme	0.16	0.25	0.34	0.14	0.11
stupratore	0.40	0.11	0.16	0.18	0.15
virtuoso	0.14	0.18	0.14	0.25	0.29

TABLE 5: An excerpt of M_{WE} for Italian. Dominant Emotion in a word is highlighted for readability.

Dataset	token	lemma	lemma#PoS
Rappler (all)	0.31	0.31	0.30
Rappler (filtered)	0.33	0.32	0.32
Corriere (all)	0.22	0.25	0.24
Corriere (filtered)	0.27	0.30	0.30

TABLE 6: Averaged Pearson's correlation over all emotions on left-out sets for Rappler and Corriere, using all documents in the datasets vs. filtering out those without emotional annotations.

Cutoff	Rappler			Corriere		
	token	lemma	l#p	token	lemma	l#p
1	0.33	0.32	0.31	0.26	0.30	0.29
10	0.33	0.33	0.33	0.27	0.31	0.30
20	0.33	0.33	0.32	0.25	0.30	0.29
50	0.33	0.32	0.31	0.21	0.28	0.27
100	0.31	0.31	0.30	0.16	0.24	0.24

TABLE 7: Frequency cutoff impact on Pearson's correlation for left-out sets for Rappler and Corriere (l#p: lemma#PoS).

over tokenization. A possible explanation for this difference could be the fact that the Italian language (Corriere data

Cutoff	Rappler			Corriere		
	token	lemma	l#p	token	lemma	l#p
1	165k	154k	249k	116k	72k	81k
10	37k	30k	44k	20k	13k	13k
20	26k	20k	29k	12k	8k	8k
50	16k	12k	16k	6k	4k	4k
100	10k	8k	10k	3k	3k	3k

TABLE 8: Number of words in generated lexica using different cutoff values.

Dataset	token	lemma	lemma#PoS
Rappler (standard)	0.33	0.33	0.33
Rappler (TF-IDF weighted)	0.28	0.28	0.27
Corriere (standard)	0.27	0.31	0.30
Corriere (TF-IDF weighted)	0.17	0.17	0.16

TABLE 9: Averaged Pearson's correlation over all emotions on left-out sets for Rappler and Corriere, using our standard text representations vs. applying TF-IDF weighted representations.

is in Italian) is morphologically richer than English (e.g., the adjective *good*, that can be written as *buono* or *buona*, *buoni*, *buone*); thus, lemmatization can reduce data sparseness that harms the final lexicon quality.

Furthermore, Tables 10 and 11 show the results obtained using all three versions of our resource, measured by Pearson correlation between the emotion annotation and the computed value (as indicated before). When possible, obtained values are compared to those of *DepecheMood*₂₀₁₄; in general, it can be seen that the current work improves the performance with respect to the earlier version by a significant margin (6 points on average).

Emotion	DM ₂₀₁₄	token	lemma	l#p
AFRAID	0.30	0.38	0.39	0.38
AMUSED	0.16	0.33	0.32	0.31
ANGRY	0.40	0.39	0.40	0.40
ANNOYED	0.15	0.21	0.21	0.21
DON'T CARE	0.16	0.21	0.22	0.21
HAPPY	0.30	0.35	0.35	0.35
INSPIRED	0.31	0.36	0.36	0.35
SAD	0.36	0.39	0.41	0.41
AVERAGE	0.27	0.33	0.33	0.33

TABLE 10: Pearson's correlation on Rappler left-out set.

Emotion	token	lemma	lemma#PoS
ANNOYED	0.40	0.44	0.44
AFRAID	0.14	0.16	0.15
SAD	0.35	0.40	0.39
AMUSED	0.20	0.22	0.22
HAPPY	0.29	0.32	0.31
AVERAGE	0.27	0.31	0.30

TABLE 11: Pearson's correlation on Corriere left-out set.

Considering the naïve approach that has been used, we can reasonably conclude that the quality and coverage of our resource are the reason of such results and that adopting more complex approaches (i.e., compositionality) can further improve the performance of emotion recognition.

4 EVALUATION

In order to thoroughly evaluate the generated lexica, we assess their performance in both regression and classification tasks. The methodology described in Section 3, used to obtain emotion scores for a given sentence/document, is common to all these experiments. The experiments are also restricted to the English DM++, as we are not aware of Italian datasets for emotion recognition.

For all these experiments, we used the best DM++ configuration from the previous section. That is, filtering out untagged documents and using a frequency cut-off set at 10, with no term weighting.

We report the performance obtained on the datasets described in Table 12, and compare such results with the related previous works. Furthermore, in Table 13, we compare the coverage statistics over the same datasets for DM₂₀₁₄ and DM++ (both with and without frequency cut-off). As can be seen, using DM++ with cut-off 10 still grants a significantly higher coverage with respect to DM₂₀₁₄, without losing too much coverage as compared to the version without cut-off.

Name	Domain	Reference
SemEval-2007	News headline	[42]
Tweet Emotion Intensity Dataset (WASSA-2017)	Twitter messages	[43]
Blog	Blog posts	[44]
CLPsych-2016	Mental health blogs	[45]
Emotion Twitter	Twitter messages	[46]
SemEval-2018	Twitter messages	[47]

TABLE 12: Annotated public datasets used in the evaluation.

4.1 Unsupervised Regression Experiments

The SemEval 2007 dataset on "Affective Text" [42] was gathered for a competition focused on emotion recognition

in over one thousand news headlines, both in regression and classification settings. This dataset was meant for unsupervised approaches (only a small development sample was provided) to avoid simple text categorization approaches.

Dataset	DM ₂₀₁₄	DepecheMood++					
		1 cutoff			10 cutoff		
		tok	lem	l#p	tok	lem	l#p
SemEval07	0.64	0.91	0.94	0.92	0.85	0.88	0.85
WASSA17	0.40	0.66	0.62	0.65	0.56	0.52	0.53
Blog posts	0.64	0.93	0.92	0.91	0.84	0.83	0.80
CLPsych16	0.57	0.87	0.87	0.87	0.81	0.78	0.76
Multi. Emo.	0.63	0.92	0.91	0.91	0.84	0.83	0.80
SemEval18	0.35	0.60	0.56	0.60	0.50	0.45	0.47

TABLE 13: Statistics on words coverage per headline. tok: tokens, lem: lemma, l#p: lemma and PoS.

It is to be observed that the affective dimensions present in the test set – based on the six basic emotions model [16] – do not exactly match with the ones provided by Rappler's Mood Meter; therefore, we adopted the mapping previously proposed in [2] for consistency.

Table 14 reports the results obtained using our lexicon on the SemEval 2007 test data set. DM++ is found to improve consistently upon the previous version (DM₂₀₁₄) on all emotions, granting an improvement over the previous SOTA of up to 6 points.

Emotion	DM ₂₀₁₄	token	lemma	lemma#PoS
ANGER	0.37	0.47	0.47	0.44
FEAR	0.51	0.60	0.59	0.60
JOY	0.34	0.38	0.37	0.35
SADNESS	0.44	0.46	0.48	0.50
SURPRISE	0.19	0.21	0.23	0.22
Average	0.37	0.43	0.43	0.42

TABLE 14: Pearson's correlation on SemEval 2007 dataset.

4.2 Supervised Regression Experiments

Turning to evaluation on supervised regression tasks, the approach we use is inspired by [48]: rather than only using happiness scores to predict happiness, we also use the scores for all emotions. This approach is justified by the evidence that emotion scores often tend to be correlated or anti-correlated (e.g., joy and surprise are correlated, joy and sadness are anti-correlated).

Hence, for each dataset we built N prediction models (one for each emotion present in the dataset, using the lexicon scores computed using either tokens, lemmas, or lemma#PoS), as features: $e_j \sim [lex_1, \dots, lex_i, \dots, lex_N]$ where e_j is the predicted score for emotion j using all the available lex_i , where lex_i is the average score (computed on the test sentence) derived from the lexicon for emotion i .

We have used several learning algorithms: linear regression, Support Vector Machines (SVM), decision tree, random forests, and multilayer perceptron in a ten-fold cross-validation setting. Again, consider that we are not trying to optimize the models mentioned above, but just trying to understand if there is room for strong supervised baselines that uses simple machine learning methodologies on top of our simple features (i.e., feeding the DepecheMood++ lexicon scores to each emotion model). Table 15 shows that

this is the case, with improvements ranging from 8 to 15 points, depending on the dataset.

Dataset	Word rep.	LR	SVM	DT	RF	MLP	DM++
Rappler	token	0.38	0.38	0.37	0.40	0.38	0.33
	lemma	0.37	0.37	0.36	0.40	0.39	0.33
	lemma#p	0.36	0.36	0.35	0.38	0.38	0.33
Corriere	token	0.35	0.35	0.34	0.39	0.37	0.27
	lemma	0.36	0.36	0.35	0.40	0.38	0.31
	l#p	0.35	0.36	0.35	0.40	0.37	0.30
SemEval	token	0.49	0.49	0.49	0.53	0.46	0.43
	lemma	0.50	0.50	0.48	0.53	0.47	0.43
	l#p	0.49	0.49	0.50	0.53	0.48	0.42

TABLE 15: Regression results in supervised settings: Pearson’s correlation averaged over all emotions. LR: linear regression, DT: decision trees, RF: random forest, MLP: multilayer perceptron.

Additionally, we have performed another experiment that compares DepecheMood++ to existing approaches on a popular emotion dataset: Tweet Emotion Intensity Dataset. We replicated the approach outlined in [43] on the dataset therein presented, using DepecheMood++ as lexicon.

The results are reported in Table 16. Replicating the original work, Pearson’s correlation has been used to measure the performance; since DepecheMood++ is not a Twitter-specific lexicon, we also report non-domain-specific approaches.

It can be seen that DepecheMood++ improves over DM₂₀₁₄ on all emotions, while the lexicon from [49] performs slightly better only on *joy*. In an aggregate view of the problem (average of all emotions) DepecheMood++ yields the best performance among the non-Twitter-specific lexica.

Lexicon	Anger	Fear	Joy	Sad	Avg.
Bing Liu	0.33	0.31	0.37	0.23	0.31
MPQA	0.18	0.20	0.28	0.12	0.20
NRC-EmoLex	0.18	0.26	0.36	0.23	0.26
SentiWordNet	0.14	0.19	0.26	0.16	0.19
DM ₂₀₁₄	0.35	0.28	0.27	0.30	0.30
DM++ token	0.33	0.32	0.34	0.40	0.33
DM++ lemma	0.36	0.32	0.36	0.42	0.35
DM++ lemma#PoS	0.31	0.31	0.34	0.41	0.34

TABLE 16: Comparison with generic lexica of [43] in the task of emotion intensity prediction. Avg. is the average of all four emotions.

4.3 Supervised Classification Experiments

Finally, akin to Section 4.2 but this time in a classification setting, we performed additional experiments to benchmark DepecheMood++ against existing works in the literature.

In a first experiment, we replicated the evaluation methodology described in [44], tackling emotion detection in blog data with the accuracy score. Consistently with the original work detailed in [44], we trained Naive Bayes and Support Vector Machines models using only the DepecheMood++ affective scores. This methodology serves as a comparison to the original work, where General Inquirer and WordNet-Affect annotations were used. Results are reported in Table 17, including the performance of the original DM₂₀₁₄ lexicon for comparison purposes. We observe that DepecheMood++ brings a significant improvement and outperforms the other models.

System	Naive Bayes	SVM
[44]		
General Inquirer + WN-Affect	72.08	73.89
DM ₂₀₁₄	75.86	76.89
DM++ token	76.01	78.04
DM++ lemma	76.50	78.03
DM++ lemma#PoS	74.18	76.18

TABLE 17: Comparison with [44] in the task of emotion classification accuracy.

Further, we replicated the work detailed in [45], and assessed and compared with it the performance of DepecheMood++ on a domain-specific task: in [45], the authors tackle relevance prediction of blog posts on medical forums dedicated to mental health. It is worth noting that as [45] used the original DepecheMood₂₀₁₄, this experiment also serves the purpose of evaluating the improvements brought by DepecheMood++, which are shown in Table 18.

System	Accuracy	F1-macro
[45] using DM ₂₀₁₄	81.62	75.21
DM++ token	82.15	81.34
DM++ lemma	82.26	81.46
DM++ lemma#PoS	82.25	81.44

TABLE 18: Comparison with [45] in the task of mental health post classification.

Finally, we turn to the data used in [46]. To assess the performance of DM++ on this dataset we trained a linear SVM classifier, consistently with the experimental setup adopted by the original authors, and evaluated its performance through k-fold cross-validation ($k=5$). Similarly to the previous evaluations, the only features used are the DepecheMood++ affective scores. In order to properly balance the data, we perform under sampling using the Near Miss method [50]. Additionally, samples that do not share vocabulary with the lexica have been discarded (e.g., messages consisting of one rare word).

Table 19 reports the results in terms of average accuracy and F1-macro. Consistently with the previous evaluation, we include the performance of DM₂₀₁₄ for comparison purposes. The newer version of our resource is found to improve F1-macro over the 2014 version, but by a small margin in terms of accuracy. The original work reports an accuracy of 73.24%. Considering that the model used in [46] exploits several features other than DepecheMood (DM₂₀₁₄ version) such as WordNet Affect annotations, unigrams, POS tagging, and dependency parsing, we deem the obtained performance very positive.

Features	Accuracy	F1-macro
DM ₂₀₁₄	68.10	59.12
DM++ token	68.43	62.35
DM++ lemma	68.02	62.27
DM++ lemma#PoS	67.77	60.76

TABLE 19: Comparison with [46] in the task of multi-class emotion classification.

4.4 Comparison to Existing Lexica

With the aim of comparing our lexicon generation proposal against similar approaches, we design an additional set of

experiments.

Thus, the evaluation considers the proposed DepecheMood++ in three different scenarios, reducing its vocabulary. This vocabulary reduction is aimed at evaluating whether the performance improvements in these tasks are due to the quality of the annotations or to a high vocabulary coverage. We propose two methods for performing the vocabulary reduction. The three scenarios are: (i) **Original lexicon**, with 10 as the selected cutoff, as shown before. (ii) **Reduction method 1**: compute the common vocabulary between DM and the comparison lexicon. Then, reduce both DM and comparison lexicon to that common vocabulary. (iii) **Reduction method 2**: vary the DM cutoff parameter (from original Rappler dataset) until the vocabulary size is approximately equal to that of the comparison lexicon. Note that, when using this method, there is no need to reduce also the comparison lexicon.

Regarding the datasets, a selection over the ones included in Table 12 has been made, since the evaluation methodologies broadly vary, and some of them are not suitable for comparing different lexica (e.g., using unigrams with the vocabulary selection of a lexicon). Consequently, the selected datasets and evaluation tasks are SemEval 2007, SemEval 2018, WASSA-2017, and Emotion Twitter. Please note that for all datasets we perform a supervised regression or classification task, depending on the dataset, except for SemEval 2007, where we experiment with both unsupervised and supervised scenarios. Also, the evaluation metric is Pearson's correlation for SemEval 2007, 2018, and WASSA 2017; while F1-Score is used for the Emotion Twitter.

Table 20 shows the evaluation result. Firstly, it can be seen that DepecheMood++ has a superior performance in the news domain (SemEval 2007), both in unsupervised and supervised regimes. Nevertheless, the results indicate that in comparison to Twitter-oriented lexica (NRC Hashtag Emotion is the predominant in this aspect) our resource's performance is below. Such observation suggests that although DepecheMood++ can obtain better results than alternative resources in generic domains (e.g., news, blogs), its performance suffers when applied to the specific domain of Twitter.

4.5 Comparison to SOTA and cross-dataset evaluation

As stated, SemEval 2018 competition hosted an emotion intensity recognition task (EI-reg)³ on which we evaluate several approaches, including our lexicon. In addition, we employ a state-of-the-art system to reflect the differences between lexicon-based and more complex approaches. For this, we make use of the Bidirectional Encoder Representations from Transformers (BERT) [51], a neural network based language model using the Transformer architecture presented in [52].

In this experiment, we compare our approach, both the unsupervised and supervised variants, to the state-of-the-art BERT system. As learning algorithm, we have used the Linear SVM regressor. For comparison purposes we did not fine-tune BERT, and thus fed the pre-trained BERT contextual representations to a Linear SVM regressor. As explained in the original publication [51], the selected representations

3. <https://competitions.codalab.org/competitions/17751>

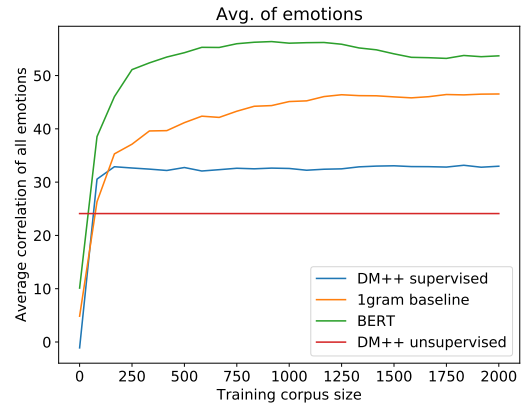


Fig. 4: Learning curves of DM++ unsupervised, DM++ supervised, 1-gram, and BERT approaches over the SemEval 2018 dataset.

consist of the sum of the representations computed in the last four layers. Also, the official unigram baseline is included in this evaluation. With the intent of checking that the comparison is correct, we have replicated such unigram baseline, seeing that the obtained results are comparable to those reported in the competition.

Table 21 shows Pearson's correlation of the mentioned approaches. As expected, the BERT system highly outperforms both the baseline and our lexicon-based approach. Notwithstanding the improvements yield by DepecheMood++ in the previously analyzed scenarios, it suffers – as can be expected – from the limitations of lexicon-based approaches when compared to complex neural models. The feature vector concatenation of DepecheMood++ and BERT has also been evaluated. Still, performance did not vary significantly from using BERT solely, and thus, we omit such results.

Lastly, we include the learning curves of the studied approaches when varying the training size data. This result is shown in Figure 4. In light of this figure, it can be seen that the BERT model rapidly obtains a fairly good performance, indicating that the quality of its pre-trained representations can be used with success, even in a scenario where training data is scarce. It can be seen that our unsupervised approach outperforms all other models when almost no training data is used. Such behavior is to be expected and suggests that the unsupervised approach can be successfully used in a domain where there are none or only a reduced number of annotated instances. Also, the supervised version of DepecheMood++ outperforms its unsupervised counterpart when enough training data is available.

In light of the results shown above, we want to explore the performance of these same approached in a cross-dataset scenario. To this end, we have performed such an experiment using both SemEval datasets: SemEval 2018 and SemEval 2007. This evaluation is of particular interest since these two datasets contain data from very different domains: Twitter and news headlines, respectively. This characteristic allows us to observe how the proposed approaches behave in such

Reduction	Lexicon	SemEval 2007 unsup.	SemEval 2007 sup.	SemEval 2018	WASSA 2017	Emotion Twitter
None	DM++	0.43	0.47	0.34	0.36	0.69
	AffectInt	0.38	0.30	0.42	0.35	0.84
	EmoLex	0.24	0.27	0.34	0.41	0.83
	Hashtag	0.22	0.29	0.45	0.47	0.83
Method 1	DM++ (AffectInt)	0.27	0.37	0.34	0.40	0.85
	AffectInt	0.37	0.31	0.41	0.41	0.84
	DM++ (EmoLex)	0.31	0.40	0.29	0.32	0.94
	EmoLex	0.23	0.27	0.29	0.35	0.83
	DM++ (Hashtag)	0.43	0.45	0.34	0.37	0.72
	Hashtag	0.22	0.30	0.45	0.46	0.86
Method 2	DM++ (AffectInt)	0.38	0.43	0.25	0.25	0.92
	DM++ (EmoLex)	0.43	0.47	0.32	0.32	0.73
	DM++ (Hashtag)	0.43	0.47	0.34	0.33	0.72

TABLE 20: Pearson’s correlation of lexicon comparison on SemEval 2007 test set with an unsupervised approach.

Approach	Anger	Fear	Joy	Sadness	Average
OFFICIAL 1GRAM	0.52	0.52	0.57	0.45	0.52
OUR 1GRAM	0.50	0.58	0.53	0.48	0.52
DM++ UNSUP.	0.25	0.25	0.21	0.24	0.24
DM++ SUP.	0.31	0.37	0.34	0.31	0.33
BERT	0.65	0.60	0.60	0.60	0.59

TABLE 21: Pearson’s correlation over all instances of SemEval 2018 dataset on the emotion intensity regression task (EI-reg).

situations.

For consistency, we include as a baseline a unigram approach, select Linear SVM as the learner algorithm, and use Pearson’s correlation over the test set of SemEval 2007 dataset. The same SemEval 2018 training set is used for all systems. At test time, the regressors predict the emotion score without doing any training on the objective dataset.

Approach	Fear	Anger	Joy	Sadness	Surprise	Average
1GRAM	0.29	0.14	0.11	0.10	-	0.16
DM++ UNSUP.	0.60	0.47	0.38	0.46	0.22	0.43
DM++ SUP.	0.60	0.50	0.40	0.47	-	0.49
BERT	0.51	0.53	0.44	0.48	-	0.49

TABLE 22: Pearson’s correlation over test set of SemEval 2007 dataset, with the regressors trained on SemEval 2018.

As shown in Table 22, the performance of the supervised DepecheMood++ approach is similar to that of BERT. This is an insightful result, indicating that the proposed technique is robust to a cross-dataset environment. Such an argument is also reinforced by attending to the unsupervised DepecheMood++ performance, which stands 0.06 points below the best correlation. Besides, the unsupervised approach has the advantage that it can predict predictions over emotions that are not present in the training data, as is the case of *surprise*, which is present in the SemEval 2007 dataset, but not in SemEval 2018. Lastly, it can be seen that the unigram baseline has a high performance drop in this cross-dataset scenario.

5 DISCUSSION OF RESULTS

Several findings that are consistent across datasets emerge from the above experiments.

As shown in Tables 10 and 14, the new DM++ lexicon effectively and consistently improves (see the additional

benchmarks reported in Tables 16, 17 and 18) over the original work [2].

Such improvements can be explained by the expansion of the training data, which enables the generated lexicon to capture emotional information better; in Figure 3, we showed the performance obtained by lexica built on random and increasing subsets of the source data, and observe consistent improvements until a saturation point is met.

Moreover, we found that adding a word frequency cutoff parameter leads to a benefit in the performance of the generated lexicon. In our experiments, we find an optimal value of 10 for both the English and Italian lexica.

Turning to the benefits of typical pre-processing stages, our experiments included tokenization, lemmatization, and PoS tagging. While the original DM₂₀₁₄ lexicon only provided a lemma#PoS-based vocabulary, we show that – for English – tokenization suffices, and further stages in the pre-processing pipeline do not significantly contribute to the precision of the generated lexicon; conversely, we obtained significant improvements by adding a lemmatization stage for Italian (see Figure 3), probably because of the morphologically-rich nature of the Italian language.

Further, as shown in Table 6, filtering out untagged documents contributes to lexicon precision, arguably resulting in a higher-quality resource.

Finally, the extensive experiments reported in the previous sections show the quality of the English lexicon we release, in diverse domains/tasks. Our results indicate that additional data would not lead to further improvements, at least for English. Conversely, we note that promising results are shown by the Italian resource we also provide to the community.

6 EXTENDING THE LEXICA

As mentioned in Section 2.2, over the last few years word embeddings have gained wide acceptance and popularity in the research community, and have proven to be very effective in several NLP tasks, including sentiment analysis. Extensive research efforts have been put into obtaining useful word representations in both monolingual and multilingual settings.

In the following, we thus describe two simple techniques allowing to (i) increase coverage of an existing lexicon, and (ii) derive a cross-lingual extension of an existing lexicon.

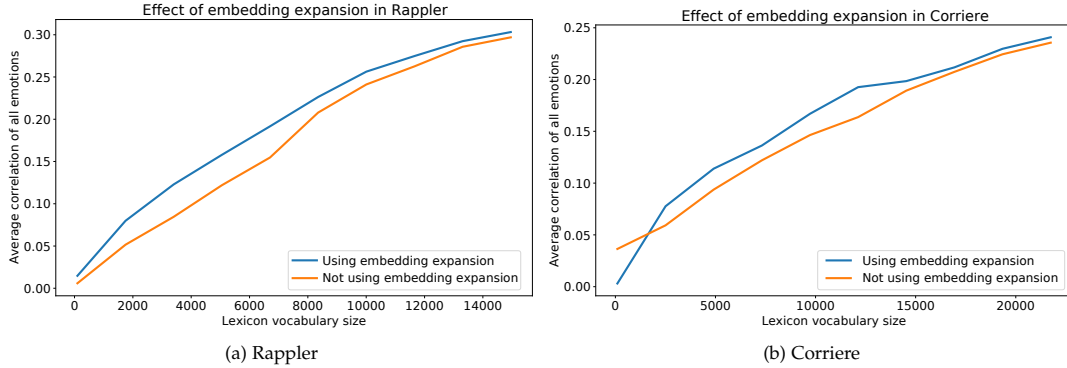


Fig. 5: Difference in performance when using the embedding expansion in both Rappler and Corriere.

6.1 Increasing Coverage through Embeddings

In order to expand lexica vocabulary using a word embedding model, we propose a technique we call *embedding expansion*: the idea is to map words that do not initially appear in a specific lexicon to a word that is contained in the lexicon mentioned above.

Hence, given an out-of-vocabulary (OOV) word w_i , we search in the embedding space for the closest word that is included in the lexicon l_j so that $d(w_i, l_j)$ is minimal, where $d(\cdot, \cdot)$ is the cosine distance in the embedding model. Finally, the emotion scores from l_j are assigned to w_i .

We have evaluated Rappler and Corriere datasets using the token-based versions of DepecheMood++, intending to observe the effect of using embedding expansion. To this end, we proceed to:

- 1) remove random subsets, of decreasing size, from the original lexicon vocabulary;
- 2) apply the expansion at each step;
- 3) measure performance against the corresponding test sets (i.e., the left-out sets described in Section 3.3).

The pre-trained word embeddings used for English are the ones published by [53], while for Italian we use those by [54]; Figure 5 shows the results of this evaluation.

As observed, performing the embedding expansion can improve the performance of the emotion lexicon. The higher improvement is achieved when the vocabulary has been reduced to roughly half of its elements for the two datasets. When the vocabulary is not reduced, instead, the improvement tends to disappear.

Thus, we conclude that this improvement can enhance the performance of lexica with low coverage by expanding their vocabulary through an embedding model. Nevertheless, when the lexicon has a high coverage (as in the case of DepecheMood++), further extending it using embedding expansion does not lead to meaningful improvements.

6.2 Cross-Lingual Projection to an Unseen Language

An important limitation of the proposed lexicon generation method lies in the limited availability of emotion annotated corpora. In light of this, and to delve into cross-lingual

settings with the generated lexicon, we have performed a projection of the DM++ resource to the French language.

Methodology. In order to derive a French version from the original English lexicon, using the DM++ token-based version and the ground-truth bilingual dictionaries released within the Facebook MUSE library [55], we performed the following steps:

- 1) eliminate cross-lingual homographs (optional), i.e. tokens that are present in both source and target languages, but may mean different things;
- 2) aggregate synonyms by averaging the emotional values of the source tokens they map to: for instance the French *injustifié* maps to the English *unwarranted*, *unjustified*, *wrongful*.

Evaluation. To evaluate the derived French lexicon, we resort to the Cross-Lingual Sentiment dataset [56], which contains comparable bilingual data in both English and French. More specifically, this dataset contains 2,000 Amazon reviews from three distinct domains: music, DVD, and books. We have used the provided train/test splits for replicability purposes. For all domains and both languages, the samples are binary annotated, and their distribution is balanced.

Lexicon variant	Books	DVD	Music	Avg F1
DM++ token (EN)	63.55	64.70	66.20	64.81
DM++ token (FR) - setting 1	61.74	60.26	63.65	61.88
DM++ token (FR) - setting 2	59.96	62.37	62.34	61.55

TABLE 23: Comparison between English and French variants of DM++ in the Cross-Lingual Sentiment dataset. Metric is F1-Score. For the French lexicon, setting 1 refers to a lexicon built using cross-lingual homographs removal, while setting 2 refers to a lexicon built without such step.

We summarize the results obtained for this experiment in Table 23. As can be expected, we notice that a slight decrease of performance appears when using the lexicon in a cross-lingual setting, an effect likely due to the degradation of the lexicon as caused by the simple English-French projection process adopted. Nevertheless, the performance decrease is not substantial, with an overall loss of $\sim 3\%$; further, we see that the optional step of removing cross-lingual

homographs is only marginally beneficial. In the future, we plan to leverage the actual word representations provided by aligned multi-lingual embeddings to refine the projection procedure.

7 CONCLUSIONS

The contributions of this paper are two-fold: first, we release to the community two new high-performance and high-coverage lexica, targeting English and Italian languages; second, we extensively benchmark different setup decisions affecting the construction of the two resources, and further evaluate the performance obtained on several datasets/tasks exhibiting a wide diversity in terms of domain, languages, settings, and task. Our findings are summarized below.

Better baselines come cheap: we have shown how simple classifiers/regressors built on top of the proposed lexica and without additional features obtain good performance even on domain-specific tasks, and can provide more challenging baselines when evaluating complex task-specific models; we hypothesize that such computationally cheap approaches might benefit any lexicon.

Target language matters: we built our lexica for two languages, English and Italian, using consistent techniques and data, a fact that allowed us to experiment with different settings and cross-evaluate the results. In particular, we found that, for English, building a token-based vocabulary suffices and further pre-processing stages do not help; for Italian, our experiments highlighted significant improvements when adding a lemmatization step. We interpret this in light of the morphologically-rich nature of the Italian language with respect to English.

Embeddings do help (once again): we investigated a simple embeddings-based extension approach, and showed how it benefits both performance and coverage of the lexica. We deem such technique particularly promising when dealing with limited annotated datasets or with the projection to languages for which annotated corpora are not available.

ACKNOWLEDGEMENTS

This work has been partially funded by the Spanish Ministry of Economy through the project Semola (TEC2015-68284-R).

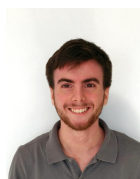
REFERENCES

- [1] S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010, pp. 26–34.
- [2] J. Staiano and M. Guerini, "DepecheMood: a lexicon for emotion analysis from crowd-annotated news," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 427–433.
- [3] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.
- [4] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," *Mining Text Data*, pp. 415–463, 2012.
- [5] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? finding strong and weak opinion clauses," in *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI '04)*, 2004, pp. 761–769.
- [6] G. Paltoglou, M. Thelwall, and K. Buckley, "Online textual communications annotated with grades of emotion strength," in *Proceedings of the 3rd International Workshop of Emotion: Corpora for research on Emotion and Affect (EMOTION)*, 2010, pp. 25–31.
- [7] S. M. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- [8] M. M. Bradley and P. J. Lang, "Affective norms for English words (ANEW): Instruction manual and affective ratings," *Technical Report C-1, University of Florida*, 1999.
- [9] A. B. Warriner, V. Kuperman, and M. Brysbaert, "Norms of valence, arousal, and dominance for 13,915 English lemmas," *Behavior research methods*, vol. 45, no. 4, pp. 1191–1207, 2013.
- [10] C. Strapparava and A. Valitutti, "WordNet-Affect: an affective extension of WordNet," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2004, pp. 1083–1086.
- [11] A. Neviarouskaya, H. Prendinger, and M. Ishizuka, "EmoHeart: conveying emotions in second life based on affect sensing from text," *Advances in Human-Computer Interaction*, vol. 2010, p. 1, 2010.
- [12] E. Cambria, T. Mazzocco, A. Hussain, and C. Eckl, "Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space," in *Proceedings of the International Symposium on Neural Networks*, 2011, pp. 601–610.
- [13] S. M. Mohammad and P. D. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, pp. 436–465, 2013.
- [14] S. M. Mohammad and S. Kiritchenko, "Using hashtags to capture fine emotion categories from tweets," *Computational Intelligence*, vol. 31, no. 2, pp. 301–326, 2015.
- [15] S. M. Mohammad, "Word affect intensities," in *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [16] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion," *Journal of Personality and Social Psychology*, vol. 17, pp. 124–129, 1971.
- [17] C. E. Izard, *Human emotions*. Plenum Press, 1977.
- [18] R. Plutchik, "A general psychoevolutionary theory of emotion," *Theories of emotion*, vol. 1, pp. 3–31, 1980.
- [19] A. Bandhakavi, N. Wiratunga, S. Massie, and D. Padmanabhan, "Lexicon generation for emotion detection from text," *IEEE Intelligent Systems*, vol. 32, no. 1, pp. 102–108, Jan 2017.
- [20] S. Kiritchenko and S. M. Mohammad, "Capturing reliable fine-grained sentiment associations by crowdsourcing and best-worst scaling," in *Proceedings of The 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2016.
- [21] S. Buechel and U. Hahn, "Emotion representation mapping for automatic lexicon construction (mostly) performs on human level," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 2892–2904.
- [22] J. R. Fontaine, K. R. Scherer, and C. Soriano, *Components of emotional meaning: A sourcebook*. Oxford University Press, 2013.
- [23] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.
- [24] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1422–1432.
- [25] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING 2014)*, 2014, pp. 69–78.
- [26] A. Ambartsoumian and F. Popowich, "Self-attention: A better building block for sentiment analysis neural network classifiers," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2018, pp. 130–139.
- [27] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014, pp. 1555–1565.
- [28] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings for sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 534–539.
- [29] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, and M. Zhou, "Sentiment embeddings with applications to sentiment analysis," *IEEE transactions on knowledge and data Engineering*, vol. 28, no. 2, pp. 496–509, 2015.

- [30] Y. Li, Q. Pan, T. Yang, S. Wang, J. Tang, and E. Cambria, "Learning word representations for sentiment analysis," *Cognitive Computation*, vol. 9, no. 6, pp. 843–851, Dec 2017.
- [31] Y. Mansar, L. Gatti, S. Ferradans, M. Guerini, and J. Staiano, "Fortia-FBK at SemEval-2017 task 5: Bullish or bearish? inferring sentiment towards brands from financial news headlines," *arXiv preprint arXiv:1704.00939*, 2017.
- [32] Z. Zhang, Q. Zhou, and H. Wu, "NLPZZX at SemEval-2018 task 1: Using ensemble method for emotion and sentiment intensity determination," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 116–122.
- [33] G. Daval-Ferrot, A. Boucheikif, and A. Moreau, "Epita at SemEval-2018 task 1: Sentiment analysis using transfer learning approach," in *Proceedings of The 12th International Workshop on Semantic Evaluation*, 2018, pp. 151–155.
- [34] M. Köper, E. Kim, and R. Klinger, "IMS at EmoInt-2017: Emotion intensity prediction with affective norms, automatically extended resources and deep learning," in *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 2017, pp. 50–57.
- [35] S. M. Mohammad and F. Bravo-Marquez, "WASSA-2017 shared task on emotion intensity," in *Proceedings of the Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA)*, 2017.
- [36] V. Bobicev, M. Sokolova, and M. Oakes, "What goes around comes around: learning sentiments in online medical forums," *Cognitive Computation*, vol. 7, no. 5, pp. 609–621, 2015.
- [37] M. Guerini and J. Staiano, "Deep feelings: A massive cross-lingual study on the relation between emotions and virality," in *Proceedings of the 24th International Conference on World Wide Web (WWW '15)*, 2015, pp. 299–305.
- [38] C. Strapparava and R. Mihalcea, "Learning to identify emotions in text," in *Proceedings of the 2008 ACM symposium on Applied computing*, 2008, pp. 1556–1560.
- [39] G. Mishne, "Experiments with mood classification in blog posts," in *Proceedings of ACM SIGIR 2005 Workshop on Stylistic Analysis of Text for Information Access*, 2005.
- [40] J. R. Bellegarda, "Emotion analysis using latent affective folding and embedding," in *Proceedings of the NAACL HLT 2010 workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 2010, pp. 1–9.
- [41] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu, "Building large-scale Twitter-specific sentiment lexicon: A representation learning approach," in *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, 2014, pp. 172–182.
- [42] C. Strapparava and R. Mihalcea, "SemEval-2007 task 14: Affective text," in *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval)*, 2007, pp. 70–74.
- [43] S. M. Mohammad and F. Bravo-Marquez, "WASSA-2017 shared task on emotion intensity," *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 34–49, 2017.
- [44] S. Aman and S. Szpakowicz, "Identifying expressions of emotion in text," in *Proceedings of the International Conference on Text, Speech and Dialogue*, 2007, pp. 196–205.
- [45] A. Cohan, S. Young, and N. Goharian, "Triaging mental health forum posts," in *Proceedings of the Third Workshop on Computational Linguistics and Clinical Psychology*, 2016, pp. 143–147.
- [46] R. C. Balabantaray, M. Mohammad, and N. Sharma, "Multi-class Twitter emotion classification: A new approach," *International Journal of Applied Information Systems*, vol. 4, no. 1, pp. 48–53, 2012.
- [47] S. M. Mohammad, F. Bravo-Marquez, M. Salameh, and S. Kiritchenko, "Semeval-2018 Task 1: Affect in tweets," in *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, 2018.
- [48] D. Beck, T. Cohn, and L. Specia, "Joint emotion analysis via multi-task gaussian processes," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1798–1803.
- [49] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.
- [50] I. Mani and I. Zhang, "kNN approach to unbalanced data distributions: a case study involving information extraction," in *Proceedings of the Workshop on Learning from Imbalanced Datasets (LIDTA)*, 2003.
- [51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.
- [53] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [54] R. Tripodi and S. Li Pira, "Analysis of Italian word embeddings," *arXiv preprint arXiv:1707.08783*, 2017.
- [55] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," *arXiv preprint arXiv:1710.04087*, 2017.
- [56] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (ACL 10)*, 2010, pp. 1118–1127.



systems in order to enhance sentiment and emotion analysis techniques.



Lorenzo Gatti is a postdoctoral researcher at the University of Twente (The Netherlands). He received his PhD from the University of Trento (Italy) in 2018, under the supervision of Prof. Oliviero Stock and funded by FBK-irst, with a thesis on natural language processing for computational creativity. Besides NLP, his current research interests include natural language generation, in particular of affective language.



Jacopo Staiano is Research Lead at reciTAL. He received his PhD from the University of Trento (Italy) under supervision of Prof. Nicu Sebe in 2014. He held post-doctoral positions at FBK and UPMC-Sorbonne Université before transitioning to industry as a senior researcher. He has received the Honorable Mention at ACM DIS 2012, the Best Paper Award at ACM UbiComp 2014, and the UMUAL James Chen Annual Award in 2016.



Marco Guerini is a researcher in Computational Linguistics, focusing on persuasive communication, sentiment analysis and social media. He graduated in Philosophy and holds a Ph.D. in Information and Communication Technologies since 2006. In 2011 his activities have been partially funded by a Google Research Award while in 2016 he received a research sponsorship from eBay for his activities. He is author of several scientific publications published in top-level conference proceedings and international journals and program committee member at international conferences.

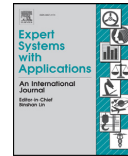
3.2.5 Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications

Title	Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications
Authors	Araque, Oscar and Corcuera-Platas, Ignacio and Sánchez-Rada, J. Fernando and Iglesias, Carlos A.
Journal	Expert Systems with Applications
Impact factor	JCR 2019 5.452
Scopus citations	152 (02/09/2020)
Web of Science citations	103 (02/09/2020)
Google Scholar citations	224 (02/09/2020)
ISSN	0957-4174
Volume	77
Year	2017
Keywords	Ensemble, Machine learning, Natural language processing, Sentiment analysis
Pages	236–246
DOI	https://doi.org/10.1016/j.eswa.2017.02.002
Online	http://www.sciencedirect.com/science/article/pii/S0957417417300751
Abstract	Deep learning techniques for Sentiment Analysis have become very popular. They provide automatic feature extraction and both richer representation capabilities and better performance than traditional feature based techniques (i.e., surface methods). Traditional surface approaches are based on complex manually extracted features, and this extraction process is a fundamental question in feature driven methods. These long-established approaches can yield strong baselines, and their predictive capabilities can be used in conjunction with the arising deep learning methods. In this paper we seek to improve the performance of deep learning techniques integrating them with traditional surface approaches based on manually extracted features. The contributions of this paper are sixfold. First, we develop a deep learning based sentiment classifier using a word embeddings model and a linear machine learning algorithm. This classifier serves as a baseline to compare to subsequent results. Second, we propose two ensemble techniques which aggregate our baseline classifier with other surface classifiers widely used in Sentiment Analysis. Third, we also propose two models for combining both surface and deep features to merge information from several sources. Fourth, we introduce a taxonomy for classifying the different models found in the literature, as well as the ones we propose. Fifth, we conduct several experiments to compare the performance of these models with the deep learning baseline. For this, we use seven public datasets that were extracted from the microblogging and movie reviews domain. Finally, as a result, a statistical study confirms that the performance of these proposed models surpasses that of our original baseline on F1-Score.



Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Enhancing deep learning sentiment analysis with ensemble techniques in social applications



Oscar Araque*, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, Carlos A. Iglesias

Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación, Departamento de Ingeniería de Sistemas Telemáticos,
Avenida Complutense 30, Madrid, Spain

ARTICLE INFO

Article history:

Received 30 June 2016

Revised 31 January 2017

Accepted 1 February 2017

Available online 3 February 2017

Keywords:

Ensemble

Deep learning

Sentiment analysis

Machine learning

Natural language processing

ABSTRACT

Deep learning techniques for Sentiment Analysis have become very popular. They provide automatic feature extraction and both richer representation capabilities and better performance than traditional feature based techniques (i.e., surface methods). Traditional surface approaches are based on complex manually extracted features, and this extraction process is a fundamental question in feature driven methods. These long-established approaches can yield strong baselines, and their predictive capabilities can be used in conjunction with the arising deep learning methods. In this paper we seek to improve the performance of deep learning techniques integrating them with traditional surface approaches based on manually extracted features. The contributions of this paper are sixfold. First, we develop a deep learning based sentiment classifier using a word embeddings model and a linear machine learning algorithm. This classifier serves as a baseline to compare to subsequent results. Second, we propose two ensemble techniques which aggregate our baseline classifier with other surface classifiers widely used in Sentiment Analysis. Third, we also propose two models for combining both surface and deep features to merge information from several sources. Fourth, we introduce a taxonomy for classifying the different models found in the literature, as well as the ones we propose. Fifth, we conduct several experiments to compare the performance of these models with the deep learning baseline. For this, we use seven public datasets that were extracted from the microblogging and movie reviews domain. Finally, as a result, a statistical study confirms that the performance of these proposed models surpasses that of our original baseline on F1-Score.

© 2017 The Authors. Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license.

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

1. Introduction

The growth of user-generated content in web sites and social networks, such as Twitter, Amazon, and Trip Advisor, has led to an increasing power of social networks for expressing opinions about services, products or events, among others. This tendency, combined with the fast spreading nature of content online, has turned online opinions into a very valuable asset. In this context, many Natural Language Processing (NLP) tasks are being used in order to analyze this massive information. In particular, Sentiment Analysis (SA) is an increasingly growing task (Liu, 2015), whose goal is the classification of opinions and sentiments expressed in text, generated by a human party.

The dominant approaches in sentiment analysis are based on machine learning techniques (Pang, Lee, & Vaithyanathan, 2002; Read, 2005; Wang & Manning, 2012). Traditional approaches frequently use the Bag Of Words (BOW) model, where a document is mapped to a feature vector, and then classified by machine learning techniques. Although the BOW approach is simple and quite efficient, a great deal of the information from the original natural language is lost (Xia & Zong, 2010), e.g., word order is disrupted and syntactic structures are broken. Therefore, various types of features have been exploited, such as higher order n -grams (Pak & Paroubek, 2010). Another kind of feature that can be used is Part Of Speech (POS) tagging, which is commonly used during a syntactic analysis process, as described in Gimpel et al. (2011). Some authors refer to this kind of features as *surface* forms, as they consist in lexical and syntactical information that relies on the pattern of the text, rather than on its semantic aspect.

Some prior information about sentiment can also be used in the analysis. For instance, by adding individual word polarity to the previously described features (Pablos, Cuadros, & Rigau, 2016). This

* Corresponding author.

E-mail addresses: o.araque@upm.es (O. Araque), ignacio.cplatas@alumnos.upm.es (I. Corcuera-Platas), jfernando@dit.upm.es (J.F. Sánchez-Rada), cif@gsi.dit.upm.es (C.A. Iglesias).

<http://dx.doi.org/10.1016/j.eswa.2017.02.002>

0957-4174/© 2017 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license. (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

prior knowledge usually takes the form of *sentiment lexicons*, which have to be gathered. Sentiment lexicons are used as a source of subjective sentiment knowledge, where this knowledge is added to the previously described features (Cambria, 2016; Kiritchenko, Zhu, & Mohammad, 2014; Melville, Gryc, & Lawrence, 2009; Nasukawa & Yi, 2003).

The use of lexicon-based techniques has a number of advantages (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011). First, the linguistic content can be taken into account through mechanisms such as sentiment valence shifting (Polanyi & Zaenen, 2006) considering both intensifiers (e.g. very bad) and negations (e.g. not happy). In addition, sentiment orientation of lexical entities can be differentiated based on their characteristics. Moreover, language-dependent characteristics can be included in these approaches. Nevertheless, lexicon-based approaches have several drawbacks: the need of a lexicon that is consistent and reliable (Taboada et al., 2011), as well as the variability of opinion words across domains (Turney, 2002), contexts (Ding, Liu, & Yu, 2008) and languages (Perez-Rosas, Banea, & Mihailescu, 2012). These dependencies make it hard to maintain domain independent lexicons (Qiu, Liu, Bu, & Chen, 2009).

In general, extracting complex features from text, figuring out which features are relevant, and selecting a classification algorithm are fundamental questions in the machine learning driven methods (Agarwal, Xie, Voysha, Rambow, & Passonneau, 2011; Sharma & Dey, 2012; Wilson, Wiebe, & Hoffmann, 2009). Traditional approaches rely on manual feature engineering, which is time consuming.

On the other hand, deep learning is a promising alternative to traditional methods. It has shown excellent performance in NLP tasks, including Sentiment Analysis (Collobert et al., 2011). The main idea of deep learning techniques is to learn complex features extracted from data with minimum external contribution (Bengio, 2009) using deep neural networks (Alpaydin, 2014). These algorithms do not need to be passed manually crafted features: they automatically learn new complex features. Nevertheless, a characteristic feature of deep learning approaches is that they need large amounts of data to perform well (Mikolov, Chen, Corrado, & Dean, 2013). Both automatic feature extraction and availability of resources are very important when comparing the traditional machine learning approach and deep learning techniques.

However, it is not clear whether the domain specialization capacity of traditional approaches can be surpassed with the generalization capacity of deep learning based models in all NLP tasks, or if it is possible to successfully combine these two techniques in a wide range of applications.

In this paper, we propose a combination of these two main sentiment analysis approaches through several ensemble models in which the information provided by many kinds of features is aggregated. In particular, this work considers an ensemble of classifiers, where several sentiment classifiers trained with different kinds of features are combined, and an ensemble of features, where the combination is made at the feature level. In order to study the complementarity of the proposed models, we use six public test datasets from two different domains: Twitter and movie reviews. Moreover, we performed a statistical study on the results of these ensemble models in comparison to a deep learning baseline we have also developed. We also present the complexity of the proposed ensemble models. Besides, we present a taxonomy that classifies the models found in the literature and the ones proposed in this work.

With our proposal we seek answers to the following questions, using the empirical results we have obtained as basis:

1. Is there a framework for characterizing existing approaches in relation to the ensemble of deep and traditional techniques in sentiment analysis?

2. Can deep learning approaches benefit from their ensemble with surface approaches?
3. How do different deep and surface ensembles compare in terms of performance?

The rest of the paper is organized as follows. Section 2 shows previous work on both ensemble techniques and deep learning approaches. Section 3 describes the proposed taxonomy for classifying ensemble methods that merge surface and deep features, whereas Section 4 addresses the proposed classifier and ensemble models. In Section 5, we describe the designed experimental setup. Experimental results are presented and analyzed in Section 6. Finally, Section 7 draws conclusions from previous results and outlines the future work.

2. Related work

In this section we offer a brief summary of the previous work in the context of ensemble methods and deep learning algorithms for Sentiment Analysis.

2.1. Ensemble methods for sentiment analysis

In the field of ensemble methods, the main idea is to combine a set of models (base classifiers) in order to obtain a more accurate and reliable model in comparison with what a single model can achieve. The methods used for building upon an ensemble approach are many, and a categorization is presented in Rokach (2005). This classification is based on two main dimensions: how predictions are combined (rule based and meta learning), and how the learning process is done (concurrent and sequential).

Regarding the first dimension, on the one hand, in *rule based* approaches predictions from the base classifiers are treated by a rule, with the aim of averaging their predictive performance. Examples of rule based ensembles are the majority voting, where the output prediction per sample is the most common class; and the weighted combination, which linearly aggregates the base classifiers predictions. On the other hand, *meta learning* techniques use predictions from component classifiers as features for a meta-learning model.

As explained in Xia, Zong, and Li (2011), weighted combinations of feature sets can be quite effective in the task of sentiment classification, since the weights of the ensemble represent the relevance of the different feature sets (e.g. n-grams, POS, etc.) to sentiment classification, instead of assigning relevance to each feature individually. The benefits of rule based ensembles were shown also in Fersini, Messina, and Pozzi (2014), where several variants of voting rules are exhaustively studied in a variety of datasets, with an emphasis on the complexity that results from the use of these approaches. In a different work, Fersini, Messina, and Pozzi (2016) have compared the majority voting rule with other approaches, using three types of subjective signals: adjectives, emoticons, emphatic expressions and expressive elongations. They report that adjectives are more impacting than the other considered signals, and that the average rule is able to ensure better performance than other types of rules. Also, in Xia et al. (2011) a meta-classifier ensemble model is evaluated, obtaining performance improvements as well. An adaptive meta-learning model is described in Aue and Gamon (2005), which offers a relatively low adaptation effort to new domains. Besides, both rule based and meta-learning ensemble models can be enriched with extra knowledge, as illustrated in Xia and Zong (2011). These authors propose the use of a number of rule based ensemble models, namely a sum rule and two weighted combination approaches trained with different loss functions. The base classifiers are trained with n-grams and POS features. These models obtain significant results for cross-domain sentiment classification.

As for the second dimension, *concurrent* models divide the original dataset into several subsets from which multiple classifiers learn in a parallel fashion, creating a classifier composite. The most popular technique that processes the sample concurrently is bagging (Rokach, 2005). Bagging intends to improve the classification by combining the predictions of classifiers built on random subsets of the original data. On the contrary, *sequential* approaches do not divide the dataset but there is an interaction between the learning steps, taking advantage from previous iterations of the learning process to improve the quality of the global classifier. An interesting sequential approach is boosting, which consists in repeatedly training low-performance classifiers on different training data. The classifiers trained in this manner are then combined into a single classifier that can achieve better performance than the component classifiers.

An example of bagging performance in the sentiment analysis task can be found in Sehgal and Song (2007), where bagging and other classification algorithms are used to show that the sentiment evolution and the stock value trend are closely related. Fersini et al. (2014) also show several experimental results in relation to the bagging techniques, attending also to the associated model complexity. Moreover, some authors have shown that bagging techniques are fairly robust to noisy data, while boosting techniques are quite sensitive (Maclin & Opitz, 1997; Melville, Shah, Mihalkova, & Mooney, 2004; Prusa, Khoshgoftaar, & Dittman, 2015). The suitability of bagging and boosting ensembles is also experimentally confirmed by Wang, Sun, Ma, Xu, and Gu (2014). This work also includes the study of a different ensemble technique, random subspace, that consists in modifying the training dataset in the feature space, rather than on the instance space. The authors stand out the better performance of random subspace in comparison with similar approaches, such as bagging and boosting. Another study (Whitehead & Yeager, 2010) shows a comparison between bagging and boosting on a standard opinion mining task. Besides, Lin, Wang, Li, and Zhou (2015) proposes a three phase framework of multiple classifiers, where an optimal subset of classifiers is automatically chosen and trained. This framework is tested in several real-world datasets for sentiment classification.

Nevertheless, these works also show that ensemble techniques not always improve the performance in the sentiment analysis task, and that there is not a global criteria to select a certain ensemble technique.

2.2. Deep learning approaches

In the realm of Natural Language Processing much of the work in deep learning has been oriented towards methods involving learning word vector representations using neural language models (Kim, 2014). Continuous representations of words as vectors has proven to be an effective technique in many NLP tasks, including sentiment analysis (Tang, Wei, Yang et al., 2014). In this sense, *word2vec* is one of the most popular approaches that allows modeling words as vectors (Mikolov, Chen et al., 2013). *Word2vec* is based on the Skip-gram and CBOW models to perform the computation of the distributed representations. While CBOW aims to predict a word given its context, Skip-gram predicts the context given a word. *Word2vec* computes continuous vector representations of words from very large datasets. The computed word vectors retain a huge amount of syntactic and semantic regularities present in the language (Mikolov, Yih, & Zweig, 2013), expressed as relation offsets in the resulting vector space. These word-level embeddings are encoded by column vectors in an embedding matrix $W \in \mathbb{R}^d \times |V|$, where $|V|$ is the size of the vocabulary. Each column $W_i \in \mathbb{R}^d$ corresponds to the word embeddings vector of the i -th word in the vocabulary. The transformation of a word w into its word embedding vector r_w is made by using the matrix-vector

product:

$$r_w = Wv_w$$

where v_w is an one-hot vector of size $|V|$ which has value index at w and zero in the rest. The matrix W components are parameters to be learned, and the dimension of the word vectors d is a hyperparameter to be chosen. The vector representations computed by these techniques can result very effective when used with a traditional classifier (e.g. logistic regression) for sentiment classification, as shown by Zhang, Xu, Su, and Xu (2015). An approach based in *word2vec* is *doc2vec* (Le & Mikolov, 2014), that models entire sentences or documents as vectors. An additional method in representation learning is the auto-encoder, which is a type of artificial neural network applied to unsupervised learning. Auto-encoders have been used for learning new representations on a wide range of machine learning tasks, such as learning representations from distorted data, as illustrated in Chen, Weinberger, Sha, and Bengio (2014).

In deep learning for SA, an interesting approach is to augment the knowledge contained in the embedding vectors with other sources of information. This added information can be sentiment specific word embedding as in Tang, Wei, Yang et al. (2014), or as in a similar work, a concatenation of manually crafted features and these sentiment specific word embeddings (Tang, Wei, Qin, Liu, & Zhou, 2014). In the work presented by Zhang and He (2015) the feature set extracted from word embeddings is enriched with latent topic features, combining them in an ensemble scheme. They also experimentally demonstrate that these enriched representations are effective for improving the performance of polarity classification. Another approach that incorporates new information to the embeddings is described in Su, Xu, Zhang, and Xu (2014), in which deep learning is used to extract sentiment features in conjunction with semantic features. Severn and Moschitti (2015) describe an approach where distant supervised data is used to refine the parameters of the neural network from the unsupervised neural language model. Also, a collaborative filtering algorithm can be used, as is detailed in Kim et al. (2013), where the authors add sentiment information from a small fraction of the data. In the line of adding sentiment information, in Li et al. (2015) is portrayed how a sentiment Recursive Neural Network (RNN) can be used in parallel to another neural network architecture. In general, there is a growing tendency which tries to incorporate additional information to the word embeddings created by deep learning networks. An interesting work is the one described in Vo and Zhang (2015), where both sentiment-driven and standard embeddings are used in conjunction with a variety of pooling functions, in order to extract the target-oriented sentiment of Twitter comments. Enriching the information contained in word embeddings is not the only trend in deep learning for SA. The study of the compositionality in the sentiment classification task has proven to be relevant, as shown by Socher et al. (2013). This work proposes the Recursive Neural Tensor Network (RNTN) model, and it also illustrates that RNTN outperforms previous models on both binary and fine-grained sentiment analysis. The RNTN model represents a phrase using word vectors and a parse tree, computing vectors for higher nodes in the tree using a tensor-based composition function. In relation to the ensemble schemes showed in Section 2.1, some authors (Mesnil, Mikolov, Ranzato, & Bengio, 2014) have used a geometric mean rule to combine three sentiment models: a language model approach, continuous representations of sentences and a weighted BOW. That ensemble exhibits a high performance on sentiment estimation of movie reviews, and better performance than its component classifiers.

To the best of our knowledge, a hybrid approach in which deep learning algorithms, classic feature engineering and ensemble tech-

Table 1

Proposed taxonomy for ensemble of surface and Deep features. S represents surface features, G and A stand for generic word vectors and affect word vectors, respectively. The combination of the features and/or word vectors is indicated with '+'. We consider the combination *No ensemble*/S+G+A not possible since it requires different types of features. Proposed approaches are marked with '*'.

	S	S+G	G	G+A	A	S+A	S+G+A
No ensemble	Pang et al. (2002); Read (2005) Pak and Paroubek (2010); Wang and Manning (2012) Gimpel et al. (2011); Kouloumpis, Wilson, and Moore (2011) Nasukawa and Yi (2003); Taboada et al. (2011) Melville et al. (2009); Qiu et al. (2009) Kiritchenko et al. (2014)	Su et al. (2014), Kim et al. (2013)	M_G* , Shirani-Mehr (2012), Collobert et al. (2011)	Severyn and Moschitti (2015)	Tang, Wei, Yang et al. (2014), Socher et al. (2013)		–
Classifier ensemble	Xia and Zong (2011); Xia et al. (2011), Aue and Gamon (2005); Fersini et al. (2016), Rokach (2005); Sehgal and Song (2007), Prusa et al. (2015); Whitehead and Yaeger (2010) Fersini et al. (2014); Lin et al. (2015) Wang et al. (2014)	CEM_{SG}* , Zhang and He (2015) Mesnil et al. (2014)					CEM_{SGA}*
Feature ensemble	Agarwal et al. (2011); Wilson et al. (2009) Xia and Zong (2010)	M_{SG}*		M_{GA}* , Li et al. (2015), Vo and Zhang (2015)		Tang, Wei, Qin et al. (2014)	M_{SGA}*

niques for sentiment analysis are used has not been thoroughly studied.

3. Ensemble taxonomy

This section presents the proposed taxonomy for ensemble techniques applied to Sentiment Analysis in both surface and deep domains. This classification intends to summarize the work found in the literature as well as to compare these models with the ones we propose. Also, with this, we address the first question raised in Section 1 regarding how combination techniques can be classified.

The taxonomy can be expressed as combination of two different dimensions. Each dimension represents a characteristic of the studied approaches. On the one hand, one dimension considers which features are used in the model. Those features can be either surface features (which stands for S), generic automatic word vectors (G), or affect word vectors specifically trained for the sentiment analysis task (A). On the other hand, the other dimension attends to how the different model resources are combined. These combinations can be: using no ensemble method at all, through a ensemble of classifiers, or taking advantage of a feature ensemble. Table 1 shows a representation of this taxonomy, where the two dimensions appear as rows for the first dimension, and columns for the second dimension. We have classified all the reviewed work in this paper using the proposed taxonomy, obtaining a visual layout of the techniques that are used in each approach in relation with both ensemble methods and the combination of surface and deep features.

Regarding the dimension that tackles the ensemble techniques, in the *No ensemble* category we find the classifiers that do not make use of an ensemble technique. Under the *Classifier ensemble* category we classify the approaches that are based on ensemble

techniques (Section 2.1), such as the voting rule or a meta-learning technique, to name a few. In the same manner, the *Feature ensemble* category contains the approaches that make use of feature combination techniques. The feature ensemble consists in combining different set of features into a unified set that is then fed to a learning algorithm.

As for the dimension that represents which features are used, several possibilities are represented: only surface features, generic or affect words vectors (S, G and A respectively), where only one type of feature is used. Besides, this dimension also takes into account the combination of different types of features: S+G (surface features combined with generic word vectors), G+A (generics word vectors with affect embeddings), S+A (surface features combined with affect word embeddings), and S+G+A (all three types of features combined in the same model).

These two dimensions are combined, creating a grid where the different approaches can be classified. The blank spaces in the taxonomy represent techniques that, to the extent of our knowledge, have not been studied. As such, they represent work that can be addressed in the future.

In conclusion, the introduced taxonomy provides a framework for characterizing and comparing ensemble approaches in sentiment analysis. This framework provides us with the opportunity to characterize and compare existing research works in sentiment analysis using ensemble techniques. Moreover, the framework can help us to provide guidelines to choose the most efficient and appropriate ensemble method for a specific application.

4. Sentiment analysis models

This section presents the sentiment analysis models proposed in our work. These models have been validated in the Twitter and

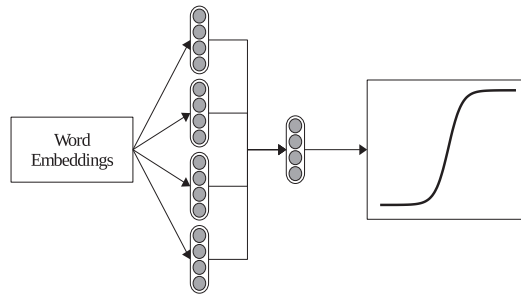


Fig. 1. Schematic representation of the baseline model, M_G . The word vectors are combined into a fixed dimension vector, and then fed to a logistic regressor, which determines the polarity of the document.

movie reviews domains (Section 5). First we describe the developed deep learning based analyzer used as baseline for the rest of the paper, and after this we detail the proposed ensemble models. These models are: ensemble of classifiers and ensemble of features. Regarding the ensemble of classifiers, we tackle two main approaches in further experiments: fixed rule and meta-learning models.

4.1. Deep learning classifier (M_G)

Generic word vectors, also denoted as pre-trained word vectors, can be captured by word embeddings techniques such as *word2vec* (Mikolov, Chen et al., 2013) and *GloVe* (Pennington, Socher, & Manning, 2014). Generic vectors are extracted in an unsupervised manner i.e., they are not trained for a specific task. These word vectors contain semantic and syntactic information, but do not enclose any specific sentiment information. Nevertheless, with the intention of exploiting the information contained in these generic word vectors, we have developed a sentiment analyzer model based on deep word embedding techniques for feature extraction, in order to compare it to other approaches in the task of Sentiment Analysis. The computed word vectors are combined into a unique vector of fixed dimension that represents the whole message. Then, this vector is fed to a logistic regression algorithm. The computation of the combined vector can be made using a set of convolutional functions, or using an embedding that transforms documents into a vector. In this way, the proposed baseline model codification is input dependent, as can be seen in Section 5. In this paper, we propose the use of *word2vec* for short texts, where the word vectors of the text are combined with convolutional functions; and *doc2vec* for long ones, representing each document with a vector. The combination of word vectors from short texts are obtained through the *min*, *average* and *max* convolutional functions. These functions may be combined through the concatenation of its resulting vectors. The combination of n of these functions produces a vector of nd dimensions, where d is the original dimension of the word vectors.

A diagram of this model is shown in Fig. 1. We refer to this model as M_G , with the G standing for generic word vectors.

4.2. Ensemble of classifiers (CEM)

Our objective is to combine the information from surface and deep features. The most straightforward method is to combine them at the classification level. In this way, we propose an ensemble model which combines classifiers trained with deep and surface features. Thus, knowledge from the two sets of features is combined, and this composition has more information than its

base components. This model combines several base classifiers which make predictions from the same input data. These predictions can be subsequently used as new data for extracting a single prediction of sentiment polarity. This ensemble model aims to improve the sentiment classification performance that each base classifier can achieve individually, obtaining better performance. There are many possibilities for the combination of the base classifiers predictions that outputs a final sentiment polarity (e.g. a fixed rule or a meta learning technique). Also, any number of base classifiers can be combined into this ensemble model. A schematic diagram of this proposal is illustrated in Fig. 2. We denote this model as CEM, which stands for Classifier Ensemble Model. The subscript indicates the types of features its base classifiers have been trained with, like in CEM_{SG} , where the ensemble combines classifiers trained with surface features and generic word vectors.

Next, the two ensemble techniques used in this ensemble model in the experimentation section are further described.

4.2.1. Fixed rule model

This model seeks to combine the predictions from different classifiers using a simple fixed rule. Consequently, this ensemble does not need to learn from examples. The rules used in this approach can be any fixed rule used in ensemble models. In this work the rule used for the ensemble is the voting rule by majority. This rule counts the predictions of component classifiers and assigns the input to the class with most component predictions. In case of a match, a fixed class can be selected as the predicted by the model.

4.2.2. Meta classifier model

In the meta-classifier technique, the outputs of the component classifiers are treated as features for a meta-learning model. One advantage of this approach is that it can learn, i.e. adapt to different situations. As for the selection of the learning algorithm of this approach, there is no indication as of which one should be used. In this work, we select the Random Forest algorithm, as it can achieve high performance metrics in sentiment analysis (da Silva, Hruschka, & Hruschka, 2014; Zhang et al., 2011).

4.3. Ensemble of features (M_{SG} and M_{CA})

This model is proposed with the aim of combining several types of features into a unified feature set and, consequently, combine the information these features give. In this way, a learning model that learns from this unified set could achieve better performance scores than one that learns from a feature subset.

In this sense, we can distinguish two main types of ensembles of features. The first type is the ensemble of features that combines both surface and deep learning features. We address to this first model type as M_{SG} , as it combines surface features and generic word vectors. The second type consists on an ensemble of features that were completely extracted using deep learning techniques. This second type is referred as M_{CA} , combining both generic and affect word vectors. We refer to affect vectors as the result from training a set of pre-trained word vectors for a specific task, which in this case it would be SA.

Additionally, we also propose a third feature ensemble model, where all the three types of features are combined. This model, where surface features, generic word vectors and affect word vectors are combined is denoted by M_{SGA} . A diagram representing two instances of the model is shown in Fig. 2.

5. Experimental study

This section describes the experiments conducted in order to answer the questions formulated in the introduction (Section 1).

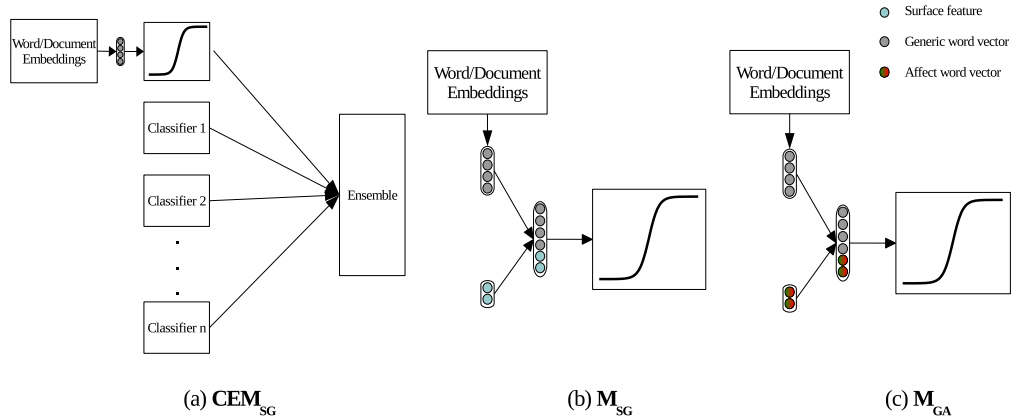


Fig. 2. Diagram of how the different classifiers and features are combined in the CEM_{SG} (Classifier Ensemble Model combining surface features and generic word vectors), M_{SG} and M_{GA} models.

Table 2

Statistics of the SemEval2014/2014, Vader, STS-Gold and Sentiment140 datasets.

Dataset	Positive	Negative	Total	Average #words
SemEval2013	2315	861	3176	23
SemEval2014	2509	932	3441	22
Vader	2901	1299	4200	16
STS-Gold	632	1402	2034	16
Sentiment140	800,000	800,000	1,600,000	15
IMDB	25,000	25,000	50,000	255
PL04	1000	1000	2000	723

Each performance experiment is made with six different datasets, widely used by the community of Sentiment Analysis. The metric used in this work is the macro averaged F1-Score. Accuracy, Precision and Recall are also computed for all the experiments, and the interested reader can find these results in the web¹. We also publish the computed vectors that have been used in the deep models.

These experiments (Section 6.2) are aimed to compare the performance between the deep learning baseline we have developed (M_G) and the proposed ensemble models. Also, some experiments (Section 6.1) are also aimed to characterize the sentiment analysis performance for each individual classifier of the CEM models. For the last purpose, we have collected several sentiment analyzers for composing a classifier ensemble.

As for the sentiment analysis of natural language, it is conducted at the message level, so it is not necessary to split the input data into sentences. The classifiers label each comment as either positive or negative.

5.1. Datasets

The datasets used for testing are *SemEval 2013*, *SemEval 2014* (Rosenthal, 2014), *Vader* (Hutto, 2015), *STS-Gold* (Saif, Fernandez, He, & Alani, 2013), *IMDB* (Mass et al., 2011) and *PL04* (Pang et al., 2002). Also, we use the *Sentiment 140* (Go, Bhayani, & Huang, 2009) and *IMDB* datasets for training and developing our deep learning baseline, M_G . These datasets are described next, and some statistics are summarized in Table 2.

The *SemEval 2013* test corpus is composed of English comments extracted from Twitter on a range of topics: several entities, prod-

ucts and events. Similarly, we have also use the *SemEval 2014* test dataset. In both *SemEval* datasets, the data is not public but must be downloaded from the source first. As some users have already deleted their comments online, we have not been able to recover the original datasets, but subsets of it. Besides, since the development dataset contains only binary targets (positive and negative), we have made an alignment processing of the *SemEval* datasets, filtering other polarity values. The obtained sizes are detailed in Table 2.

The *Vader* dataset contains 4200 tweet-like messages, originally inspired by real Twitter comments. A subset of these messages is specifically designed to test some syntactical and grammatical features that appear in the natural language. The *STS-Gold* dataset for Twitter, which has been collected as a complement for Twitter sentiment analysis evaluations processes (Saif et al., 2013).

As for the training data of our Twitter baseline model, the selected dataset is the *Sentiment 140* dataset, containing 1,600,000 Twitter messages extracted using a distant supervision approach (Go et al., 2009). The abundance of data in this dataset is very beneficial to our deep learning approach, as it requires large quantities of data to extract a fairly good model, as pointed out by Mikolov, Chen et al. (2013).

Regarding the movie reviews domain, *IMDB* contains 50,000 polarized messages, using the score of each review as a guide for the polarity value. Besides, this dataset contains 50,000 unlabeled messages that have been used for training the movie reviews baseline model. We use this dataset for the training of the movie reviews baseline model. The *PL04* dataset is a well-known dataset in this domain. For the results in this dataset, we report the 10-fold cross validation metrics using the authors' public folds, in order to make our results comparable with the ones found in the literature.

5.2. Baseline training

Due to the different characteristics of the two studied domains (Twitter and movie reviews), the vector computing process for the baseline model has been made differently. In the Twitter domain, the word vectors computed by word2vec are combined using convolutional functions. For the movie reviews domain, doc2vec is used for the combination of the word vectors. For the implementation of this model, we use the gensim library (Řehůřek & Sojka, 2010).

We found that the use of the convolutional functions in large text documents does not yield better performance than doc2vec

¹ <http://gsi.dit.upm.es/~oaraque/enhancing-dl>.

Table 3
Effectiveness of the convolutional functions on the Sentiment140 development dataset.

Convolutional function	F-Score
<i>max</i>	74.82
<i>avg</i>	77.53
<i>min</i>	74.99
<i>max + avg</i>	77.63
<i>max + min</i>	76.7
<i>avg + min</i>	77.70
<i>max + avg + min</i>	77.73

combinations. Hence, we performed an evaluation of these two approaches on the two development datasets. While the convolutional combinations yield a F1 score of 77.53% in the Sentiment140 dataset, they also achieve 73.66% in the IMDB dataset. When using doc2vec the F1 scores are 75.00% and 89.45% in the Sentiment140 and IMDB datasets respectively. Considering the average number of words presented in Table 2 and the difference on the performance of each approach, we use the convolutional functions for the twitter domain, where short texts are analyzed and the doc2vec technique for the movie reviews domain, which contains large documents.

Regarding the training process for the short text word embeddings, we empirically fixed the dimension of the word vectors generated to 500. We use 1,280,000 tweets randomly selected from the Sentiment 140 dataset. Once this model is extracted, we feed a logistic regression model (implementation from scikit-learn) with the vectors of each tweet and the labels from the original dataset. The movie reviews baseline has been similarly trained, with the 50,000 unsupervised documents of the IMDB dataset, setting the dimension of the document vectors to 100. The same linear model is used for the classification of the document vectors. All the performance metrics have been obtained using K-fold cross validation, with folds of 10.

With respect to the convolutional functions, we have conducted an effectiveness test of the *max*, *average* and *min* functions on the Sentiment140 development set. The results are shown in Table 3. As can be seen, the *avg* function is very close to the performance of the complete set of functions *max*, *avg* and *min*. Consequently, we select the *avg* function as the one used for further experiments, as it provides very good results compared to the rest, and it also reduces the computational complexity of the experimentation. No pooling functions are used in the movie reviews, as there is no need to combine different word vectors.

Lastly, the preprocessing of natural language, we tokenized the input data and removed punctuation, excepting the most common ('.', '!', '?'). We also transformed URLs, numbers and usernames (@username) into especial characters to normalize the data. The preprocessing is applied to all the texts before generating the word vectors.

5.3. Ensemble of classifiers

In order to improve the performance of the deep learning baseline, we have built an ensemble composed of this analyzer and six different sentiment classifiers. Following, a list and a brief description of each of these classifiers is shown:

- sentiment140 (Go et al., 2009). It uses Naive Bayes, Maximum Entropy and Support Vector Machines trained with unigrams, bigrams and POS features.
- Stanford CoreNLP (Manning et al., 2014) is the RNTN approach shown in Section 2.2, proposed by Socher et al. (2013).

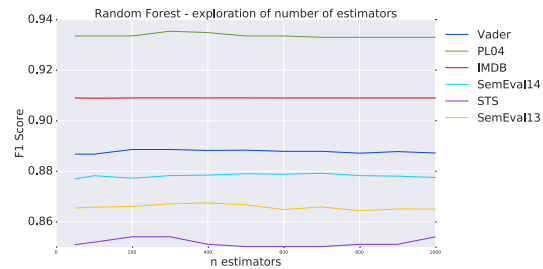


Fig. 3. Cross validation of the number of estimators on the Random Forest algorithm used for the meta-learning ensemble.

- Sentiment WSD (Kathuria, 2015), which uses SentiWordNet (Esuli & Sebastiani, 2006), performing the sentiment estimation based on the polarities of each word.
- Vivekn (Narayanan, Arora, & Bhatia, 2013). It is based in a Naive Bayes classifier trained with word n-grams and using several techniques, such as negation handling, feature selection and laplacian smoothing.
- pattern.en (De Smedt & Daelemans, 2012) uses a Support Vector Machines algorithm fed with polarity and subjectivity values for each word, WordNet vocabulary information and POS annotation.
- TextBlob Sentiment Classifier (Loria, 2016), a modular approach which as default configuration uses a Naive Bayes classifier trained with unigram features.

We have built ensemble classifiers using two combining techniques in the CEM model: a rule based method and a meta learning approach, both using the predictions of the classifiers composing the ensemble as features for the next step. For the meta-learning approach, we use the implementation of scikit-learn of the Random Forest algorithm. For this algorithm we have used 100 as the default number of estimators. As is shown in Fig. 3, the value of this parameter does not affect to the classification performance in the range from 50 to 1000.

Additionally, two versions of the CEM model have been implemented for the experiments. While the CEM_{SG} combines the six aforementioned classifiers and the M_G model; the CEM_{SGA} version combines the base classifiers from CEM_{SG} with the M_{SG}, M_{SG+bigrams} and M_{GA} models.

5.4. Ensemble of features

Based on the work by Mohammad, Kiritchenko, and Zhu (2013), we have selected the following surface features: SentiWordnet (Esuli & Sebastiani, 2006) lexicon values for each word, as well as total number of positive, neutral and negative words extracted with this lexicon; number of exclamation, interrogation and hash-tags marks '!?', '#'; number of words that are all in caps and number of words that have been elongated 'goood'. This feature set has been cross validated on the development sets, with the objective of obtaining the smaller surface feature set that yields the best classification performance.

With the aim of complementing the surface features, we have also explored the role of n-grams. More specifically, bigrams are used, as the introduction of unigrams and trigrams did not improve the classification performance. The ensemble of features model that includes generic word vectors, the described surface features and bigrams is represented as M_{SG+bigrams}.

As for the M_{GA} model, we use the word vectors obtained by Tang, Wei, Yang et al. (2014). More specifically, we use the vectors extracted using the SSWE_r neural model. These vectors have been

Table 4

Macro averaged F-Score of all the base sentiment classifiers. TB represents the TextBlob classifier.

Dataset	sent140	CoreNLP	WSD	vivekn	pattern	TB
SemEval2013	78.92	46.95	76.18	72.14	82.50	82.51
SemEval2014	60.67	42.95	75.35	59.97	71.86	71.92
Vader	78.76	60.19	77.75	63.54	85.98	85.71
STS-Gold	75.07	59.68	75.35	69.61	82.86	68.27
IMDB	72.91	88.56	87.41	87.45	75.43	75.47
PL04	71.66	86.09	79.03	86.22	70.64	70.82

extracted for learning sentiment-specific word vectors but not general semantic information, so we use them as affect word vectors. Also, for the composition of a tweet vector, we have used the average function on the word vectors, as the combination of the other convolutional functions did not improve the results of the model.

6. Results

The conducted experiments show the sentiment classification performance of each base classifier separately (including our deep learning baseline) on each of the six test datasets, as well as the metrics for the ensemble of classifiers and ensembles of features. In this section, the experimental results are shown and discussed. The experimental results for the proposed models are gathered in Table 5.

6.1. Base classifiers performance

As it can be seen in Table 4, the better F-score performance is achieved by TextBlob in SemEval2013, by the WSD classifier with an important difference over TextBlob in SemEval2014; while pattern.en has a slightly better performance than the rest in the Vader dataset and has also the better performance in the STS-Gold dataset by far. The classifier with the lower performance is CoreNLP in all Twitter test sets. In contrast, in the IMDB and PL04 datasets (movie reviews), CoreNLP achieves the better performance in IMDB and the second best in PL04.

As expected, the nature of the domains has a strong impact on the performance of the base classifiers, since they have been trained in a specific domain. Some classifiers (e.g. WSD, pattern and TextBlob) are more suitable to the Twitter domain, while others (e.g. CoreNLP and vivekn) are better adapted to the review domain. In general, none of the base classifiers exhibits a high performance in all the baseline datasets.

Finally, the average F1 score performance for the base classifiers is 73.02, 63.79, 75.32, 71.81, 81.20 and 77.41% in the SemEval2013, SemEval2014, Vader, STS-Gold, IMDB and PL04 datasets respectively.

6.2. Classifiers and features classifiers performance

CEM models gather the predictions from M_C baseline and the other six base classifiers whose classification performance has

been analyzed. The voting and meta-learning techniques are used as ensemble techniques. It can be seen in the Table 5 that nearly all the ensemble models surpass the baseline, as well as all the other base classifiers. In fact, the best performance is achieved in 4 out of 6 datasets by these CEM classifiers.

As for the feature ensemble models, they also push the performance further than the baseline. The $M_{SG+bigrams}$ ensemble is very close to the best metrics in almost all the test datasets. Also, it seems that $M_{SGA+bigrams}$ is suffering overfitting, as the combination of all three types of features decreases the performance when comparing to $M_{SG+bigrams}$ model. This could be due to the increase in the number of features used to train the model.

Moreover, as an additional observation, it can be seen that the better improvements are achieved by CEM_{SG}^{Vo} and CEM_{SG}^{Mel} models, with 3.65 and 2.53% of performance gain in STS-Gold and SemEval2013 datasets respectively, and by CEM_{SGA}^{Mel} model in the IMDB and PL04 datasets, with improvements of 1.48% and 5.77% respectively.

Although the biggest improvements have been achieved with CEM models, the feature ensemble models also improve the baseline, sometimes by a large margin. Considering this type of models, the better results are achieved by the $M_{SG+bigrams}$ model. This fact could be explained attending to fact that bigrams can successfully capture modified verbs and nouns (Wang & Manning, 2012), such as the negation.

Also, the M_{SG} model results are comparable, generally, to the best performances in the Twitter domain. Nevertheless, this model does not yield such results in the movie reviews domain. This result indicates that combining word vectors through convolutional functions in long texts does not lead to high sentiment classification performances. We can conclude that the transformation of the convolutional functions on the sentiment signals that are contained in the affect word vectors is retained when applied to short texts, but lost in long texts.

Attending to the difference of performance between the $M_{SGA+bigrams}$ and the CEM_{SGA}^{Vo} and CEM_{SGA}^{Mel} models, we see that the same types of features do not yield the same result. We make the assumption that dividing the features into smaller sets, as it is done in the ensemble models, benefits the classification performance. The division is made at the classifier level, since these ensemble models combine the predictions of classifiers trained with features (e.g. surface, generic and affect). Considering that the whole set SGA (including bigrams) is a complex collection of features, and based on the experimental results, the assumption is that this division of features prevents overfitting. These results are in line with those by Alpaydin (2014) and Xia et al. (2011). This shows that when dividing a complex set of features into simpler subsets, an ensemble can yield better performance.

6.3. Statistical analysis

In order to compare the different proposed models in this work, a statistical test has been applied on the experimental results. Concretely, the Friedman test with the corresponding Bonferroni-Dun

Table 5

Macro averaged F-Score of the proposed sentiment classifiers. The last row shows the Friedman rank. In bold the best classifier for each dataset, and in the last row the best classifier attending to the Bonferroni-Dunn test.

Dataset	M_C	CEM_{SG}^{Vo}	CEM_{SG}^{Mel}	M_{SG}	$M_{SG+bigrams}$	M_{GA}	$M_{SGA+bigrams}$	CEM_{SGA}^{Vo}	CEM_{SGA}^{Mel}
SemEval2013	85.34	87.78	87.87	86.36	86.53	87.54	86.26	86.26	86.97
SemEval2014	86.16	84.16	87.63	87.03	88.19	88.05	86.94	85.90	88.07
Vader	87.71	87.92	88.85	88.07	88.93	88.89	88.89	89.52	89.48
STS-Gold	83.43	83.52	84.56	84.73	89.24	85.27	85.26	87.08	85.59
IMDB	89.45	84.06	89.68	89.58	90.41	89.50	90.41	89.92	90.93
PL04	88.72	86.48	93.65	88.39	94.33	88.67	86.76	93.87	94.49
Friedman rank	7.83	7.5	4.5	6.17	2.67	4.33	5.58	4.25	2.17

post-hoc test, that are described by Demšar (2006). These tests are specially oriented to the comparison of several classifiers on multiple data sets.

Friedman's test is based on the rank of each algorithm in each dataset, where the best performing algorithm gets the rank of 1, the second best gets rank 2, etc. Ties are resolved by averaging their ranks. r_j^i is the rank of the j th of the k algorithms and on the i -th of N datasets. Friedman test uses the comparison of average ranks $R_j = \frac{1}{N} \sum_i r_j^i$, and states that under the null-hypothesis (all the algorithms are equal so their ranks R_j are equal) the Friedman statistic, with $k - 1$ degrees of freedom, is:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left(\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right)$$

Nevertheless, Demšar (2006) shows that there is a better statistic that is distributed according to the F-distribution, and has $k - 1$ and $(k - 1)(N - 1)$ degrees of freedom:

$$F_F = \frac{(N - 1)\chi_F^2}{N(k - 1) - \chi_F^2}$$

If the null-hypothesis of the Friedman test is rejected, post-hoc tests can be conducted. In this work we employ the Bonferroni-Dunn test, as it allows to compare the results of several algorithms to a baseline. In this case, all the proposed models are compared against M_G . This test can be computed through comparing the critical difference (CD) with a series of critical values (q_α), which Demšar (2006) summarizes. The critical difference can be computed as:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

For the computation of both tests, the ranks have been obtained. The average ranks (R_j) are showed in Table 5. The α values is set to 0.05 for the following calculations. With those averages, $\chi_F^2 = 24.56$, $F_F = 5.24$, and the critical value $F(k - 1, (k - 1)(N - 1)) = 2.18$. As $F_F > F(8, 40)$, the null-hypothesis is rejected and the post-hoc test can be conducted. According with this, the critical difference is 4.31. Following, the difference between the average ranks of the baseline and the j th model is compared to the CD and, if greater, we can conclude that the j th algorithm performs significantly better than the baseline.

Friedman's test has pointed the CEM_{SGA}^{Mel} and the $M_{SG+bigrams}$ models as the two best classification models, followed by the CEM_{SGA}^{Vo} and M_{CA} models. Besides, the Bonferroni-Dunn test points out that CEM_{SGA}^{Mel} and $M_{SG+bigrams}$ models perform significantly better than the baseline. These results indicate that the hypothesis suggested in question 2 is supported, since the combination of different sources of information improves the performance of sentiment analysis. As for the rest of the models, it is not possible to reach a conclusion with the current data.

On top of this, an interesting result of these experiments is that the performance of the meta learning approach is higher than that of the fixed rule scheme. While the meta learning ensemble with all types of features (SGA + bigrams) is significantly better than the baseline, the voting model is not. This could be caused by the learning capabilities of the meta-classifier technique, feature that the fixed ensemble methods like voting rule do not have.

6.4. Computational complexity

One possible drawback of ensemble approaches is their higher cost in terms of computational resources. With the aim of analyzing the efficiency of the proposed models, the computational cost is presented. The results for this cost analysis are studied at train time, since the costs in the test phase do not result relevant for

Table 6

Computational complexity of the word embeddings approaches, in both model training and vector computation.

Word Embeddings approach		Sentiment140	IMDB
word2vec	Train model	109.7 s	96.1 s
	Compute vectors	152.9 s	80.4 s
doc2vec	Train model	7 h	2 h
	Compute vectors	8.5 s	6.4 s
SSWE	Train model	–	25 d
	Compute vectors	87.5 s	179.2 s

Table 7

Computational complexity of the proposed models in training time.

Model	Sentiment140	IMDB
M_G	12.7 s	0.9 s
M_{SG}	13.7 s	0.9 s
$M_{SG+bigrams}$	977.5 s	37.2 s
M_{CA}	12.9 s	1 s
$M_{SGA+bigrams}$	977.8 s	37.4 s

this analysis. All these measurements were made in a Intel Xeon with 12 cores available and a memory friendly environment.

In relation to the training and computation of the word embeddings approaches, Table 6 presents the associated computational cost. It can be seen that word2vec is the lighter model at train time by a large margin. Also, the implementation of SSWE largely increases the computational complexity. We believe that implementing this model for a GPU environment can have a great impact on the time performance of the SSWE training. Please note that the SSWE trained model for Twitter is available for research, and so the training using the Sentiment140 dataset has not been performed. Besides, we can see that computing the pooling functions on the word vectors increases the complexity, as can be seen by comparing with the doc2vec approach.

Combining different sets of features increases the computational complexity, as Table 7 shows. The largest increment can be found in the $M_{SG+bigrams}$ and $M_{SGA+bigrams}$ models, which use bigrams in the learning process. In this way, it can be seen that using feature ensemble with bigrams and other sets of features leads to the addition of complexity to the model. The difference of training times between the Sentiment140 and IMDB datasets is due to their number of instances, being larger in the first.

Finally, the CEM models do not introduce a relevant complexity to the model at training time. The ensemble of classifiers based on the voting scheme do hardly introduce a cost to the computation, as there is no learning process in this case. For the meta-learning scheme, the maximum time taken in the meta learning process is 1.5 s, with no significant difference between the training data.

7. Conclusions and future work

This paper proposes several models where classic hand-crafted features are combined with automatically extracted embedding features, as well as the ensemble of analyzers that learn from these varied features. In order to classify these different approaches, we propose a taxonomy of ensembles of classifiers and features that is based on two dimensions. Furthermore, with the aim of evaluating the proposed models, a deep learning baseline is defined, and the classification performances of several ensembles are compared to the performance of the baseline. With the intention of conducting a comparative experimental study, six public datasets are used for the evaluation of all the models, as well as six public sentiment classifiers. Finally, we conduct an statistical analysis in order to empirically verify that combining information from varied fea-

tures and/or analyzers is an adequate way to surpass the sentiment classification performance.

There were three main research questions that drove this work. The first question was whether there is a framework to characterize existing approaches in relation to the ensemble of traditional and deep techniques in sentiment analysis. To the best of our knowledge, our proposal of a taxonomy and the resulting implementations is the first work to tackle this problem for sentiment analysis.

The second question was whether the sentiment analysis performance of a deep classifier can be leveraged when using the proposed ensemble of classifiers and features models. Observing the scores table and the Friedman ranks (Table 5), we see that the proposed models generally improve the performance of the baseline. This indicates that the combination of information from diverse sources such as surface features, generic and affect word vectors effectively improves the classifier's results in sentiment analysis tasks.

Lastly, we raised the concern of which of the proposed models stand out in the improvement of the deep sentiment analysis performance. In this regard, the statistical results point out the CEM_{SG}^{MeL} and $M_{SG+bigrams}$ models as the best performing alternatives. As expected, these models effectively combine different sources of sentiment information, resulting in a significant improvement with respect to the baseline. We remark the $M_{SG+bigrams}$ model, as it does not involve an ensemble of many classifiers, but only a classifier that is trained with an ensemble of deep and surface features.

To summarize, this work takes advantage of the ensemble of existing traditional sentiment classifiers, as well as the combination of generic, sentiment-trained word embeddings and manually crafted features. Nevertheless, Considering the results of this work, we believe that a possible line of work would be applying these models to the task of aspect based sentiment analysis, with the hope of improving the classification performance. Furthermore, we intend to extend the domain of the proposed models to other languages and even paradigms, like Emotion analysis.

Acknowledgements

This research work is supported by the EC through the H2020 project MixedEmotions (Grant Agreement no: 141111), the Spanish Ministry of Economy under the R&D project Semola (TEC2015-68284-R) and the project EmoSpaces (RTC-2016-5053-7); by ITEA3 project SOMEDI (15011); and by MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

References

- Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the workshop on languages in social media* (pp. 30–38). Association for Computational Linguistics.
- Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.
- Aue, A., & Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of recent advances in natural language processing (RANLP)*: vol. 1, 2–1.
- Bengio, Y. (2009). Learning deep architectures for ai. *Foundations and Trends® in Machine Learning*, 2, 1–127.
- Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31, 102–107.
- Chen, M., Weinberger, K. Q., Sha, F., & Bengio, Y. (2014). Marginalized denoising auto-encoders for nonlinear representations. In *ICML* (pp. 1476–1484).
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12, 2493–2537.
- De Smedt, T., & Daelemans, W. (2012). Pattern by python. *The Journal of Machine Learning Research*, 13, 2063–2067.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, 1–30.
- Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining WSDM '08* (pp. 231–240). New York, NY, USA: ACM. doi:10.1145/1341531.1341561.
- Esuli, A., & Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC: 6* (pp. 417–422). Citeseer.
- Fersini, E., Messina, E., & Pozzi, F. (2014). Sentiment analysis: Bayesian ensemble learning. *Decision Support Systems*, 68, 26–38. doi:10.1016/j.dss.2014.10.004. <http://www.sciencedirect.com/science/article/pii/S0167923614002498>.
- Fersini, E., Messina, E., & Pozzi, F. (2016). Expressive signals in social media languages to improve polarity detection. *Information Processing & Management*, 52, 20–35.
- Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., et al. (2011). Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies: Short papers - volume 2 HLT '11* (pp. 42–47). Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=2002736.2002747>.
- Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford, 1, 12.
- Hutto, C. (2015). *Vader sentiment github repository*. <https://github.com/cjhutto/vaderSentiment>. Accessed on May 30, 2016.
- Kathuria, P. (2015). *Sentiment wsd github repository*. https://github.com/kevincobain2000/sentiment_classifier/. Accessed on May 30, 2016.
- Kim, J., Yoo, J.-B., Lim, H., Qiu, H., Kozareva, Z., & Galstyan, A. (2013). Sentiment prediction using collaborative filtering. *ICWSM*.
- Kim, Y. (2014). *Convolutional neural networks for sentence classification*. arXiv preprint arXiv:1408.5882.
- Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 723–762.
- Kouloumpis, E., Wilson, T., & Moore, J. D. (2011). Twitter sentiment analysis: The good the bad and the omg!. *ICWSM*, 11, 538–541.
- Le, Q. V., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML: vol. 14* (pp. 1188–1196).
- Li, C., Xu, B., Wu, G., He, S., Tian, G., & Zhou, Y. (2015). Parallel recursive deep model for sentiment analysis. In T. Cao, E.-P. Lim, Z.-H. Zhou, T.-B. Ho, D. Cheung, & H. Motoda (Eds.), *Advances in knowledge discovery and data mining. In Lecture Notes in Computer Science: Vol. 9078* (pp. 15–26). Springer International Publishing. doi:10.1007/978-3-319-18032-8_2. http://dx.doi.org/10.1007/978-3-319-18032-8_2.
- Lin, Y., Wang, X., Li, Y., & Zhou, A. (2015). Integrating the optimal classifier set for sentiment analysis. *Social Network Analysis and Mining*, 5, 1–13.
- Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.
- Loria, S. (2016). *Textblob documentation page*. <https://textblob.readthedocs.org/en/dev/index.html>. Accessed on May 30, 2016.
- Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Vol-1* (pp. 142–150). Association for Computational Linguistics.
- Maclin, R., & Opitz, D. (1997). An empirical evaluation of bagging and boosting. *AAAI/IAAI*, 1997, 546–551.
- Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., & McClosky, D. (2014). The stanford CoreNLP natural language processing toolkit. In *Association for computational linguistics (ACL) system demonstrations* (pp. 55–60). <http://www.aclweb.org/anthology/P/P14/P14-5010>.
- Melville, P., Gryc, W., & Lawrence, R. D. (2009). Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining KDD '09* (pp. 1275–1284). New York, NY, USA: ACM. doi:10.1145/1557019.1557156. <http://doi.acm.org/10.1145/1557019.1557156>.
- Melville, P., Shah, N., Mihalkova, L., & Mooney, R. J. (2004). Experiments on ensembles with missing and noisy data. In F. Roli, J. Kittler, & T. Windeatt (Eds.), *Multiple classifier systems: 5th international workshop, MCS 2004, Cagliari, Italy, June 9–11, 2004. proceedings* (pp. 293–302). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-25966-4_29. http://dx.doi.org/10.1007/978-3-540-25966-4_29.
- Mesnil, G., Mikolov, T., Ranzato, M., & Bengio, Y. (2014). Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *arXiv preprint arXiv:1412.5335*.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *HLT-NAACL* (pp. 746–751).
- Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242. <http://arxiv.org/abs/1308.6242>.
- Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced naive bayes model. *CoRR*, abs/1305.6143. <http://arxiv.org/abs/1305.6143>.
- Nasukawa, T., & Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture* (pp. 70–77). ACM.

- Pablos, A. G., Cuadros, M., & Rigau, G. (2016). A comparison of domain-based word polarity estimation using different word embeddings. In N. C. C. Chair, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the tenth international conference on language resources and evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).
- Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*: vol. 10 (pp. 1320–1326).
- Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on empirical methods in natural language processing-Volume 10* (pp. 79–86). Association for Computational Linguistics.
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*: vol. 14 (pp. 1532–1543).
- Perez-Rosas, V., Banea, C., & Mihalcea, R. (2012). Learning sentiment lexicons in spanish. In *LREC*: vol. 12 (p. 73).
- Polanyi, L., & Zaenen, A. (2006). Computing attitude and affect in text: Theory and applications. In *chapter Contextual Valence Shifters* (pp. 1–10). Dordrecht: Springer Netherlands. http://dx.doi.org/10.1007/1-4020-4102-0_1.
- Prusa, J., Khoshgoftaar, T., & Dittman, D. (2015). Using ensemble learners to improve classifier performance on tweet sentiment data. In *Information reuse and integration (IRI), 2015 IEEE international conference on* (pp. 252–257). doi:10.1109/IRI.2015.49.
- Qiu, G., Liu, B., Bu, J., & Chen, C. (2009). Expanding domain sentiment lexicon through double propagation. In *IJCAI*: vol. 9 (pp. 1199–1204).
- Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL student research workshop* (pp. 43–48). Association for Computational Linguistics.
- Řehůřek, R., & Sojka, P. (2010). Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks* (pp. 45–50). Valletta, Malta: ELRA. <http://is.muni.cz/publication/884893/en>.
- Rokach, L. (2005). Ensemble methods for classifiers. In O. Maimon, & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 957–980). Springer US. http://dx.doi.org/10.1007/0-387-25465-X_45.
- Rosenthal, S. (2014). *Semeval 2014 task 9 description*. <http://alt.qcri.org/semeval2014/task9/>. Accessed on May 30, 2016.
- Saif, H., Fernandez, M., He, Y., & Alani, H. (2013). *Evaluation datasets for twitter sentiment analysis: a survey and a new dataset, the sts-gold*.
- Sehgal, V., & Song, C. (2007). Sops: Stock prediction using web sentiment. In *Data mining workshops, 2007. ICDM workshops 2007. seventh IEEE international conference on* (pp. 21–26). doi:10.1109/ICDMW.2007.100.
- Severyn, A., & Moschitti, A. (2015). Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 959–962). ACM.
- Sharma, A., & Dey, S. (2012). A comparative study of feature selection and machine learning techniques for sentiment analysis. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium* (pp. 1–7). ACM.
- Shirani-Mehr, H. (2012). *Applications of deep learning to sentiment analysis of movie reviews*.
- da Silva, N. F., Hruschka, E. R., & Hruschka, E. R. (2014). Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66, 170–179.
- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., & Ng, A. Y. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*: 1631 (p. 1642). Citeseer.
- Su, Z., Xu, H., Zhang, D., & Xu, Y. (2014). Chinese sentiment classification using a neural network tool word2vec. In *Multisensor fusion and information integration for intelligent systems (MFI), 2014 international conference on* (pp. 1–6). doi:10.1109/MFI.2014.6997687.
- Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37, 267–307.
- Tang, D., Wei, F., Qin, B., Liu, T., & Zhou, M. (2014). Coooolll: A deep learning system for twitter sentiment classification. In *Proceedings of the 8th international workshop on semantic evaluation (semeval 2014)* (pp. 208–212).
- Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., & Qin, B. (2014). Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL* (1) (pp. 1555–1565).
- Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 417–424). Association for Computational Linguistics.
- Vo, D.-T., & Zhang, Y. (2015). Target-dependent twitter sentiment classification with rich automatic features. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI 2015)* (pp. 1347–1353).
- Wang, G., Sun, J., Ma, J., Xu, K., & Gu, J. (2014). Sentiment classification: The contribution of ensemble learning. *Decision support systems*, 57, 77–93.
- Wang, S., & Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2* (pp. 90–94). Association for Computational Linguistics.
- Whitehead, M., & Yaeger, L. (2010). Sentiment mining using ensemble classification models. In *Innovations and advances in computer sciences and engineering* (pp. 509–514). Springer.
- Wilson, T., Wiebe, J., & Hoffmann, P. (2009). Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35, 399–433.
- Xia, R., & Zong, C. (2010). Exploring the use of word relation features for sentiment classification. In *Proceedings of the 23rd international conference on computational linguistics: Posters COLING '10* (pp. 1336–1344). Stroudsburg, PA, USA: Association for Computational Linguistics. <http://dl.acm.org/citation.cfm?id=1944566.1944719>.
- Xia, R., & Zong, C. (2011). A pos-based ensemble model for cross-domain sentiment classification. In *IJCNLP* (pp. 614–622). Citeseer.
- Xia, R., Zong, C., & Li, S. (2011). Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181, 1138–1152. doi:10.1016/j.ins.2010.11.023. <http://www.sciencedirect.com/science/article/pii/S0020025510005682>.
- Zhang, D., Xu, H., Su, Z., & Xu, Y. (2015). Chinese comments sentiment classification based on word2vec and svm perf. *Expert Systems with Applications*, 42, 1857–1863.
- Zhang, K., Cheng, Y., Xie, Y., Honbo, D., Agrawal, A., Palsetia, D., et al. (2011). Ses: Sentiment elicitation system for social media data. In *Data mining workshops (ICDMW), 2011 IEEE 11th international conference on* (pp. 129–136). IEEE.
- Zhang, P., & He, Z. (2015). Using data-driven feature enrichment of text representation and ensemble technique for sentence-level polarity classification. *Journal of Information Science*, 41, 531–549.

General Discussion, Conclusions and Future Research

In the end, research is oriented to obtain rigorous conclusions on which we are able to keep searching. Defining the right questions, and giving an answer to them is part of the process. This chapter discusses the whole thesis, raising conclusions from the obtained results, as well as outlining some future work lines.

4.1 Overview

Previously, Chapter 1 has introduced the main areas to which this thesis has made contributions, as well as its objectives. In this regard, Figure 4.1 illustrates how the work developed in this thesis distributed along with the different objectives, showing our different contributions.

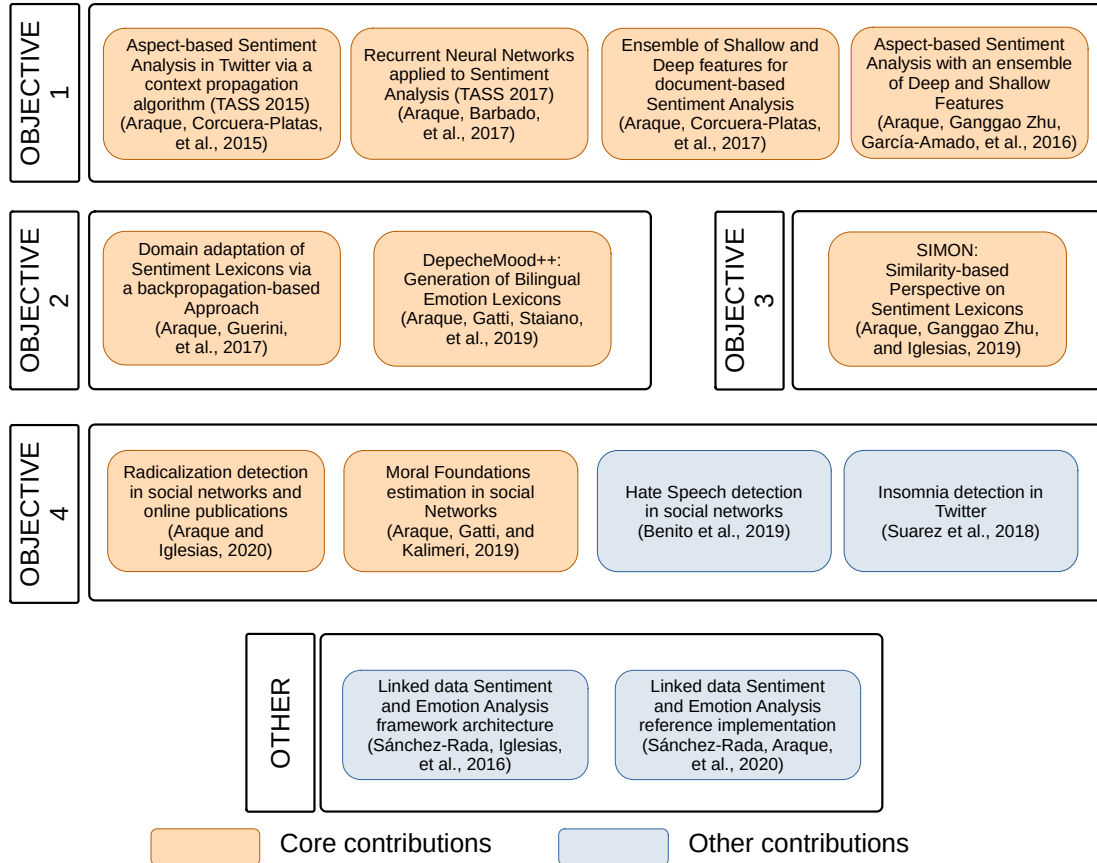


Figure 4.1: Summary of the contributions of this thesis, organized by objectives.

In a general perspective, the main contributions can be categorized according to the four objectives we have declared (Sect. 1.4): (1) definition of a framework that combines surface and deep features for Sentiment Analysis; (2) applying statistical information to generate sentiment and emotion lexicons; (3) application of distributed representations and lexicons for Sentiment Analysis; (4) transference of the output from the previous objective to other domains. One of the most relevant contributions of this thesis is SIMilarity-based sentiment projectiON (SIMON). This feature extraction model that exploits both a word embedding model and a domain-oriented lexicon to compute text representations that can be used to train posterior machine learning models. While the SIMON model has been

initially proposed in the context of Sentiment Analysis, it has been later adapted to be used in a variety of different domains.

As outlined in Section 1.1, several contributions fall into the Sentiment and Emotion analysis fields. In the context of aspect-based Sentiment Analysis, we have implemented an aspect context propagation algorithm that extracts the aspect-related words. This work is presented in **“Aspect Based Sentiment Analysis of Spanish Tweets”** (Section A.2.9). While the mentioned work improves how the aspects are extracted from text, we have also made efforts toward improving both the classification and the sentiment that can be associated with said aspects. That last work is described in **“Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis”** (Section A.2.7).

To improve and deepen into document-based Sentiment Analysis, we firstly have done a thorough study of the state-of-the-art approaches that tackle deep learning. Also, we have proposed a taxonomy to classify such models. Following such a classification, we have designed several ensemble models that combine surface, deep, and lexicon features. The publication **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5) presents this work. Continuing with the work on deep learning applied to Sentiment Analysis, broad experimentation over the RNN architecture is done in **“Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets”** (Section A.2.6).

As the center of this thesis, another contribution is made in the context of document-level Sentiment Analysis, where the SIMilarity-based sentiment projectiON (SIMON) model is proposed. As mentioned, the SIMON model offers a similarity-based perspective on domain lexicons. Indeed, it allows exploiting both a word embedding model and a domain lexicon, extracting useful features that can be later used to train a learning algorithm. SIMON is described in the publication **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3).

Regarding sentiment and emotion lexicons, generating such important resources is an open research problem that can significantly affect the performance of posterior Sentiment and Emotion Analysis systems. This thesis has contributed to this field by proposing a generic model that uses the neural network strategy of backpropagation to generate the annotations of a sentiment lexicon. Such work is presented in **“Neural domain adaptation of sentiment lexicons”** (Section A.2.5). Additionally, this thesis deepens into the study of compositional semantics to generate emotion lexicons for several languages is done in **“DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet**

Powerful Techniques” (Section 3.2.4). In this last work, a thorough analysis of the lexicon generation is performed, including the effect of word embeddings to enhance the lexicon’s performance.

As described, the developments made in the field of Sentiment Analysis have partly motivated the redirection of efforts toward other NLP-related domains. In this sense, a contribution of this thesis revolves around the detection of terrorist radicalization in texts, which is described in **“An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity”** (Section 3.2.1). This work adapts the SIMON model to the radicalization analysis domain and deepens the effect of Emotion Analysis in this task. This work constitutes an example of how a contribution in Sentiment and Emotion Analysis drives the advancement in another domain.

On a similar note, this thesis contributes to the computational application of the Moral Foundations Theory. Applying and designing computational approaches to extract moral values from texts is a novel field, and can largely benefit from NLP advancements. In this regard, a moral lexicon has been created by means of crowdsourcing, and in order to exploit it effectively, the SIMON model is used and compared against ad-hoc solutions. This work is presented in **“MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”** (Section 3.2.2).

4.2 Scientific results

In this section, we discuss the scientific results of this thesis, as well as their relation to the objectives presented in Section 1.4. We include a list of the publications that have contributed to each objective. For the full details on each publication, both the abstract and the full text can be found in the referenced pages.

4.2.1 Objective 1: Surface and Deep methods framework

Table 4.1: Publications related to Objective 1

Page	Title	Impact	Contribution
95	Enhancing deep learning sentiment analysis with ensemble techniques in social applications	JCR 2018 Q1 (4.292)	Definition of a taxonomy of surface and deep features with ensemble techniques in Sentiment Analysis. Proposal of models that combine different types of features and methods. Thorough evaluation of methods, obtaining positive results in relation to the effectiveness of said methods.

230	Aspect Based Sentiment Analysis of Spanish Tweets	Study of surface features for aspect-based Sentiment Analysis using a context extraction algorithm. Participation in the international competition TASS 2015.
212	Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis	Adaptation of ensemble models of both methods and features to the domain of aspect-based Sentiment Analysis. Proposal of a novel approach for classifying aspects, used in conjunction to the Sentiment Analysis system.
205	Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets	Study of the effectiveness of RNN networks to Sentiment Analysis. Participation in the international competition TASS 2017.

As the core result related to the first objective, we addressed an in-depth analysis of both surface and deep features. As Section 1.1 introduces, the combination of these two types of features is still an open challenge in Sentiment Analysis, in terms of both its effects on the system performance and the combination strategy. Also, in the last years, advancements in distributional representations have motivated a large body of researchers to explore such popular models as word embeddings, inserting these techniques into all fields of NLP. In this context, a result of this thesis is the proposal of a taxonomy that classifies both feature extraction and ensemble approaches. This taxonomy organizes the literature into two dimensions. In one dimension, the different possibilities for feature extraction are outlined: surface features, generic word vectors, and affect word vectors. On the other dimension, the taxonomy describes how the ensemble could be done in terms of combining classifiers, or at the feature level. In some works, these possibilities are denoted late and early fusion, respectively (Poria, Cambria, et al., 2017).

The definition of a taxonomy constitutes a result in itself since it can be used to systematically organize scientific works, distributing them along dimensions that consider common characteristics. Nevertheless, the Sentiment Analysis community, and more broadly, the NLP field, are experimental. Thus, as part of these efforts, we define an evaluation methodology that can be used to evaluate Sentiment Analysis models according to the proposed taxonomy quantitatively. In this sense, we have determined the practices outlined in Chapter 2: general considerations regarding machine learning experimentation (Section 2.2); the metrics used for the computation of the results, highlighting the F1-Score (Section 2.3); and the use of the Friedman statistical test (Section 2.4) to obtain a statistical framework with which compare different models.

The models designed are oriented to combine different types of features and models, thus exploiting the resulting ensemble. In this sense, this work defines a straightforward nomenclature to models, depending on the ensemble strategy and features used. Base types

of models are M (no ensemble whatsoever), and CEM , which are models that implement ensemble of classifiers. In relation to the type of features used, we define three categories: surface (S), generic word vectors (G), and affect word vectors (A). Features are expressed in the subscript. Finally, the superscript in CEM models expresses whether the ensemble has been done by a voting strategy (Vo) or a meta-learning approach (MeL). We define the following models: M_G , M_{SG} , $M_{SG} + \text{bigrams}$, M_{GA} , CEM_{SG}^{Vo} , CEM_{SG}^{MeL} , $M_{SGA} + \text{bigrams}$, CEM_{SGA}^{Vo} , and CEM_{SGA}^{MeL} .

On an important note, a fundamental step lies in the selection of the evaluation data. The criteria used to select and manage the used datasets for the evaluation conditions the outcomes of the evaluation process. So, for this work, a broad selection of datasets has been made, representing two of the most frequently used domains in Sentiment Analysis evaluation: movie reviews and Twitter. In total, we have selected seven distinct datasets, each with different properties in terms of annotation, label distribution, and textual content.

After performing the evaluation, the results indicate that using an ensemble with a wide variety of information sources tends to improve the overall performance of a learning system. In particular, the final performance of a meta-learning approach (which implements a learning algorithm at the ensemble phase) improves over a fixed ensemble scheme. In light of the results, we can also see that the model M_{SG} achieves the best performances in the Twitter domain, but does not in the movie review texts. This result indicates that combining word vectors through convolutional functions in long texts does not lead to high sentiment classification performances. In comparison, attending to the difference in the performance of the $M_{SGA} + \text{bigrams}$, CEM_{SGA}^{Vo} , and CEM_{SGA}^{MeL} models, we draw the conclusion that dividing the features into smaller sets, as it is done in the ensemble models, benefits the classification performance.

The work that includes the proposal of the taxonomy, as well as the evaluation methodology, the selection of datasets, and the implementation of the evaluation process, is presented in the journal publication **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5).

Given the positive results obtained with the evaluation of surface and deep features in an ensemble configuration, we worked on adapting said models to the domain of aspect-based Sentiment Analysis. This poses an interesting challenge since the conditions for the learning models largely vary between types of Sentiment Analysis, frequently causing the system’s performance to drop drastically. Thus, the feature combination and ensemble models have been used in this new setting, incorporating two new elements to the system. Firstly, the mentioned context extraction algorithm is inserted into this system. As described, the

context of an aspect is extracted by this algorithm, providing the sentiment module with the text to analyze. A novel aspect classification module is developed as a second addition: it exploits the representation capabilities of both word embeddings and WordNet (G. Zhu et al., 2017; Miller, 1995). These representations are then used by an Support Vector Machines (SVM) that is trained to predict the category of the aspect. This system was trained and evaluated with data we extracted from the Yelp platform ¹, on the restaurant reviews domain. As development and testing data (Section 2.2), a dataset from the international competition SemEval 2016 (Pontiki et al., 2016), that also contains restaurant reviews. This work is presented as a conference publication in **“Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis”** (Section A.2.7).

International competitions are frequent in the field of Sentiment Analysis. In this line of work, this thesis contributed with the implementation of a context extraction algorithm (Mukherjee et al., 2012). This algorithm is able to extract the set of words that relates to a target aspect by means of exploiting the dependency parsing tree. Through data-driven validation, the set of dependency relations is pruned, retaining the most informative relations. Later, this context can be inserted into a machine learning model that is trained to predict the sentiment that corresponds to each aspect and its context. Regarding the learning approach taken in this work, a large number of surface features are used, with the aim of improving the overall performance of the system: n -grams, capitalization markers, PoS, number of hashtags, punctuation marks, number of elongated words, emoticon use, scores from a sentiment lexicon, intensifier words, and negation handling. The whole model was trained to predict sentiment from Spanish tweets. With this system, we participated in the international challenge of TASS 2015 (Villena-Román et al., 2015).

As a result, the presented system qualified for the 2nd position in the category of aspect-based Sentiment Analysis. This work was presented as conference proceedings in the publication **“Aspect Based Sentiment Analysis of Spanish Tweets”** (Section A.2.9). Concerning Objective 1, this work was an initial study on surface feature extraction for Sentiment Analysis, as we explore a large number of different feature types, evaluating their effectiveness in a learning model, including the inclusion of information from a sentiment lexicon. Still, this work does not tackle deep features.

Finally, in relation to Objective 1, an additional contribution is made in the context of the international challenge TASS 2017. As introduced in Section 1.2, the fast development of deep neural architectures has boosted the research on their application to NLP, as well

¹<https://www.yelp.com>

as the performance on learning systems. Consequently, within the scope of experimenting with ‘deep learning features’, we have designed a RNN-based end-to-end system trained to predict the sentiment in the context of Twitter. Following previous experimental procedures, in this work, a combination with the information from a sentiment lexicon is done. That is, the lexicon’s values are appended to word representations the network is learning during training. These representations are fed to the RNN network, which computes a fixed vector, that aims to represent the whole text. Finally, this final representation is inserted into a Multi-Layer Perceptron (MLP) network, that outputs the sentiment prediction. This work is presented as a conference paper in the TASS 2017 proceedings “**Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets**” (Section A.2.6).

4.2.2 Objective 2: Definition of a lexicon generation methodology

Table 4.2: Publications related to Objective 2

Page	Title	Impact	Contribution
198	Neural domain adaptation of sentiment lexicons		Proposal of a novel lexicon generation method that exploits the backpropagation algorithm to compute sentiment annotations. Evaluation with a comparison to previous methods.
82	DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques	JCR 2018 Q1 (6.288)	An improvement over a previous technique for the generation of English emotion lexicon, introduction to the Italian language. Expansion of said lexicons via word embeddings, to both a more extensive vocabulary and to the French language. Evaluation with thorough experimentation, including numerous previous words and state-of-the-art techniques.

In previous sections, we have developed on the importance of lexicons for both Sentiment and Emotion Analysis, highlighting that these represent a subjective source of information that an algorithm can not acquire automatically. Of course, as mentioned, the domain where the analysis is done is of the most considerable significance. One has to take into account that in different domains, the sentiment polarity for a word can modify its intensity, and even shift its sign (Peter D Turney, 2002). That is why a system oriented to perform an analysis on one specific domain should use a lexicon that is adapted to such domain in order to boost the system’s performance.

In addition, advancements in the application and development of novel neural architectures in NLP is remarkable. During the last years, a large number of works have appeared that use of develops neural networks for Sentiment Analysis (Young et al., 2018). While

many previous works tackle the issue of domain adaptation in Sentiment Analysis, a neural network has not been used before to generate a sentiment lexicon. Under this perspective, and having obtained the results above (Section 4.2.1), we have contributed with a methodology that can generate a sentiment lexicon from annotated texts.

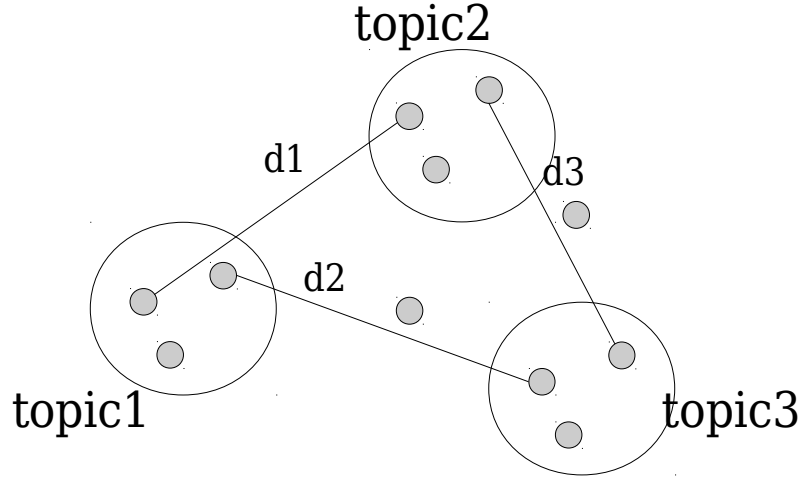


Figure 4.2: Graphical schema of how domain centrality is computed.

Our first step in automatically generating a domain-adapted lexicon begins by characterizing the concept of *domain*. Understanding how we can model what a domain is in the context of Sentiment Analysis results on benefit for domain adaptation applications. In this context, accurately characterizing a domain can substantiate the lexicon adaptation process. As previous knowledge, we would like to know some characteristics of the data before initiating any process, so we can have a measure of how well the method is going to perform. This thesis tackles this problem by proposing a data-driven metric that aims to characterize a set of documents D . The proposed metric aims to estimate the semantic centrality of D . That is, how specific to a certain set of topics T the dataset is. As an example, we can consider a dataset that describes some products of the electronics domain. If most of the text in that dataset refers to certain recurrent topics (phones, batteries, screens, computers), it is possibly a centered domain. On the contrary, a dataset that includes a large set of topics would have a lower centrality.

Following this, we propose a metric m that measures the centrality of D using a semantic structure: a word embedding model. As described in Section 1.2, word embedding models retain a semantic structure that can be exploited in a variety of applications (Mikolov et al., 2013). The proposed method aims to leverage the information embedded in this representation for domain characterization. The idea is to measure the distances between

a set of selected words W from D . These measures retain information on how dispersed W is on the vector space, and thus the extent of the centrality of D . Figure 4.2 shows a graphical representation of the measures in relation to a set of topics. The presented metric is defined as the average distance of the set of words W to each other. As a note, the set W is formed by selecting a number of most common words from the dataset and filtered with the stopwords set from Porter et al. (Porter et al., 1980).

After characterizing what a domain is, and how to compute the centrality of a domain, we then propose the domain lexicon generation or adaptation. The method proposed in this thesis can either compute a domain-adapted lexicon from a previous lexicon (adapting the sentiment values of the words) or generate a new lexicon from scratch. In this way, we consider the situation where we have access to distantly labeled textual data. We refer to distantly labeled data as the one that is not manually nor automatically annotated but has some meta-data associated that can be used as sentiment label. In this case, we consider product reviews that have a meta-data value belonging to a 5-score system. From this score value, we extract a sentiment signal with which we train the proposed model.

The proposed approach consists of a feedforward neural network (also called Multi-Layer Perceptron). Such a network is trained from representations of a set of documents that have associated distant labels. That is, if we consider a set of k words, the values from a particular sentiment lexicon are extracted and used as a representation for the network. Regarding the initialization of the sentiment lexicon values, as mentioned, the network uses an existing generic lexicon from which starts modifying or initiates the lexicon values randomly. The network is trained by using distant labels with a cross-entropy loss function on the task of classifying the sentiment using the lexicon values. The key idea is that, at training time, the backpropagation algorithm can be used to let the sentiment error signal flow to the lexicon values. In this way, by training the network to differentiate the sentiment of the documents, it is also modifying the lexicon values, which are a continuous range of numeric values.

When training is finished, the modified values are extracted, forming a new domain-adapted sentiment lexicon. Nevertheless, the resulting lexicon does not normally have the expected distribution of a human-readable lexicon. That is, we would expect a sentiment lexicon to be bounded in a range such as $[-1, 1]$; and that positive number should represent positive orientations, and the contrary for negative sentiment orientations. The modification of the lexicon values relies on the backpropagation algorithm, and this method has no inherent constraints on the distribution of a certain set of parameters. In order to transform the modified lexicon to one that exhibits a “human-readable” distribution, we define a transformation operation. More formally, let a be the vector that contains the lexicon values so

that the value a_i represents the lexicon value for the word i of the vocabulary. We define the transformation operation as $g((\mathbf{w} * \mathbf{a}) + b)$, $*$ being the element-wise product and g the element-wise function $g(x) = (x - a) \frac{d-c}{b-a} + c$ that transforms the values from one range $[a, b]$ to another $[c, d]$. Figure 4.3 illustrates how the transformation process operates.

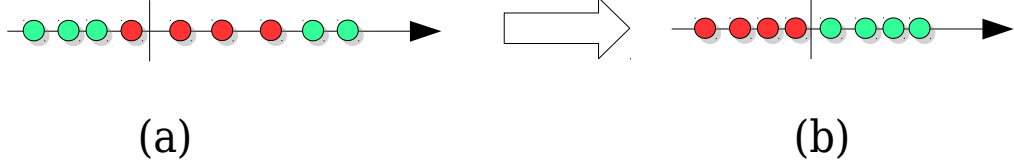


Figure 4.3: Graphical representation of polarity distribution of the modified lexicon (a) before the transformation process, and (b) after the transformation.

We call the presented domain lexicon generation model SEntiment adapted to the Domain Lexicon (SEDLex). This method has been trained on a dataset extracted from the Amazon Product Dataset (McAuley et al., 2015; He et al., 2016), which contain millions of product reviews extracted from Amazon. The test dataset is the Multi Domain Sentiment Dataset v2 (Blitzer et al., 2007) that contains a set of product reviews already annotated. In the evaluation, we compare the presented method against other lexicon generation approaches (Staiano et al., 2014), as well as a known sentiment lexicon (Gatti et al., 2016). In conclusion, the experiments have shown that this model effectively adapts a sentiment lexicon to a specific domain, outperforming the performance of a linear classifier trained with only the lexicon as features with respect to a generic sentiment lexicon. Furthermore, the generated lexicons show a strong correlation between their values and the sentiment labels of the test set. This allows us to use the generated lexicons for simple sentiment classification without the need for any learning process.

Also, interestingly enough, additional evaluation results indicate that there is a relation between the value of the proposed centrality metric and the performance improvement of the generated sentiment lexicons. This last observation suggests that using the centrality measure, before performing any costly training process, can serve as an indication of the final performance of the SEDLex method. This work, including the centrality measure and the lexicon generation method, has been published in the conference paper “**Neural domain adaptation of sentiment lexicons**” (Section A.2.5).

Continuing with the work on lexicon generation, this thesis has continued on the work of automatically generating lexical resources that can be used subsequently in a Sentiment or Emotion Analysis system. In fact, as a contribution that is motivated by the positive results in the generation of a sentiment lexicon, we have made efforts in the line of generating

an emotion lexicon. This work has continued that described in Staiano et al., 2014. In this thesis, we present an extension of an existing and widely used emotion lexicon for English, as well as a novel version of the lexicon targeting Italian. For this, the technique of compositional semantics is used.

Consistently with (Staiano et al., 2014), the lexica creation is based on a distributional assumption for which the emotions evoked by a word are the emotions that are associated with the documents in which such words appear more often. This also means that we can ‘project’ the emotions of a document to its words. This technique consists on the following steps: (i) a document-by-emotion matrix (M_{DE}) is produced, containing the vectors that match each document and its emotion annotation; (ii) a word-by-document matrix (M_{WD}) is computed, which contains the number of occurrences of words in a given document, divided by the total number of words in that document, as recommended in (Staiano et al., 2014); (iii) matrix multiplication is performed between the document-by-emotion and the document-by-word matrices ($M_{DE} \cdot M_{WD}$) to obtain a raw word-by-emotion matrix (M_{WE}). This method allows us to ‘merge’ words with emotions by summing the products of the weight of a word with the weight of the emotions in each document. Finally, a transformation is done over M_{WE} by first applying normalization column-wise (to eliminate the over-representation for happiness as discussed previously) and then scaling the data row-wise, summing up to one. For training this method, we have collected a wide collection of online news from the sources `rappler.com` and `corriere.it`. These matrices that we call DepecheMood++ represent our emotion lexica for English and Italian, and are freely available for research purposes.

Furthermore, we have performed an extensive evaluation of the generated resources. We show how simple techniques can be used, both in supervised and unsupervised experimental settings, to boost performance on datasets and tasks of varying degrees of domain-specificity. For this, we have used seven different datasets, whose domains are news, blog posts, and twitter messages. In terms of the different tasks, we evaluate DepecheMood++ in the task of unsupervised regression, supervised regression, supervised classification, comparison to other lexica, and cross-validation with comparisons to the state-of-the-art. In these experiments, we report an extensive comparative analysis against other available emotion lexica and state-of-the-art supervised approaches, showing that DepecheMood++ emerges as the best-performing non-domain-specific lexicon in unsupervised settings. We also observe that simple learning models on top of DM++ can provide more challenging baselines.

Additionally, we finally introduce embedding-based methodologies to perform a) vocabulary expansion to address data scarcity and b) vocabulary porting to new languages in case

training data is not available.

In order to expand lexica vocabulary using a word embedding model, we propose a technique we call *embedding expansion*: the idea is to map words that do not initially appear in a specific lexicon to a word that is contained in the lexicon mentioned above. Hence, given an out-of-vocabulary (OOV) word, we search in the embedding space for the closest word that is included in the lexicon so that the distance to the OOV word is minimal, in terms of the cosine distance in the embedding model. Finally, the emotion scores from the lexicon word are assigned to the OOV word.

An important limitation of the proposed lexicon generation method lies in the limited availability of emotion annotated corpora. In light of this, and to delve into cross-lingual settings with the generated lexicon, we have performed a projection of the DM++ resource to the French language. In order to derive a French version from the original English lexicon, using the DM++ token-based version and the ground-truth bilingual dictionaries released within the Facebook MUSE library (Conneau et al., 2017), we performed the following steps: (i) eliminate cross-lingual homographs (optional), i.e., tokens that are present in both source and target languages, but may mean different things; (ii) aggregate synonyms by averaging the emotional values of the source tokens they map. For instance the French *injustifié* maps to the English *unwarranted*, *unjustified*, *wrongful*.

The described work on generating a bilingual emotion lexicon, its evaluation, and expansion methodologies is presented in the paper “**DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques**” (Section 3.2.4).

4.2.3 Objective 3: Definition of a similarity-based model over distributed representations

Table 4.3: Publications related to Objective 3

Page	Title	Impact	Contribution
67	A semantic similarity-based perspective of affect lexicons for sentiment analysis	JCR 2018 Q1 (5.101)	Proposal, implementation, and experimental evaluation of the SIMilarity-based sentiment projectiON (SIMON) model. Study of different characteristics of the method, such as the selection of lexicon words.

As presented, lexical resources are widely popular in the field of Sentiment Analysis, as they represent a resource that directly encodes subjective sentimental knowledge. Usually, sentiment lexicons are used for polarity estimation through the matching of words contained in a text and their associated lexicon sentiment polarities. Nevertheless, such resources have

limitations in vocabulary coverage and domain adaptation. Besides, many recent techniques exploit the concept of distributed semantics, normally through word embeddings. In this thesis, we computed a semantic similarity metric between text words and lexica vocabulary. With the use of this metric, this paper proposes a sentiment classification model that uses the semantic similarity measure in combination with embedding representations. We call such a model SIMilarity-based sentiment projectiON (SIMON). This work is described in the journal publication “**A semantic similarity-based perspective of affect lexicons for sentiment analysis**” (Section 3.2.3).

As stated, in general, word embeddings contain semantic and syntactic information. Arguably, it is accepted that pre-trained word vectors do not enclose specific sentiment information, as no sentiment-related signal has been included in the training process. In order to include subjective sentiment information into the proposed model, additional information sources must be included in the feature extraction process. In this way, semantic similarity features exploit the mentioned word embeddings regularities using a selection of sentiment words, namely, a sentiment lexicon vocabulary. This work proposes the representation of a particular word, that may be outside of the lexicon vocabulary, by a projection to a set of sentiment words extracted from a sentiment lexicon. Such projection is computed using the semantic similarity between words, which can be computed by means of a word embedding model or a word taxonomy. In this way, a certain word is represented by its similarities to the selection of lexicon words.

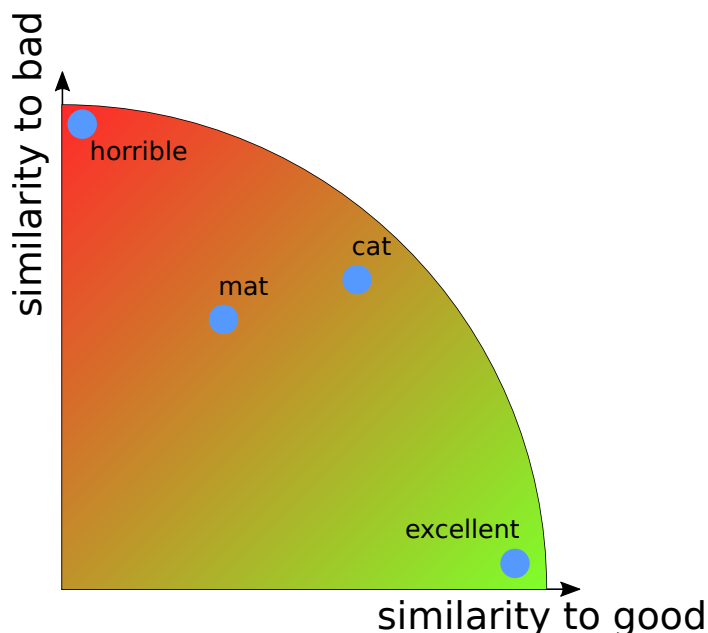


Figure 4.4: Conceptual diagram of word projection over a lexicon formed only by the set of words $\{good, bad\}$.

To illustrate the concept of projection, Figure 4.4 shows a conceptual case where only two words (*good* and *bad*) are selected from a sentiment lexicon; it can be seen that several words are projected in then in a two-dimensional space. As described, the values of that space correspond to the semantic similarity of each word (e.g., *horrible*, *cat*, *mat*, *excellent*) to the lexicon words.

The main idea of the SIMON method is that given a domain lexicon, the input text is measured against it, computing a vector that encodes the similarity between the input text and the lexicon. Such a vector encodes the similarity, as given by the word embedding model, of each of the words of the analyzed text to the lexicon words. Thus, a matrix that contains the similarities between the text and lexicon word is composed. Such a matrix is then reduced to a vector using a *pooling function* (e.g., the maximum). An illustration of how the similarity scores are computed and passed through the pooling function is shown in Figure 4.5. It can be seen that the pooling function transforms the dimension of the matrix (that is dependent on the number of words of the input text) to a fixed-dimension vector, defined by the number of selected lexicon words.

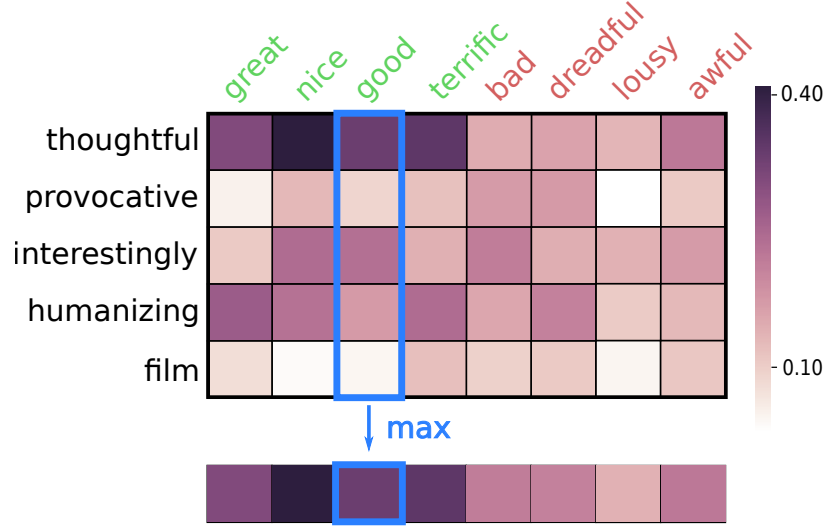


Figure 4.5: SIMON similarity computation: similarities are computed against a selection of lexicon words (in green and red), and a *max* function is applied column-wise, obtaining a feature vector.

Since this model represents a key contribution of this thesis, we explain it in detail. That is, more formally, let

$$W^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \dots, w_i^{(i)}, \dots, w_I^{(i)}\} \quad (4.1)$$

be the set of length I , formed by input tokens that constitute the text to be analyzed. This text can be a sentence, a paragraph or a whole document. Also, we consider that a lexicon

is formed by tuples of sentiment word and polarity value: $(w_j^{(s)}, s_j)$. In this way, we define the selection of target sentiment words,

$$W^{(s)} = \{w_1^{(s)}, w_2^{(s)}, \dots, w_i^{(s)}, \dots, w_L^{(s)}\} \quad (4.2)$$

which is extracted from a sentiment lexicon. Similarly, the vector $\mathbf{l} = [l_1, l_2, \dots, l_L]$ comprises the numerical sentiment values of the word in $W^{(s)}$, as given by a sentiment lexicon. The selection process that generates the set of words $W^{(s)}$ is done in two steps, following different criteria: (i) frequency of appearance in the training data, and (ii) informativeness of each word towards the training annotation.

The process to generate the features is as follows. For each word $w_i^{(i)} \in W^{(i)}$ and for each sentiment word $w_j^{(s)} \in W^{(s)}$, a similarity value is computed so that

$$S_{i,j} = \text{sim}(w_i^{(i)}, w_j^{(s)}), \quad (4.3)$$

with $S_{i,j} \in [0, 1]$. This value represents that $w_i^{(i)}$ and $w_j^{(s)}$ are no similar at all if the result is 0, and completely similar if the result is exactly 1. After iterating over all the input words $W^{(i)}$ and all sentiment words $W^{(s)}$, a matrix $S \in \mathbb{R}^{I \times L}$ can be constructed, containing all similarity values.

Following, a pooling function (maximum) is applied column-wise over S , obtaining the semantic similarity feature vector $\mathbf{p}$ of L -length:

$$p_j = \max S_{:,j} = \max \text{sim}(w_k^{(i)}, w_j^{(s)}) \quad \text{for } k \in \{1, 2, \dots, I\} \quad (4.4)$$

As seen in the results, the best results are obtained when using the embedding-based similarity measure. In this sense, a previously trained word embedding model is used. Nevertheless, similarity measures can be computed using any word embedding model and are not dependent on the embedding dimension either. The embedding similarity is implemented using the dot product between an input word w_i and a sentiment word l_j :

$$\text{sim}(w_i^{(i)}, w_j^{(s)}) = E_{w_i^{(i)}}^T E_{w_j^{(s)}} \quad (4.5)$$

where $E \in \mathbb{R}^{|V| \times d}$ is the embedding matrix, and $E_{w_i^{(i)}} \in \mathbb{R}^d$ and $E_{w_j^{(s)}} \in \mathbb{R}^d$ are the word vectors associated with words w_i and l_j respectively.

In such a way, the model can exploit the knowledge in word embeddings and the domain information that the lexicon offers. As an additional remark, this method does not need large corpora to be trained and thus can be used in problems where annotated data is scarce.

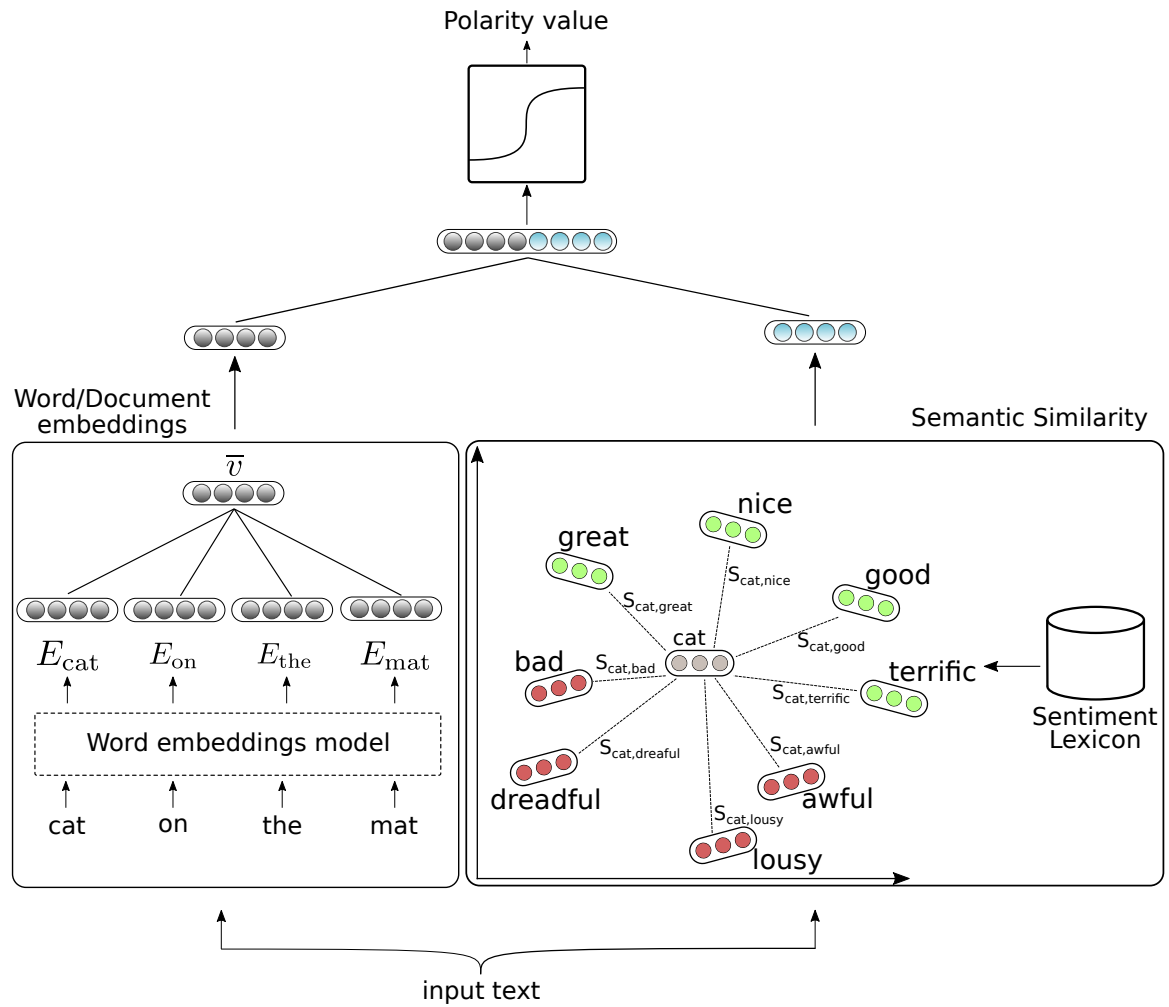


Figure 4.6: SIMON system architecture diagram, when combined with embedding text representations.

The proposed model uses an embedding-based textual representation that transforms the input text into a fixed-length feature vector. As previously studied **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5), the number of words in a text directly affects the effectiveness of embedding-based representations. Accordingly, this work makes use of two variations of the same feature extraction method. The first one is aimed at short texts, as the ones found in the Twitter platform. In this variation, word vectors are extracted for each word in the input text. Following our previous study, the average pooling operation is performed on all word vectors, resulting in a vector of the same dimension as the original word vectors that are then fed to a logistic regressor. As for the second variation, the representation of a text uses Paragraph Vector (Le et al., 2014), and it is used in long texts. This distinction between short texts (typical of online sources) and long texts (more commonly found in review sites) has shown an improvement in the performance of the presented Sentiment Analysis models. Figure 4.6 shows how SIMON can be used in combination with these embedding-based text representations.

In order to assess the effectiveness of this model, we have performed an extensive evaluation. For this, we have used eight different Sentiment Analysis datasets, and four sentiment lexicons from the state-of-the-art, in order to properly assess the effectiveness of the SIMON model. As part of this evaluation, we have shown that not all words in the lexicon have the same importance when estimating sentiment polarity. In fact, the SIMON model contemplates a mechanism to select lexicon words in order to exploit this characteristic. Also, experimental results show that the proposed feature extraction method yields fairly good performances when used with a simple classifier. We consider this result a promising one since more complex learning architectures can probably boost the overall performance. At the same time, when compared against the keyword matching approach (Section 1.1), the experiments indicate that the use of semantic distance method can extract subjective sentiment information from a lexicon better.

Following, in this work, we raised the concern of how the lexicon characteristics affect the proposed feature extraction process. This question is oriented to predict the performance of the model when confronted with a new dataset. For this, several similarity metrics between vocabularies have been defined, determining the correlation between these metrics and the performance difference in cross-dataset and cross-lexicon experiments. In this regard, the experiments indicate that the lexicon words largely determines the resulting Sentiment Analysis performance. That is, similar lexicon word selections yield similar Sentiment Analysis performances.

4.2.4 Objective 4: Transference of a similarity-based model that exploits lexicons to other domains

Table 4.4: Publications related to Objective 4

Page	Title	Impact	Contribution
38	An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity	JCR 2018 Q1 (4.098)	Adaptation of the SIMON method and proposal of a new emotion-based feature extraction method applied to radicalization detection. Introduction of a novel dataset for the domain.
55	MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction	JCR 2018 Q1 (5.101)	Introduction of a novel lexicon for the domain. Adaptation of the SIMON method and proposal of two new feature extraction methods applied to moral value estimation.

Section 1.1 has motivated how, in light of the positive results obtained with the fulfillment of Objective 3, this thesis expanded its work in order to adapt the developed methods to newer Natural Language Processing domains and tasks. Many fields can largely benefit from the introduction of NLP. In this sense, we aim to apply, adapt, and develop the models that previous objectives have outputted, highlighting the case of the SIMON model. In this thesis, the work revolves around two domains: radicalization detection and moral value estimation.

Attending to the contributions to radicalization detection, we propose a machine learning model that exploits two kinds of information sources: (i) emotion and (ii) embedding-based semantic similarity features (through the SIMON model). Figure 4.7 shows the model architecture. The first approach has been designed to take advantage of an existing emotion dictionary in order to generate features that can be used for radicalization detection. Such an approach generates simple but useful features by computing a statistical summary of the emotion valence of the words in the text. It uses the emotion annotations of an emotion lexicon, aggregates them, and computes several pooling functions in order to summarize the information.

In this manner, this work proposes the use of an emotion lexicon in order to extract emotion-driven features that are, as explained, fed to a machine learning algorithm. With this, we intend to investigate whether this kind of information is relevant for the task of radicalization detection and to which extent. So, an emotion lexicon-based representation is proposed that makes use of statistical measures to encode the emotion of the text. For simplicity, we denote this method as *EmoFeat* (Emotion Features).

In the context of *EmoFeat*, consider an emotion lexicon composed by a vocabulary $W^{(l)} = \{w_1^{(l)}, \dots, w_i^{(l)}, \dots, w_P^{(l)}\}$ and a vector of numeric annotations $\mathbf{L} = [\mathbf{l}_1, \dots, \mathbf{l}_i, \dots, \mathbf{l}_P]$. Note that such lexicon has an annotation $\mathbf{l}_i$ for each word w_i , with a total of P $(w_i, \mathbf{l}_i)$ pairs. Note also that $\mathbf{l}_i$ is a vector expressing the intensity of each emotion for word $w_i^{(l)}$ in the lexicon. An emotion vector has dimensionality $\mathbf{l}_i \in \mathbb{R}^m$, and thus the emotion lexicon annotation matrix is $\mathbf{L} \in \mathbb{R}^{P \times m}$, where m is the number of emotions considered in the lexicon. Following, let $W^{(i)} = \{w_1^{(i)}, \dots, w_j^{(i)}, \dots, w_I^{(i)}\}$ be the set of length I composed by the input words.

For each word w_k of the intersection $W^{(l)} \cap W^{(i)}$, the associated emotion vector is extracted from $\mathbf{L}$. This process outputs a matrix that contains the emotion annotation for all the input words that appear in the lexicon. Next, several statistical measures are taken in order to represent such a matrix as a feature vector. The proposed measures are average, maximum, and median. Please note that these measures can be used independently: for example, only average and median could be used, removing maximum. As a result, a feature vector is obtained with dimension $n \cdot m$, where n is the number of statistical measures selected.

Algorithm 1 EmoFeat

Require: Emotion lexicon composed by a vocabulary $W^{(l)}$ and annotations $\mathbf{L}$

Ensure: $\mathbf{v} \in \mathbb{R}^{n \cdot m}$, the final feature vector

```

1: for all  $w_k \in W^{(l)} \cap W^{(i)}$  do
2:    $\mathbf{E}_{k,:} \leftarrow \text{emotionAnnotation}(w_k, \mathbf{L})$ 
3: end for
4: for  $i \leftarrow 1, n$  do
5:   for  $j \in 1, m$  do
6:      $index \leftarrow i + n(j - 1)$ 
7:      $\mathbf{v}_{index} \leftarrow \text{statMeasure}(i, \mathbf{E}_{:,j})$ 
8:   end for
9: end for

```

The proposed feature extraction method can be expressed as an algorithm, as shown in Algorithm 1. The function *emotionAnnotation* extracts the emotion annotation vector corresponding to word w_k from matrix $\mathbf{L}$. Also, the function *statMeasure* computes the corresponding statistical measure from the column j of matrix $\mathbf{E}$. The index i indicates which statistical measure is applied (e.g., average if $i = 1$, maximum if $i = 2$, etc.).

As seen in the experiments, these emotion-based features can yield interestingly high scores, reaching an F-Score of 92.41% in a known dataset. It is reasonable to assume that these emotional features can be useful when performing radicalization detection. We have

also performed a cross-dataset evaluation in order to further assess the effectiveness of this feature extraction method. Results from the cross-dataset experiments indicate that the proposed combination can help with generalization issues. That is, combining the emotion features in a cross-dataset setting can improve the final performance, resulting in a more robust system.

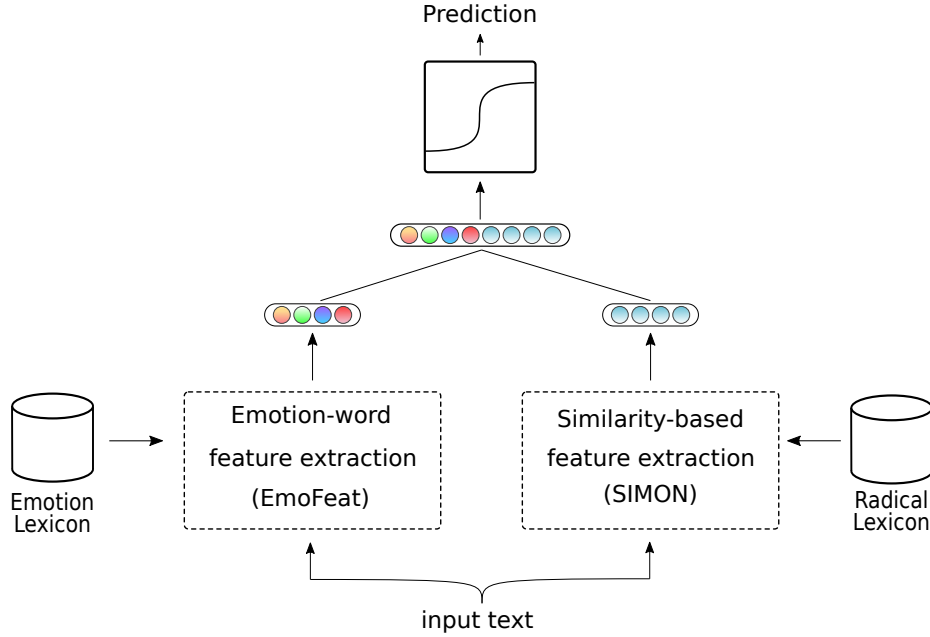


Figure 4.7: Architecture of the radicalization detection model, that combines a novel emotion representation with the SIMON adapted to this domain.

As for the second approach, the SIMON model has been adapted to extract radicalization oriented features. As stated, the application of SIMON towards this domain is achieved by composing domain-oriented lexicons. The generation of domain adapted lexicons is a challenge in itself; please see “**Neural domain adaptation of sentiment lexicons**” (Section A.2.5). However, in this work, we proposed a simple method for the generation of a domain-specific lexicon. Please note that numeric annotations are not needed for the use of a lexicon made by the SIMON method. Thus, a collection of words is generated through filtering by the frequency of appearance in the training data. This selection of words is used as a lexicon for the SIMON method. Such a selection method is oriented to represent a simple baseline to capture the vocabulary of a specific domain. We refer to this frequency-based selection as *FreqSelect*. Given the experimental results, it is clear that this kind of feature can accurately represent the analyzed text, allowing simple classifiers to obtain high classifications metrics: 99.11%, 86.22%, and 94.02% for the three datasets used in the evaluation. Thus, given these promising results, we conclude that the application of this method could be further studied.

Moreover, we have evaluated the use of the vocabulary obtained with *FreqSelect*, comparing it with similar selections. Seeing the results, we believe that such an approach performs positively. Also, as an additional contribution, the *Magazines dataset* is presented, which expands the scope of this and future research.

This work that contributes to the field of radicalization detection is presented in the journal publication “**An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity**” (Section 3.2.1).

Another research work that tackles the challenge of radicalization detection is presented in “**Radical Text Detection based on Stylometry**” (Section A.2.1), where style features are used to aid a machine learning system to detect radical content. Such features, when used in combination with bigram representations, improve the classification results.

As mentioned, this thesis has also contributed to the field of moral values estimation. We do so by operating in reference to the Moral Foundations Theory (MFT) (see Section 1.2). In this work, we have presented *MoralStrength*, a lexicon of approximately 1,000 lemmas, obtained as an extension of the Moral Foundations Dictionary (MFD), based on WordNet synsets. For this, we started from the original Moral Foundations Dictionary (MFD), expanding the vocabulary employing WordNet synsets. After this, manual filtering was done in order to remove forms that arose but are not related to moral values. Moreover, for each lemma *MoralStrength* provides with a crowdsourced numeric assessment of Moral Valence, indicating the strength with which a lemma is expressing the specific value. The presented resource includes the moral value annotations using a Likert scale, ranging from 1 to 9. Still, for computational purposes, this scale can be transformed without losing information. Figure 4.8 shows how this lexicon is constructed and how it can be used for posterior moral value estimation.

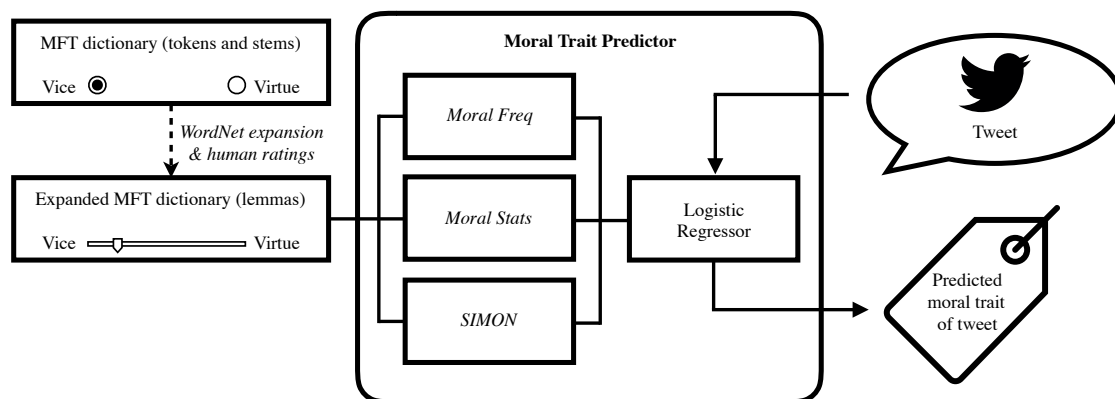


Figure 4.8: Overview of the process, from dictionary expansion to moral value prediction.

For the generated moral lexicon, we propose the following feature extraction approaches, which can be divided into those that solely exploit the semantic information of each word, and those who exploit the moral valence associated to the word. More specifically, we propose three lexicon utilization approaches: (i) frequency counts, (ii) statistical summary, and (iii) word embedding similarity-based representations using the SIMON model. The two first approaches use both the words and their moral values, while the third one makes use solely of the selection of words, ignoring the associated numeric moral values. Figure 4.9 illustrates these approaches. These three methods described above are used as feature extractors. In the next experiments, we feed these features to a logistic regression classifier. Such a simple learning algorithm is used to evaluate the performance of the proposed features, without exploiting more complex learning methods.

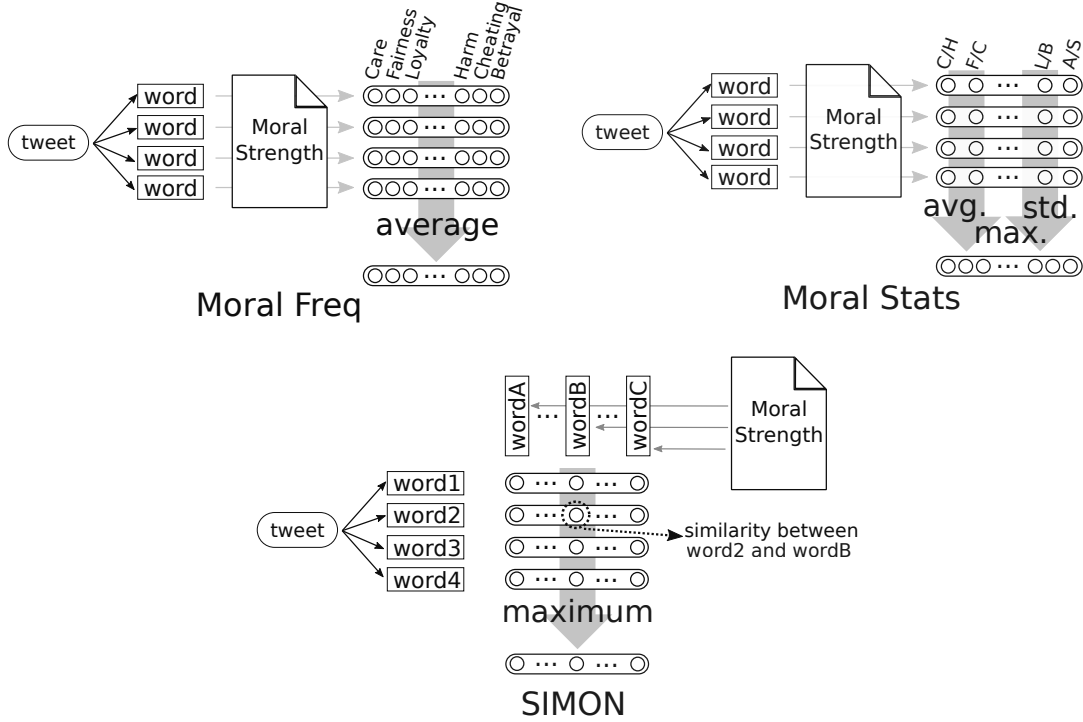


Figure 4.9: Diagram of the proposed feature extraction approaches that utilize the presented MoralStrength lexicon.

The two first methods (i and ii), are called Moral Freq. and Moral Stats, respectively. ***Moral Freq.*** consists of counting the number of words that express a specific moral dimension in a binary way. To decide if a specific word expresses a moral, we apply a simple rule: if the word has its moral value lower than a certain threshold, it does not convey that moral; if higher, the word does express that moral. Given the properties of the generated moral lexicon, the threshold is set at 5. We represent a given text with a 10-dimensional

vector, which contains the corresponding normalized frequency counts, each for each moral extremity. For instance, *care/harm* are represented by two dimensions, one for *care* and other for *harm*. As for ***Moral Stats.***, given a specific text, we generate a statistical summary of the moral valence distribution of the contained words in the text. In the statistical summary, we included (i) the average, (ii) the standard deviation, (iii) the median, and (iv) the maximum value. As a result, the text is represented by a 20-dimensional vector consisting of the statistical values obtained from the lexicon annotations. In relation to how we use the SIMON method, we project the analyzed text over the selection of words from *MoralStrength*, resulting in a vector representation that encodes the similarity of the document to the specific moral dimension. That is, for each moral trait, the words extracted from the proposed moral lexicon have been used as the set of selected words for the SIMON method.

In order to evaluate the proposed methods, an extensive evaluation has been carried out. We have used six distinct datasets, extracted from a recent repository known as the Moral Foundations Twitter Corpus (Joseph Hoover et al., 2019). We also compare against previous state-of-the-art models and a unigram baseline. Thus, we have evaluated moral-wise all our models, and combinations of those (e.g., SIMON + Moral Freq., unigrams + SIMON + Moral Stats), using both over-sampling and under-sampling. For enforcing our conclusions, we have included a Friedman test (see Section 2.4). Interestingly, all our models improve the prediction performance with respect to the current state-of-the-art for all moral dimensions. The most prominent approaches - as indicated by the Friedman ranking - combine pure textual (e.g., unigrams) with lexicon-based representations (e.g., the *Moral Freq*, the *Moral Stats*, and *SIMON*). Hence, we argue that moral lexicons can be successfully employed for moral values classification from a given text since when this information is considered, the models yield higher performance.

The work that advances on moral value estimation, including the lexicon and model proposal and their evaluation, are presented in the journal publication “**MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction**” (Section 3.2.2).

4.2.5 Other

Table 4.5: Publications not directly related to the core of this thesis.

Page	Title	Impact	Contribution
221	Senpy: A Pragmatic Linked Sentiment Analysis Framework		Definition of a linked data framework for Sentiment and Emotion Analysis.

Table 4.5: Publications not directly related to the core of this thesis.

176	Senpy: A framework for semantic sentiment and emotion analysis services	Q1 (5.101)	Reference implementation of a linked data framework for Sentiment and Emotion Analysis.
180	GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter		Adaptation of the SIMON model to Hate Speech detection, also exploiting the vocabulary selection and its characteristics. Participation in the international competition SemEval 2019. This work qualified for the 5th position globally.
189	How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining		Application of the SIMON methodology to insomnia detection in Twitter. Exploitation of similarity-based feature extraction and vocabulary selection.

Finally, there are additional contributions that are aligned with the lines of this thesis, but do not directly advance the main objectives. Two of these publications address the definition and implementation of a framework for Sentiment and Emotion Analysis online, based on linked-open data techniques. These are “**Senpy: A Pragmatic Linked Sentiment Analysis Framework**” (Section A.2.8) and “**Senpy: A framework for semantic sentiment and emotion analysis services**” (Section A.2.2). We have also contributed with two additional publications that apply or exploit the SIMON method to different domains, in line with Objective 4. These are “**GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter**” (Section A.2.3) and “**How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining**” (Section A.2.4).

4.3 Software Publication

Nowadays, software tools that accompany scientific publications are valued since they allow researchers to replicate the work, as well as advance into the topic. This thesis has, as additional results, software tools that implement the proposed models, as well as evaluation implementations. We detail a list of the software tools this thesis has produced, which is detailed in Table 4.6:

- A collection of Sentiment Analysis resources (4.3.1) to perform the analysis out of the box.
- The `gsitk` library (4.3.2), that gathers the following implementations:
 - Surface feature extraction methods.
 - Deep feature extraction methods.

- Lexicon feature extraction methods.
- Word embedding based methods.
- The SIMilarity-based sentiment projectiON (SIMON) implementation
- The SEntiment adapted to the Domain Lexicon (SEDLex) implementation (4.3.3), for generating domain-specific lexicons.
- The DepecheMood++ resource (4.3.4), and evaluation code.
- The MoralStrength (4.3.5) resource, with evaluation and code examples. Also, a graphical interface is available (4.3.6).

Ref.	Name	URL
4.3.1	Sentiment Analysis Resources	https://github.com/oaraque/sentiment-analysis
4.3.2	gsitk library	https://github.com/gsi-upm/gsitk
4.3.3	SEDLex implementation	https://lab.gsi.upm.es/oaraque/sedlex
4.3.4	DepecheMood++ resource and code	https://github.com/marcoguerini/DepecheMood
4.3.5	MoralStrength resource and code	https://github.com/oaraque/moral-foundations
4.3.6	MoralStrength GUI	https://github.com/oaraque/moral-foundations-gui

Table 4.6: Overview of code, resources, and utilities of this thesis.

4.4 Conclusions

Sentiment and Emotion Analysis, which we aggregate as Affect Analysis, represent prominent research domains in the field of Natural Language Processing. Affect Analysis has many applications, such as marketing, reputation management, customer support, competitor management, etc., and daily we can find new ones. One of the most exciting aspects of Affect Analysis is that the methods, techniques, and scientific advancements that occur can be applied to a plethora of other domains and fields. Such a characteristic makes Affect Analysis a field with scientific interest. This thesis is an example of this aspect of scientific advancement. In this way, we have made a number of relevant contributions in the context of both Sentiment and Emotion Analysis. After conducting thorough evaluations and studies of the designed models, we have then investigated the application and adaptation of some of these techniques in more recent fields.

In the first set of contributions, we can find the works on Sentiment Analysis. We have studied the effectiveness of surface features with classical machine learning models. With this kind of system, we have participated in an international competition (TASS 2015). Following, this thesis has contributed with a taxonomy that organizes the body of literature on Sentiment Analysis, identifying the types of features and combinations that can be done. In short, we define two main types of features: surface and deep features. We also study and evaluate the effectiveness of these features, combining them at different levels: at the feature and model level. The idea of implementing model ensembles is very effective, and we have experimentally verified that it can reach high overall performance. This study represents a relevant asset to the thesis since it establishes an experimental framework that is followed during the development of the thesis. As a note, the article where this work is presented, **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5), has received 224 cites (as reported by Google Scholar; Scopus indicates 152, and Web of Science 103, all visited on 2nd September 2020).

Next, we have applied these advancements to aspect-based Sentiment Analysis, a more granular study, with positive results. Such a study establishes that the presented learning models are, in fact, adaptable to different applications within Sentiment Analysis. Another observation that emerges is that neural networks are a central method in both Affect Analysis and, in general, Natural Language Processing. This is to be expected since, in the last years, the use of Deep Learning has dramatically increased.

In the previous contributions, lexicon resources have been studied in combination with other types of features. Nevertheless, sentiment and emotion lexicons are a key element in the vast majority of learning systems. These resources encode subjective affect information, making explicit this affect signals to the model. As the second bag of contributions, this thesis advances into the process of both sentiment and emotion lexicon generation. One of the models that this thesis proposes and explores exploits the training mechanism of classic and modern neural networks: backpropagation. We exploit this method to generation sentiment annotations to a given vocabulary, by training a neural network that generates the numerical annotations, but is not used for classification later. It is interesting how this method works, since it uses a known mechanism of neural networks, but does not makes use of an extensive architecture, nor requires enormous computing power. In the study of this sentiment lexicon generation, we assess its effectiveness in comparison to previous lexicons, observing that its performance is superior even when the domain is shifted. In a similar line of work, another contribution of this thesis consists of an application of a computational technique, Compositional Semantics, to generate bilingual emotion lexicons. We call these resources *DepecheMood++*. Apart from the resources, in this work, this thesis contributed

with an extensive evaluation that utilizes a wide range of tasks, datasets, and comparison against state-of-the-art models. All these experiments offer an interesting insight into the capacities of the resources and can establish a framework for future scientific work.

It has been already established the effectiveness of neural network approaches in NLP. Following this trend, the thesis has addressed the research area of distributional semantics. In this way, we have advanced into the use of such techniques, with the proposal and implementation of a novel model that effectively exploits the capacities of distributional semantics methods; that is, word embedding models. As discussed, the center of this process is the SIMilarity-based sentiment projectiON (SIMON) model and its adaptability to any domain. We believe that a relevant aspect of the SIMON model is the knowledge sources it uses (a word embedding model and a domain lexicon), as well as its adaptability to any domain. SIMON exploits the similarity measure that can be computed in a word embedding model word-wise, as well as the domain information contained in a lexicon. These two elements are very prevalent in both Sentiment and Emotion Analysis but are also widely used in other domains. In fact, one of the most common approaches to model domain knowledge consists in generating a lexicon that expresses that information. We have experimentally observed the effectiveness of the SIMON model in a variety of datasets when using different lexicons. A battery of statistical tests has supported these results. One of the relevant observations in relation to SIMON is that we have seen that using a lexicon through a word embedding model, even if it is a generic model, increases the performance in comparison to word-matching methods (Section 1.1). This could serve as a reference for future utilization. Another aspect of this study is the comparison between embedding and taxonomy-based semantic similarity. In this way, we have evaluated the performance of both WordNet and embedding based features, observing that embedding features yield better results. We argue that this difference is due to the difference in vocabulary coverage, which is intimately related to the WordNet’s generation process and embedding models resources. While WordNet is a manually-generated lexical resource, thus it is limited in coverage; embedding models are automatically inferred, which leads to more coverage in terms of vocabulary.

Additionally, this work has allowed the analysis of the characteristics of the lexicons we use, experimenting how varying the conditions of the lexicons affects the performance of the method. As indicated, we conclude that the selection of lexicon, with a focus on the word selection, greatly affects the performance of the final learning system. This relation is relevant to the point that we have been able to correlate between the differences in lexicon words and final performance.

This thesis tackles the adaptation of the SIMON model to other research areas. We have already discussed how the design of SIMON allows us to exploit it in diverse scenarios with positive results. In this way, we have adapted our method to radicalization detection. This work is interesting since not all NLP techniques are used in this research area, and transferring them can result beneficial. We have observed that the SIMON model performs effectively in this domain, achieving high-performance metrics. In addition, this thesis does not only transfer the SIMON model to radicalization detection but also some techniques from Emotion Analysis. This follows previous studies in the literature that highlight the role of emotions in the radicalization process. We have developed a method for extracting emotion information from text using an emotion lexicon, aggregating it to SIMON features, and training basic learning models to detect radicalization. This is an interesting addition, and to the extent of our knowledge, the effect of the emotions present in the text over a learning system has not been thoroughly studied in the task of radicalization detection. These proposals achieve high scores in three distinct datasets (one of those we have compiled), which we believe represents an exciting advancement, and hope that this will foster research in this area.

Another example of how the computation methods can be applied to another domain is the work we have done in moral value estimation. We have proposed a novel lexicon for the domain, that encodes moral values for words, using crowdsourcing. Attending to the results we have obtained in both Sentiment Analysis and radicalization detection, this work delves into the application of the SIMON model. In this way, this work utilizes SIMON to efficiently exploit the generated lexicon to perform moral value estimation on a large number of domain-specific datasets. We also have proposed two additional methods from utilizing said lexicon, based on keyword matching. The experimental results indicate that, in this domain, SIMON emerges as an effective method for moral value estimation, and that combining its features with unigrams achieves the best results. In fact, this work achieves state-of-the-art results in some datasets. Thus, we have observed that the SIMON model can offer high performance in a domain so different such as moral value estimation. This represents an interesting insight, indicating that the developed models are effective and that this line of research represents a fruitful path indeed.

The adaptability of the SIMON model is further evaluated in another two subsequent works that are not central to this thesis. Those works tackle (i) Hate Speech and (ii) Insomnia detection. In these works, we have observed this same trend: properly adapting the SIMON model yields high performances in a wide variety of NLP areas. The broad diversity of domains where our approach has been evaluated serves as indicative of the method’s validity. We hope that this will aid future research.

We believe that incorporating computational approaches into research areas that do not normally benefit from them can result highly beneficial. There are a number of social science challenges and questions that can be addressed with computational approaches. The capacity to collect and analyze massive amounts of data has transformed fields such as biology and physics. We believe that the same can happen to social science. In fact, a new research field is emerging, called computational social science (Lazer et al., 2009). Even if it is slow, the number of publications that fit these criteria is increasing, and we expect a larger growth.

In conclusion, interdisciplinary studies represent one of the most attractive and innovative forms of research. We expect that with this thesis, we have contributed, at least slightly, to foster research in this direction.

Lastly, with this thesis, the proposed hypotheses have been validated:

Hypothesis 1. *The combination of deep and surface textual features, including affect lexicons, enhances final Sentiment Analysis performance.*

It has been supported mainly by the results of our work on the combination of features and models for SA, where the experimental validation shows that results can be greatly improved in these cases. This work is detailed in **“Enhancing deep learning sentiment analysis with ensemble techniques in social applications”** (Section 3.2.5). It is also supported by posterior works that apply these techniques, generally reporting positive results.

Hypothesis 2. *A method that exploits both distributional representations and affect lexicons through semantic similarity can be leveraged for Sentiment Analysis effectively.*

Such a hypothesis has been validated with our proposed method SIMilarity-based sentiment projectiON (SIMON). Indeed, this method exploits the mentioned information sources and has shown positive results in Sentiment Analysis when used in an extensive evaluation. This work is described in **“A semantic similarity-based perspective of affect lexicons for sentiment analysis”** (Section 3.2.3).

Hypothesis 3. *A model implementing a semantic similarity-based exploitation of a lexicon for Sentiment Analysis can be transferred to other Natural Language Processing areas that show domain-specific traits (e.g., radicalization detection, moral value estimation).*

This hypothesis has been validated by our publications **“An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity”** (Section 3.2.1) and **“MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”** (Section 3.2.2), which illustrate how our SIMON model can be adapted to different domains, such as radicalization detection and moral value esti-

mation. These work both obtained positive results. In the case of radicalization detection, our methods have achieved high-performance metrics. In contrast, in the case of moral value estimation, our work has surpassed the performance of the previous state-of-the-art approach.

4.5 Future Research

During this thesis, we have visited several research areas and have been able to identify a variety of paths that can enhance existing research. This section discusses them, offering a summary of the directions that future research could take.

This thesis has shown how surface and deep features, when combined, can offer the best performance in Sentiment Analysis. One of the types of features that is included in deep features are sentiment oriented embeddings. That is, word vectors that have been generated by a model that considers both generic word embedding information and sentiment orientation (Section 1.2.2). Still, new models that exploit sentiment lexicons, incorporating this information into word embeddings, could be very interesting. Since it is difficult to combine word embeddings and lexicon features, this combination at the stage of embedding training could enhance the capacity of regular embeddings. Another improvement could lie in embedding surface features, that also offer useful information, normalizing in this way the dimensionality and characteristics of the feature vectors used for later classification. In fact, one could understand traditional word embedding models as the result of applying the embedding framework to unigram features. This thesis has shown that combining features and models can greatly enhance the overall performance in Sentiment Analysis. Following the previous ideas, performing these combinations through operations inside a neural network could improve said combination, allowing for the learning algorithm to adapt this combination automatically.

In other lines of this thesis, we have discussed how the presented SEntiment adapted to the Domain Lexicon (SEDLex) model allows us to generate a domain-adapted lexicon in Sentiment Analysis. SEDLex implements this by exploiting the backpropagation algorithm, generating sentiment annotations of a unique dimension. In order to model emotion annotations, our method would require to do this in several dimensions. It is not clear if this methodology would work when considering an arbitrary number of dimensions or if the obtained results could be comprehensible by humans. Research in this line can be interesting, considering that the SEDLex method can be easily adapted to other research areas, as we have done in this thesis. In this way, our method could allow us to generate radicalization or moral value lexicons with the resources we currently have.

We believe it is also interesting that the SEDLex approach has higher complexity in comparison to the proposal based on compositional semantics (from which we have obtained DepecheMood++). Future research should explore under what circumstances these two approaches improves, taking into account parameters such as obtained performance, complexity, and required data. In this way, we could obtain further insight into the applicability of each model. Future work could even explore the combination of the two approaches, maybe through using the compositional semantics approach to initialize the SEDLex method.

On the improvement of the SIMON model, there is a straightforward approach that could be taken. It has been explained how SIMON generates a matrix of similarities that is later reduced to a vector by means of a pooling function. Instead of using such a simple technique, future work could reduce the dimensionality using a Convolutional Neural Network (CNN) or Recurrent Neural Networks (RNN) neural network. This, of course, would increment the complexity of the method, but could also improve its performance. That is, implementing this kind of operation would allow the network to select from a high number of reductions, training it with annotated data. On another line of work, this thesis has shown how the vocabulary selection of a lexicon greatly affects the classification performance. Consequently, improving the selection methods could potentially improve said performance. For this, future research could explore new information sources to include in the vocabulary selection, including frequency in the dataset (e.g., TF-IDF) or importance as given by a classifier. Joining several metrics that indicate the importance of a set of words could improve the selection mechanism.

Regarding our work in radicalization detection, many lines of research could be addressed. As seen, with the current methods we have developed for generating lexicons, we could generate a lexicon for radicalization detection. This resource could be used for improved learning models, but also for obtaining further insights into the radical language, answering questions such as “Which words denote a more radical tone?”. Additionally, obtaining domain-oriented word vectors could largely improve the text representations, leading to an improvement in classification performance. As in the case of Sentiment Analysis, these word embeddings would incorporate generic semantic information, as well as radicalization specific knowledge. In a similar way that before, this resource could also offer researchers information regarding radical language use. In more exploratory work, studying the effect of temporal effects on the radical narratives can be interesting, since the use of the language is prone to change as time progresses.

In the field of moral value estimation, this thesis introduces a lexicon, MoralStrength, that directly encodes moral values. In this thesis, we have worked into expanding lexicons

through word embeddings, thus incorporating new words and transferring the lexicon's annotations. Of course, this same process can be adapted to this area, enhancing the lexicon we have produced. Also, similarly to the previous future work, it would be interesting to explore how specific knowledge could be encoded in domain-oriented word vectors, allowing for the development of complex learning methods. Moreover, the word embedding representation based on moral similarity could be enhanced with the obtained assessments of moral valence, or even combined with sentiment features from the analyzed text. As for the social sciences, there are numerous issues where detecting the morals narrative can significantly improve our understanding of the peoples' dispositions in, for instance, controversial social phenomena as well as the evolution of opinions over time.

In a general note, this thesis has proved that techniques that are developed in fields such as Sentiment and Emotion Analysis can be adapted to a wide variety of research areas. This adaptation process opens a door for future research in an array of domains. We believe that the modeling techniques presented (machine learning models, word embeddings, lexical resources) can be directly applied to many areas, allowing researchers to implement learning capabilities as needed. We hope that with our contribution, and those of many others, future research will benefit from computer science, opening to new promising new avenues.

Bibliography

- Abbasi, Ahmed and Hsinchun Chen (2007). “Affect intensity analysis of dark web forums”. In: *2007 IEEE Intelligence and Security Informatics*. IEEE, pp. 282–288. DOI: 10.1109/ISI.2007.379486.
- Agarwal, Apoorv et al. (2011). “Sentiment analysis of twitter data”. In: *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pp. 30–38.
- Agarwal, Swati and Ashish Sureka (2015a). “Applying social media intelligence for predicting and identifying on-line radicalization and civil unrest oriented threats”. In: *arXiv preprint arXiv:1511.06858*.
- (2015b). “Topic-specific youtube crawling to detect online radicalization”. In: *International Workshop on Databases in Networked Information Systems*. Springer, pp. 133–151. DOI: 10.1007/978-3-319-16313-0_10.
- (2015c). “Using knn and svm based one-class classifier for detecting online radicalization on twitter”. In: *International Conference on Distributed Computing and Internet Technology*. Springer, pp. 431–442. DOI: 10.1007/978-3-319-14977-6_47.
- Agathangelou, Pantelis et al. (Apr. 2018). “Learning patterns for discovering domain-oriented opinion words”. In: *Knowledge and Information Systems* 55.1, pp. 45–77. ISSN: 0219-3116. DOI: 10.1007/s10115-017-1072-y.
- Alizadeh, Meysam et al. (2019). “Psychology and morality of political extremists: evidence from Twitter language analysis of alt-right and Antifa”. In: *EPJ Data Science* 8.1, p. 17. DOI: 10.1140/epjds/s13688-019-0193-9.
- Almatarneh, Sattam and Pablo Gamallo (2017). “Automatic Construction of Domain-Specific Sentiment Lexicons for Polarity Classification”. In: *15th International Conference on Practical Applications of Agents and Multi-Agent Systems*. Ed. by Springer. Fernando De la Prieta, Zita Vale, Luis Antunes, Tiago Pinto, Andrew Campbell, Vicente Julián, Antonio J.R. Neves, María N. Moreno. DOI: 10.1007/978-3-319-61578-3_17.
- Alpaydin, Ethem (2014). *Introduction to machine learning*. MIT press.
- Ambartsoumian, Artaches and Fred Popowich (2018). “Self-attention: A better building block for sentiment analysis neural network classifiers”. In: *arXiv preprint arXiv:1812.07860*. DOI: 10.18653/v1/W18-6219.

- Araque, Oscar, Rodrigo Barbado, et al. (2017). “Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets”. In: *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN*. CEUR Workshop Proceedings 1896. URL: http://ceur-ws.org/Vol-1896/p8_gsi_tass2017.pdf.
- Araque, Oscar, Ignacio Corcuera-Platas, et al. (July 2017). “Enhancing deep learning sentiment analysis with ensemble techniques in social applications”. en. In: *Expert Systems with Applications* 77, pp. 236–246. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2017.02.002. URL: <http://www.sciencedirect.com/science/article/pii/S0957417417300751> (visited on 01/14/2020).
- Araque, Oscar, Ignacio Corcuera, et al. (2015). “Aspect Based Sentiment Analysis of Spanish Tweets”. In: *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015*. Ed. by Julio Villena-Román et al. Vol. 1397. CEUR Workshop Proceedings. CEUR-WS.org, pp. 29–34. URL: <http://ceur-ws.org/Vol-1397/gsi.pdf> (visited on 03/04/2020).
- Araque, Oscar, Lorenzo Gatti, and Kyriaki Kalimeri (Nov. 2019). “MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction”. en. In: *Knowledge-Based Systems*, p. 105184. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2019.105184. URL: <http://www.sciencedirect.com/science/article/pii/S095070511930526X> (visited on 01/14/2020).
- Araque, Oscar, Lorenzo Gatti, Jacopo Staiano, et al. (2019). “DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques”. In: *IEEE Transactions on Affective Computing*, pp. 1–1. ISSN: 2371-9850. DOI: 10.1109/TAFEC.2019.2934444.
- Araque, Oscar, Marco Guerini, et al. (Oct. 2017). “Neural domain adaptation of sentiment lexicons”. In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. ISSN: null, pp. 105–110. DOI: 10.1109/ACIIW.2017.8272598.
- Araque, Oscar and Carlos A. Iglesias (2020). “An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity”. In: *IEEE Access* 8, pp. 17877–17891. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2967219.
- Araque, Oscar, Ganggao Zhu, Manuel García-Amado, et al. (Dec. 2016). “Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis”. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. ISSN: 2375-9259, pp. 900–907. DOI: 10.1109/ICDMW.2016.0132.
- Araque, Oscar, Ganggao Zhu, and Carlos A. Iglesias (Feb. 2019). “A semantic similarity-based perspective of affect lexicons for sentiment analysis”. en. In: *Knowledge-Based*

- Systems* 165, pp. 346–359. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2018.12.005. URL: <http://www.sciencedirect.com/science/article/pii/S0950705118305926> (visited on 01/14/2020).
- Ashcroft, Michael et al. (2015). “Detecting jihadist messages on twitter”. In: *Intelligence and Security Informatics Conference (EISIC), 2015 European*. IEEE, pp. 161–164. DOI: 10.1109/EISIC.2015.27.
- Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani (2010). “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining”. In: *LREC*. Vol. 10, pp. 2200–2204.
- Badaro, Gilbert et al. (2014). “A large scale Arabic sentiment lexicon for Arabic opinion mining”. In: *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 165–173. DOI: 10.3115/v1/W14-3623.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *CoRR*. arXiv: 1409.0473.
- Bandhakavi, A. et al. (Jan. 2017). “Lexicon Generation for Emotion Detection from Text”. In: *IEEE Intelligent Systems* 32.1, pp. 102–108. ISSN: 1541-1672. DOI: 10.1109/MIS.2017.22.
- Baroni, Marco, Georgiana Dinu, and Germán Kruszewski (2014). “Don’t count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors”. In: *ACL (1)*, pp. 238–247. DOI: 10.3115/v1/P14-1023.
- Bengio, Yoshua (2009). “Learning Deep Architectures for AI”. In: *Foundations and Trends® in Machine Learning* 2.1, pp. 1–127. ISSN: 1935-8237. DOI: 10.1561/2200000006.
- Benito, Diego, Oscar Araque, and Carlos A. Iglesias (2019). “GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter”. en. In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA: Association for Computational Linguistics, pp. 396–403. DOI: 10.18653/v1/S19-2070. URL: <https://www.aclweb.org/anthology/S19-2070> (visited on 01/14/2020).
- Bermingham, Adam et al. (2009). “Combining social network analysis and sentiment analysis to explore the potential for online radicalisation”. In: *Social Network Analysis and Mining, 2009. ASONAM’09. International Conference on Advances in*. IEEE, pp. 231–236. DOI: 10.1109/ASONAM.2009.31.
- Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). “Latent Dirichlet allocation”. In: *Journal of Machine Learning research* 3, pp. 993–1022.
- Blitzer, John, Mark Dredze, and Fernando Pereira (2007). “Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification”. In: *Proceedings of the 45th annual meeting of the association of computational linguistics*, pp. 440–447.

- Bojanowski, Piotr et al. (2017). “Enriching word vectors with subword information”. In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. DOI: 10.1162/tacl_a_00051.
- Borth, Damian et al. (2013). “Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content”. In: *Proceedings of the 21st ACM international conference on Multimedia*. ACM, pp. 459–460. DOI: 10.1145/2502081.2502268.
- Boyd, Ryan L et al. (2015). “Values in Words: Using Language to Evaluate and Understand Personal Values”. In: *Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM)*. Oxford, UK, pp. 31–40.
- Bradley, Margaret M. and Peter J. Lang (1999). “Affective norms for English words (ANEW): Instruction manual and affective ratings”. In: *Technical Report C-1, University of Florida*.
- Bravo-Marquez, Felipe, Marcelo Mendoza, and Barbara Poblete (2014). “Meta-level sentiment models for big social data analysis”. In: *Knowledge-Based Systems* 69, pp. 86–99. DOI: 10.1016/j.knosys.2014.05.016.
- Buechel, Sven and Udo Hahn (2018). “Emotion Representation Mapping for Automatic Lexicon Construction (Mostly) Performs on Human Level”. In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2892–2904.
- Cambria, Erik, Thomas Mazzocco, et al. (2011). “Sentic medoids: Organizing affective common sense knowledge in a multi-dimensional vector space”. In: *Proceedings of the International Symposium on Neural Networks*, pp. 601–610. DOI: 10.1007/978-3-642-21111-9_68.
- Cambria, Erik, Björn Schuller, et al. (2013). “New avenues in opinion mining and sentiment analysis”. In: *IEEE Intelligent Systems* 28.2, pp. 15–21. DOI: 10.1109/MIS.2013.30.
- Chalothorn, Tawunrat and Jeremy Ellman (2013). “Affect Analysis of Radical Contents on Web Forums Using SentiWordNet”. In: *International Journal of Innovation Management and Technology* 4.1, pp. 122–124.
- Chen, Hsinchun (2008). “Sentiment and affect analysis of dark web forums: Measuring radicalization on the internet”. In: *Intelligence and Security Informatics, 2008. ISI 2008. IEEE International Conference on*. IEEE, pp. 104–109. DOI: 10.1109/ISI.2008.4565038.
- Chen, Jilin et al. (2014). “Understanding individuals’ personal values from social media word use”. In: *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, pp. 405–414. DOI: 10.1145/2531602.2531608.
- Chen, Tao et al. (2017). “Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN”. In: *Expert Systems with Applications* 72, pp. 221–230. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2016.10.065.

- Chevalier, Judith A and Dina Mayzlin (2006). “The effect of word of mouth on sales: Online book reviews”. In: *Journal of marketing research* 43.3, pp. 345–354. DOI: 10.1509/jmkr.43.3.345.
- Choi, Yejin and Claire Cardie (2009). “Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification”. In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, pp. 590–598. DOI: 10.3115/1699571.1699590.
- Church, Kenneth Ward and Patrick Hanks (1990). “Word Association Norms, Mutual Information, and Lexicography”. In: *Comput. Linguist.* 16.1, pp. 22–29. ISSN: 0891-2017. DOI: 10.3115/981623.981633.
- Clifford, Scott and Jennifer Jerit (2013). “How words do the work of politics: Moral foundations theory and the debate over stem cell research”. In: *The Journal of Politics* 75.3, pp. 659–671. DOI: 10.1017/S0022381613000492.
- Collobert, Ronan et al. (2011). “Natural language processing (almost) from scratch”. In: *Journal of Machine Learning Research* 12.Aug, pp. 2493–2537.
- Conneau, Alexis et al. (2017). “Word translation without parallel data”. In: *arXiv preprint arXiv:1710.04087*.
- Correa, Denzil and Ashish Sureka (2013). “Solutions to detect and analyze online radicalization: a survey”. In: *arXiv preprint arXiv:1301.4916*.
- Crowston, Kevin, Eileen E Allen, and Robert Heckman (2012). “Using natural language processing technology for qualitative data analysis”. In: *International Journal of Social Research Methodology* 15.6, pp. 523–543. DOI: 10.1080/13645579.2011.625764.
- Dai, Andrew M and Quoc V Le (2015). “Semi-supervised Sequence Learning”. In: *Advances in Neural Information Processing Systems* 28. Ed. by C. Cortes et al. Curran Associates, Inc., pp. 3079–3087.
- Dehghani, Morteza et al. (2014). “Analyzing political rhetoric in conservative and liberal weblogs related to the construction of the “Ground Zero Mosque””. In: *Journal of Information Technology & Politics* 11.1, pp. 1–14. DOI: 10.1080/19331681.2013.826613.
- Demiroz, Gulsen et al. (2012). “Learning domain-specific polarity lexicons”. In: *Data mining workshops (ICDMW), 2012 IEEE 12th international conference on*. IEEE, pp. 674–679. DOI: 10.1109/ICDMW.2012.120.
- Demšar, Janez (2006). “Statistical comparisons of classifiers over multiple data sets”. In: *Journal of Machine learning research* 7.Jan, pp. 1–30.
- Deng, Lingjia and Janyce Wiebe (2015). “Mpqa 3.0: An entity/event-level sentiment corpus”. In: *Proceedings of the 2015 Conference of the North American Chapter of the Association*

- for *Computational Linguistics: Human Language Technologies*, pp. 1323–1328. DOI: 10.3115/v1/N15-1146.
- Devlin, Jacob et al. (2018). “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805*.
- Dewan, Prateek et al. (2017). “Towards understanding crisis events on online social networks through pictures”. In: *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM, pp. 439–446. DOI: 10.1145/3110025.3110062.
- Ding, Xiaowen, Bing Liu, and Philip S Yu (2008). “A holistic lexicon-based approach to opinion mining”. In: *Proceedings of the 2008 international conference on web search and data mining*. ACM, pp. 231–240. DOI: 10.1145/1341531.1341561.
- Du, Weifu et al. (2010). “Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon”. In: *Proceedings of the third ACM international conference on Web search and data mining*. ACM, pp. 111–120. DOI: 10.1145/1718487.1718502.
- Ekman, Paul (1992). “An argument for basic emotions”. In: *Cognition & emotion* 6.3-4, pp. 169–200. DOI: 10.1080/02699939208411068.
- Ekman, Paul and Wallace V. Friesen (1971). “Constants across cultures in the face and emotion”. In: *Journal of Personality and Social Psychology* 17, pp. 124–129. DOI: 10.1037/h0030377.
- Fernandez, Miriam, Moizzah Asif, and Harith Alani (2018). “Understanding the Roots of Radicalisation on Twitter”. In: *Proceedings of the 10th ACM Conference on Web Science*. WebSci ’18. Amsterdam, Netherlands: ACM, pp. 1–10. ISBN: 978-1-4503-5563-6. DOI: 10.1145/3201064.3201082. URL: <http://doi.acm.org/10.1145/3201064.3201082>.
- Ferrara, Emilio et al. (2016). “Predicting online extremism, content adopters, and interaction reciprocity”. In: *International conference on social informatics*. Springer, pp. 22–39. DOI: 10.1007/978-3-319-47874-6_3.
- Fontaine, Johnny RJ, Klaus R Scherer, and Cristina Soriano (2013). *Components of emotional meaning: A sourcebook*. Oxford University Press. DOI: 10.1093/acprof:oso/9780199592746.001.0001.
- Garten, Justin, Reihane Boghrati, et al. (2016). “Morality between the lines: Detecting moral sentiment in text”. In: *Proceedings of the IJCAI 2016 Workshop on Computational Modeling of Attitudes (WCMA)*. New York, NY, USA.
- Garten, Justin, Joe Hoover, et al. (2018). “Dictionaries and distributions: Combining expert knowledge and large scale textual data content analysis”. In: *Behavior Research Methods* 50.1, pp. 344–361. DOI: 10.3758/s13428-017-0875-9.

- Garten, Justin, Brendan Kennedy, et al. (2019). “Incorporating Demographic Embeddings Into Language Understanding”. In: *Cognitive science* 43.1. DOI: 10.1111/cogs.12701.
- Gatti, Lorenzo, Marco Guerini, and Marco Turchi (2016). “SentiWords: Deriving a high precision and high coverage lexicon for sentiment analysis”. In: *IEEE Transactions on Affective Computing* 7.4, pp. 409–421. DOI: 10.1109/TAFFC.2015.2476456.
- Ghajar-Khosravi, Shadi et al. (2016). “Quantifying salient concepts discussed in social media content: A case study using Twitter content written by radicalized youth”. In: *Journal of Terrorism Research* 7.2. DOI: 10.15664/jtr.1241.
- Gimpel, Kevin et al. (2010). *Part-of-speech tagging for twitter: Annotation, features, and experiments*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.
- Gligorov, Risto et al. (2007). “Using Google Distance to Weight Approximate Ontology Matches”. In: *Proceedings of the 16th International Conference on World Wide Web*. Banff, Alberta, Canada: ACM, pp. 767–776. ISBN: 978-1-59593-654-7. DOI: 10.1145/1242572.1242676.
- Graham, Jesse, Jonathan Haidt, and Brian A Nosek (2009). “Liberals and Conservatives Rely on Different Sets of Moral Foundations”. In: *Journal of Personality and Social Psychology* 96.5, p. 1029. DOI: 10.1037/a0015141.
- Grover, Ted et al. (2019). “Moral and affective differences in us immigration policy debate on twitter”. In: *Computer Supported Cooperative Work (CSCW)* 28.3-4, pp. 317–355.
- Hamilton, William L et al. (2016). “Inducing domain-specific sentiment lexicons from unlabeled corpora”. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2016. NIH Public Access, p. 595. DOI: 10.18653/v1/D16-1057.
- He, Ruining and Julian McAuley (2016). “Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering”. In: *proceedings of the 25th international conference on world wide web*, pp. 507–517. DOI: 10.1145/2872427.2883037.
- Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- Hoover, Joe et al. (2018). “Moral Framing and Charitable Donation: Integrating exploratory social media analyses and confirmatory experimentation”. In: *Collabra: Psychology* 4.1. DOI: 10.1525/collabra.129.
- Hoover, Joseph et al. (2019). “Moral Foundations Twitter Corpus: A collection of 35k tweets annotated for moral sentiment”. In: DOI: 10.1177/1948550619876629.
- Hu, Minqing and Bing Liu (2004). “Mining and summarizing customer reviews”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 168–177. DOI: 10.1145/1014052.1014073.

- Huang, Sheng, Zhendong Niu, and Chongyang Shi (2014). “Automatic construction of domain-specific sentiment lexicon based on constrained label propagation”. In: *Knowledge-Based Systems* 56, pp. 191–200. DOI: 10.1016/j.knosys.2013.11.009.
- Izard, Carroll E. (1977). *Human emotions*. Plenum Press. DOI: 10.1007/978-1-4899-2209-0.
- Jiang, Long et al. (2011). “Target-dependent twitter sentiment classification”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies- Volume 1*. Association for Computational Linguistics, pp. 151–160.
- Joulin, Armand et al. (2016). “Bag of Tricks for Efficient Text Classification”. In: *CoRR*. DOI: 10.18653/v1/E17-2068. arXiv: 1607.01759.
- Jurek, Anna, Maurice D Mulvenna, and Yaxin Bi (2015). “Improved lexicon-based sentiment analysis for social media analytics”. In: *Security Informatics* 4.1, p. 9. DOI: 10.1186/s13388-015-0024-x.
- Kaur, Rishemjit and Kazutoshi Sasahara (2016). “Quantifying moral foundations from various topics on Twitter conversations”. In: *Proceedings of the 2016 IEEE International Conference on Big Data (BigData)*. Washington D.C., USA, pp. 2505–2512. DOI: 10.1109/BigData.2016.7840889.
- Kim, Yoon (2014). “Convolutional Neural Networks for Sentence Classification”. In: *CoRR*. DOI: 10.3115/v1/D14-1181. arXiv: 1408.5882.
- Kiritchenko, Svetlana and Saif M. Mohammad (2016). “Capturing Reliable Fine-Grained Sentiment Associations by Crowdsourcing and Best–Worst Scaling”. In: *Proceedings of The 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. DOI: 10.18653/v1/N16-1095.
- Kouloumpis, Efthymios, Theresa Wilson, and Johanna D Moore (2011). “Twitter sentiment analysis: The good the bad and the omg!” In: *Icwsn* 11.538-541, p. 164.
- Lai, S. et al. (Nov. 2016). “How to Generate a Good Word Embedding”. In: *IEEE Intelligent Systems* 31.6, pp. 5–14. ISSN: 1541-1672. DOI: 10.1109/MIS.2016.45.
- Lazer, David et al. (2009). “Computational social science”. In: *Science* 323.5915, pp. 721–723.
- Le, Quoc and Tomas Mikolov (2014). “Distributed representations of sentences and documents”. In: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1188–1196.
- Lecun, Y. et al. (Nov. 1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. ISSN: 0018-9219. DOI: 10.1109/5.726791.

- Levy, Omer, Yoav Goldberg, and Ido Dagan (2015). “Improving distributional similarity with lessons learned from word embeddings”. In: *Transactions of the Association for Computational Linguistics* 3, pp. 211–225. DOI: 10.1162/tac1_a_00134.
- Li, Yang et al. (Dec. 2017). “Learning Word Representations for Sentiment Analysis”. In: *Cognitive Computation* 9.6, pp. 843–851. ISSN: 1866-9964. DOI: 10.1007/s12559-017-9492-2.
- Lin, Ying et al. (2018). “Acquiring background knowledge to improve moral value prediction”. In: *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. Barcelona, Spain, pp. 552–559. DOI: 10.1109/ASONAM.2018.8508244.
- Liu, Bing (2015). *Sentiment Analysis: Mining Opinions, Sentiments, and Emotions*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/CBO9781139084789.
- Liu, Bing and Lei Zhang (2012). “A survey of opinion mining and sentiment analysis”. In: *Mining text data*. Springer, pp. 415–463. DOI: 10.1007/978-1-4614-3223-4_13.
- López-Sánchez, Daniel et al. (2018). “Towards the Automatic Identification and Monitoring of Radicalization Activities in Twitter”. In: *International Conference on Knowledge Management in Organizations*. Springer, pp. 589–599. DOI: 10.1007/978-3-319-95204-8_49.
- Lu, Yue et al. (2011). “Automatic construction of a context-aware sentiment lexicon: an optimization approach”. In: *Proceedings of the 20th international conference on World wide web*. ACM, pp. 347–356. DOI: 10.1145/1963405.1963456.
- Maas, Andrew L et al. (2011). “Learning word vectors for sentiment analysis”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pp. 142–150.
- Manning, Christopher et al. (2014). “The Stanford CoreNLP natural language processing toolkit”. In: *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pp. 55–60. DOI: 10.3115/v1/P14-5010.
- McAdams, Dan P. and Jennifer L. Pals (Apr. 2006). “A new Big Five: Fundamental principles for an integrative science of personality”. In: *American Psychologist* 61.3, pp. 204–217. ISSN: 0003-066X. DOI: 10.1037/0003-066X.61.3.204.
- McAuley, Julian et al. (2015). “Image-based recommendations on styles and substitutes”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52. DOI: 10.1145/2766462.2767755.
- Mikolov, Tomas et al. (2013). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*. arXiv: 1301.3781.

- Miles, Andrew and Stephen Vaisey (2015). “Morality and politics: Comparing alternate theories”. In: *Social Science Research* 53, pp. 252–269. DOI: 10.1016/j.ssresearch.2015.06.002.
- Miller, George A (1995). “WordNet: a lexical database for English”. In: *Communications of the ACM* 38.11, pp. 39–41. DOI: 10.1145/219717.219748.
- Mohammad, Saif (2018). “Word Affect Intensities”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.
- Mohammad, Saif M. (2018). “Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*. DOI: 10.18653/v1/P18-1017.
- Mohammad, Saif M. and Peter D. Turney (2010). “Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon”. In: *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pp. 26–34.
- Mohammad, Saif and Svetlana Kiritchenko (2015). “Using Hashtags to Capture Fine Emotion Categories from Tweets”. In: *Computational Intelligence* 31.2, pp. 301–326. ISSN: 1467-8640. DOI: 10.1111/coin.12024.
- Mohammad, Saif, Svetlana Kiritchenko, and Xiaodan Zhu (2013). “NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets”. In: *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pp. 321–327. arXiv: 1308.6242.
- Mohammad, Saif and Peter Turney (2013). “Crowdsourcing a Word-Emotion Association Lexicon”. In: *Computational Intelligence* 29, pp. 436–465. DOI: 10.1111/j.1467-8640.2012.00460.x.
- Montgomery, DC (2005). “Design and Analysis of Experiments (6th edn.,) John Wiley and Sons”. In: *New York, NY*.
- Mooijman, Marlon et al. (2018). “Moralization in social networks and the emergence of violence during protests”. In: *Nature Human Behaviour* 2.6, pp. 389–396. DOI: 10.1038/s41562-018-0353-0.
- Muhammad, Aminu, Nirmalie Wiratunga, and Robert Lothian (2016). “Contextual sentiment analysis for social media genres”. In: *Knowledge-Based Systems* 108, pp. 92–101. DOI: 10.1016/j.knosys.2016.05.032.
- Mukherjee, Subhabrata and Pushpak Bhattacharyya (2012). “Feature specific sentiment analysis for product reviews”. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 475–487. DOI: 10.1007/978-3-642-28604-9_39.

- Nakov, Preslav et al. (2016). “SemEval-2016 task 4: Sentiment analysis in Twitter”. In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 1–18. DOI: 10.18653/v1/S16-1001.
- Neviarouskaya, Alena, Helmut Prendinger, and Mitsuru Ishizuka (2010). “EmoHeart: conveying emotions in second life based on affect sensing from text”. In: *Advances in Human-Computer Interaction 2010*, p. 1. DOI: 10.1155/2010/209801.
- Nielsen, Finn Årup (2011). “A new ANEW: Evaluation of a word list for sentiment analysis in microblogs”. In: *arXiv preprint arXiv:1103.2903*. arXiv: 1103.2903.
- Oehlert, Gary W (2010). *A first course in design and analysis of experiments*.
- Pablo, Álvaro de, Oscar Araque, and Carlos Angel Iglesias (2020). “Radical Text Detection based on Stylometry”. In: *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP 2020, Valletta, Malta, February 25-27, 2020*. Ed. by Steven Furnell et al. SCITEPRESS, pp. 524–531. DOI: 10.5220/0008971205240531. URL: <https://doi.org/10.5220/0008971205240531>.
- Pak, Alexander and Patrick Paroubek (2010). “Twitter as a corpus for sentiment analysis and opinion mining.” In: *LREc*. Vol. 10. 2010, pp. 1320–1326.
- Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan (2002). “Thumbs up?: sentiment classification using machine learning techniques”. In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. Association for Computational Linguistics, pp. 79–86. DOI: 10.3115/1118693.1118704.
- Peng, Haiyun and Erik Cambria (2018). “CSenticNet: a concept-level resource for sentiment analysis in Chinese language”. In: *Computational Linguistics and Intelligent Text Processing. Proceedings of CICLing: International Conference on Computational Linguistics and Intelligent Text Processing*. LNCS. Springer-Verlag, in press. DOI: 10.1007/978-3-319-77116-8_7.
- Pennebaker, James W, Martha E Francis, and Roger J Booth (2001). “Linguistic inquiry and word count: LIWC 2001”. In: *Mahway: Lawrence Erlbaum Associates* 71.2001, p. 2001.
- Pennington, Jeffrey, Richard Socher, and Christopher D Manning (2014). “Glove: Global vectors for word representation”. In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*. Vol. 12, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.
- Perez-Rosas, Veronica, Carmen Banea, and Rada Mihalcea (2012). “Learning sentiment lexicons in spanish”. In: *LREC*. Vol. 12, p. 73. DOI: 10.18180/teccienca.2017.22.5.
- Peters, Matthew E et al. (2018). “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365*. DOI: 10.18653/v1/N18-1202.

- Plutchik, Robert (1980). “A general psychoevolutionary theory of emotion”. In: *Theories of emotion* 1, pp. 3–31. DOI: 10.1016/B978-0-12-558701-3.50007-7.
- Pontiki, Maria et al. (2016). “Semeval-2016 task 5: Aspect based sentiment analysis”. In: *10th International Workshop on Semantic Evaluation (SemEval 2016)*. DOI: 10.18653/v1/S16-1002.
- Poria, Soujanya, Erik Cambria, et al. (2017). “A review of affective computing: From unimodal analysis to multimodal fusion”. In: *Information Fusion* 37, pp. 98–125. ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2017.02.003>. URL: <http://www.sciencedirect.com/science/article/pii/S1566253517300738>.
- Poria, Soujanya, Alexander Gelbukh, et al. (2014). “EmoSenticSpace: A novel framework for affective common-sense reasoning”. In: *Knowledge-Based Systems* 69, pp. 108–123. DOI: 10.1016/j.knosys.2014.06.011.
- Porter, Martin F et al. (1980). “An algorithm for suffix stripping.” In: *Program* 14.3, pp. 130–137. DOI: 10.1108/eb046814.
- Qiu, Guang et al. (2009). “Expanding Domain Sentiment Lexicon through Double Propagation.” In: *IJCAI*. Vol. 9, pp. 1199–1204.
- Ravi, Kumar and Vadlamani Ravi (2015a). “A survey on opinion mining and sentiment analysis: Tasks, approaches and applications”. In: *Knowledge-Based Systems* 89, pp. 14–46. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2015.06.015.
- (2015b). “A survey on opinion mining and sentiment analysis: tasks, approaches and applications”. In: *Knowledge-Based Systems* 89, pp. 14–46. DOI: 10.1016/j.knosys.2015.06.015.
- Read, Jonathon (2005). “Using emoticons to reduce dependency in machine learning techniques for sentiment classification”. In: *Proceedings of the ACL student research workshop*, pp. 43–48. DOI: 10.3115/1628960.1628969.
- Rezapour, Rezvaneh, Saumil H Shah, and Jana Diesner (2019). “Enhancing the Measurement of Social Effects by Capturing Morality”. In: *Proceedings of the Tenth Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 35–45. DOI: 10.18653/v1/W19-1305.
- Rice, Stephen K (2009). “Emotions and terrorism research: A case for a social-psychological agenda”. In: *Journal of Criminal Justice* 37.3, pp. 248–255. DOI: 10.1016/j.jcrimjus.2009.04.012.
- Rosenthal, Sara, Noura Farra, and Preslav Nakov (2017). “SemEval-2017 task 4: Sentiment analysis in Twitter”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 502–518. DOI: 10.18653/v1/S17-2088.

- Rowe, Matthew and Hassan Saif (2016). “Mining pro-ISIS radicalisation signals from social media users”. In: *Proceedings of the tenth international AAAI conference on web and social media (ICWSM 2016)*, pp. 329–338.
- Sagi, Eyal and Morteza Dehghani (2014a). “Measuring moral rhetoric in text”. In: *Social science computer review* 32 (2), pp. 132–144. DOI: 10.1177/0894439313506837.
- (2014b). “Moral rhetoric in Twitter: A case study of the US Federal Shutdown of 2013”. In: *Proceedings of the 35th Annual Meeting of the Cognitive Science Society (CogSci)*. Vol. 36, pp. 1347–1352.
- Saif, Hassan et al. (2017). “A semantic graph-based approach for radicalisation detection on social media”. In: *European semantic web conference*. Springer, pp. 571–587. DOI: 10.1007/978-3-319-58068-5_35.
- Sánchez-Rada, J. Fernando, Oscar Araque, and Carlos A. Iglesias (Feb. 2020). “Senpy: A framework for semantic sentiment and emotion analysis services”. en. In: *Knowledge-Based Systems* 190, p. 105193. ISSN: 0950-7051. DOI: 10.1016/j.knosys.2019.105193. URL: <http://www.sciencedirect.com/science/article/pii/S0950705119305313> (visited on 03/17/2020).
- Sánchez-Rada, J. Fernando, Carlos A. Iglesias, et al. (Oct. 2016). “Senpy: A Pragmatic Linked Sentiment Analysis Framework”. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. ISSN: null, pp. 735–742. DOI: 10.1109/DSAA.2016.79.
- Schnabel, Tobias et al. (2015). “Evaluation methods for unsupervised word embeddings”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 298–307. DOI: 10.18653/v1/D15-1036.
- Schwartz, H Andrew et al. (2013). “Personality, gender, and age in the language of social media: The open-vocabulary approach”. In: *PloS one* 8.9, e73791. DOI: 10.1371/journal.pone.0073791.
- Severyn, Aliaksei and Alessandro Moschitti (2015). “Twitter Sentiment Analysis with Deep Convolutional Neural Networks”. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’15. Santiago, Chile: ACM, pp. 959–962. ISBN: 978-1-4503-3621-5. DOI: 10.1145/2766462.2767830.
- Shanahan, James G, Yan Qu, and Janyce Wiebe (2006). *Computing attitude and affect in text: Theory and applications*. Vol. 20. Springer. DOI: 10.1007/1-4020-4102-0.
- Sharma, Anuj and Shubhamoy Dey (2012). “A comparative study of feature selection and machine learning techniques for sentiment analysis”. In: *Proceedings of the 2012 ACM research in applied computation symposium*, pp. 1–7. DOI: 10.1145/2401603.2401605.

- Socher, Richard et al. (2013). “Recursive deep models for semantic compositionality over a sentiment treebank”. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.
- Song, Kaisong et al. (2016). “Build Emotion Lexicon from the Mood of Crowd via Topic-Assisted Joint Non-Negative Matrix Factorization”. In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’16. Pisa, Italy: Association for Computing Machinery, pp. 773–776. ISBN: 9781450340694. DOI: 10.1145/2911451.2914759. URL: <https://doi.org/10.1145/2911451.2914759>.
- Staiano, Jacopo and Marco Guerini (2014). “Depeche Mood: a Lexicon for Emotion Analysis from Crowd Annotated News”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 427–433. DOI: 10.3115/v1/P14-2070. URL: <https://www.aclweb.org/anthology/P14-2070>.
- Stekelenburg, Jacqueliën van (2017). “Radicalization and Violent Emotions”. In: *PS: Political Science and Politics* 50.4, pp. 936–939. DOI: 10.1017/S1049096517001020.
- Stone, Philip J., Dexter C. Dunphy, and Marshall S. Smith (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT press.
- Strapparava, Carlo and Rada Mihalcea (2008). “Learning to identify emotions in text”. In: *Proceedings of the 2008 ACM symposium on Applied computing (SAC)*. Fortaleza, Ceara, Brazil, pp. 1556–1560. DOI: 10.1145/1363686.1364052.
- Strapparava, Carlo and Alessandro Valitutti (2004). “WordNet-Affect: an Affective Extension of WordNet”. In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pp. 1083–1086.
- Suarez, Daniel, Oscar Araque, and Carlos A. Iglesias (Oct. 2018). “How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining”. In: *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. ISSN: null, pp. 11–18. DOI: 10.1109/SNAMS.2018.8554488.
- Subasic, Pero and Alison Huettner (2001). “Affect analysis of text using fuzzy semantic typing”. In: *IEEE Transactions on Fuzzy systems* 9.4, pp. 483–496. DOI: 10.1109/91.940962.
- Sulis, Emilio et al. (2016). “Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not”. In: *Knowledge-Based Systems* 108, pp. 132–143. DOI: 10.1016/j.knosys.2016.05.035.
- Sun, Chi et al. (2019). “How to fine-tune BERT for text classification?” In: *China National Conference on Chinese Computational Linguistics*. Springer, pp. 194–206. DOI: 10.1007/978-3-030-32381-3_16.

- Taboada, Maite et al. (2011). “Lexicon-based methods for sentiment analysis”. In: *Computational linguistics* 37.2, pp. 267–307. DOI: 10.1162/COLI_a_00049.
- Tai, Kai Sheng, Richard Socher, and Christopher D. Manning (2015). “Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks”. In: *CoRR*. DOI: 10.3115/v1/P15-1150. arXiv: 1503.00075.
- Tang, D. et al. (Feb. 2016). “Sentiment Embeddings with Applications to Sentiment Analysis”. In: *IEEE Transactions on Knowledge and Data Engineering* 28.2, pp. 496–509. ISSN: 1041-4347. DOI: 10.1109/TKDE.2015.2489653.
- Tang, Duyu, Bing Qin, and Ting Liu (2015). “Document modeling with gated recurrent neural network for sentiment classification”. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1422–1432. DOI: 10.18653/v1/D15-1167.
- Tausczik, Yla R and James W Pennebaker (2010). “The psychological meaning of words: LIWC and computerized text analysis methods”. In: *Journal of language and social psychology* 29.1, pp. 24–54. DOI: 10.1177/0261927X09351676.
- Teernstra, Livia et al. (2016). “The morality machine: tracking moral values in Tweets”. In: *Proceedings of the 15th International Symposium on Intelligent Data Analysis (IDA)*. Stockholm, Sweden, pp. 26–37. DOI: 10.1007/978-3-319-46349-0_3.
- Thelwall, Mike, Kevan Buckley, and Georgios Paltoglou (2012). “Sentiment strength detection for the social web”. In: *Journal of the American Society for Information Science and Technology* 63.1, pp. 163–173. DOI: 10.1002/asi.21662.
- Turney, Peter D (2002). “Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews”. In: *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, pp. 417–424. DOI: 10.3115/1073083.1073153.
- Turney, Peter D, Patrick Pantel, et al. (2010). “From frequency to meaning: Vector space models of semantics”. In: *Journal of artificial intelligence research* 37.1, pp. 141–188. DOI: 10.1613/jair.2934.
- Vaswani, Ashish et al. (2017). “Attention is all you need”. In: *Advances in neural information processing systems*, pp. 5998–6008.
- Vecina, Maria L, Daniela Marzana, and Mariola Paruzel-Czachura (2015). “Connections between moral psychology and intimate partner violence: Can IPV be read through moral psychology?” In: *Aggression and violent behavior* 22, pp. 120–127. DOI: 10.1016/j.avb.2015.04.013.
- Vergani, Matteo and Ana-Maria Bliuc (2015). “The evolution of the ISIS language: a quantitative analysis of the language of the first year of Dabiq magazine”. In: *Sicurezza, Terrorismo e Società = Security, Terrorism and Society* 2.2, pp. 7–20.

- Villena-Román, Julio et al. (2015). “Overview of TASS 2015”. In: *Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015*. Ed. by Julio Villena-Román et al. Vol. 1397. CEUR Workshop Proceedings. CEUR-WS.org, pp. 13–21. URL: <http://ceur-ws.org/Vol-1397/overview.pdf>.
- Vulcu, Gabriela et al. (2014). “Generating linked-data based domain-specific sentiment lexicons from legacy language and semantic resources”. In: *5th International Workshop on EMOTION, SOCIAL SIGNALS, SENTIMENT and LINKED OPEN DATA*.
- Wang, Sida and Christopher D Manning (2012). “Baselines and bigrams: Simple, good sentiment and topic classification”. In: *Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2*. Association for Computational Linguistics, pp. 90–94.
- Warriner, Amy Beth, Victor Kuperman, and Marc Brysbaert (2013). “Norms of valence, arousal, and dominance for 13,915 English lemmas”. In: *Behavior research methods* 45.4, pp. 1191–1207. DOI: 10.3758/s13428-012-0314-x.
- Wehrmann, J. et al. (May 2017). “A character-based convolutional neural network for language-agnostic Twitter sentiment analysis”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2384–2391. DOI: 10.1109/IJCNN.2017.7966145.
- Wiedemann, Gregor (2013). “Opening up to big data: Computer-assisted analysis of textual data in social sciences”. In: *Historical Social Research/Historische Sozialforschung*, pp. 332–357.
- Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann (2009). “Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis”. In: *Computational linguistics* 35.3, pp. 399–433. DOI: 10.1162/coli.08-012-R1-06-90.
- Wu, Hui-Hsin et al. (2013). “Building a Graded Chinese Sentiment Dictionary Based on Commonsense Knowledge for Sentiment Analysis of Song Lyrics”. In: *J. Inf. Sci. Eng.* 29.4, pp. 647–662.
- Xia, Rui and Chengqing Zong (2010). “Exploring the use of word relation features for sentiment classification”. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, pp. 1336–1344.
- Yadollahi, Ali, Ameneh Gholipour Shahraki, and Osmar R. Zaiane (2017). “Current State of Text Sentiment Analysis from Opinion to Emotion Mining”. In: *ACM Comput. Surv.* 50.2, 25:1–25:33. ISSN: 0360-0300. DOI: 10.1145/3057270.
- Yang, Zichao et al. (2016). “Hierarchical attention networks for document classification”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489. DOI: 10.18653/v1/N16-1174.

- Yarkoni, Tal (2010). “Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers”. In: *Journal of Research in Personality* 44.3, pp. 363–373. DOI: 10.1016/j.jrp.2010.04.001.
- Young, Tom et al. (2018). “Recent trends in deep learning based natural language processing”. In: *ieee Computational intelligence magazine* 13.3, pp. 55–75.
- Yu, Liang-Chih et al. (2017). “Refining word embeddings for sentiment analysis”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 534–539. DOI: 10.18653/v1/D17-1056.
- Zhang, Dongwen et al. (2015). “Chinese comments sentiment classification based on word2vec and SVMperf”. In: *Expert Systems with Applications* 42.4, pp. 1857–1863. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2014.09.011.
- Zhu, Feng and Xiaoquan Zhang (2010). “Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics”. In: *Journal of marketing* 74.2, pp. 133–148. DOI: 10.1509/jmkg.74.2.133.
- Zhu, G. and C. A. Iglesias (Jan. 2017). “Computing Semantic Similarity of Concepts in Knowledge Graphs”. In: *IEEE Transactions on Knowledge and Data Engineering* 29.1, pp. 72–85. ISSN: 2326-3865. DOI: 10.1109/TKDE.2016.2610428.
- Zirn, Cécilia et al. (2011). “Fine-grained sentiment analysis with structural features”. In: *Proceedings of 5th International Joint Conference on Natural Language Processing*, pp. 336–344.

List of Figures

1.1	Overview of the research fields where this thesis has made contributions. Core contributions are delineated stronger.	10
4.1	Summary of the contributions of this thesis, organized by objectives.	108
4.2	Graphical schema of how domain centrality is computed.	115
4.3	Graphical representation of polarity distribution of the modified lexicon (a) before the transformation process, and (b) after the transformation.	117
4.4	Conceptual diagram of word projection over a lexicon formed only by the set of words <i>{good, bad}</i>	120
4.5	SIMON similarity computation: similarities are computed against a selection of lexicon words (in green and red), and a <i>max</i> function is applied column-wise, obtaining a feature vector.	121
4.6	SIMON system architecture diagram, when combined with embedding text representations.	123
4.7	Architecture of the radicalization detection model, that combines a novel emotion representation with the SIMON adapted to this domain.	127
4.8	Overview of the process, from dictionary expansion to moral value prediction.	128
4.9	Diagram of the proposed feature extraction approaches that utilize the presented MoralStrength lexicon.	129

List of Tables

1.1	Overview of related Sentiment and Emotion lexica.	13
1.2	Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.	20
1.2	Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.	21
1.2	Summary of Literature Survey regarding the use of affective technologies for processing online radicalization signals. Types are A: Analysis, D: Detection, P: Prediction.	22
2.1	Confusion matrix for a two-class scenario.	33
3.1	Publications that are directly related to this thesis.	38
4.1	Publications related to Objective 1	110
4.2	Publications related to Objective 2	114
4.3	Publications related to Objective 3	119
4.4	Publications related to Objective 4	125
4.5	Publications not directly related to the core of this thesis.	130
4.6	Overview of code, resources, and utilities of this thesis.	132
A.1	Publications not directly related to the core of the thesis.	165

Glossary

SVM Support Vector Machines

CNN Convolutional Neural Network

PoS Part-of-Speech

NLP Natural Language Processing

SA Sentiment Analysis

EA Emotion Analysis

BOW Bag Of Words

MLP Multi-Layer Perceptron

RNN Recurrent Neural Networks

LSTM Long Short-Term Memory

RNTN Recursive Neural Tensor Network

CBOW Continuous Bag of Words

LEA Law Enforcement Agencies

MFD Moral Foundations Dictionary

MFT Moral Foundations Theory

LIWC Linguistic Inquiry and Word Count

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

DDR Distributed Dictionary Representations

BERT Bidirectional Encoder Representations from Transformers

ELMo Embeddings from Language Models

SIMON SIMilarity-based sentiment projectiON

SEDLex SEntiment adapted to the Domain Lexicon

Publications

A.1 Summary of all publications

The following tables summarizes all the publications of this thesis. The following color scheme has been used: gray for conference papers, and blue for journal papers.

Table A.1: Publications not directly related to the core of the thesis.

Page	Title	Year	Venue	Ranking
167	Radical Text Detection based on Stylometry	2020	Proceedings of the 6th International Conference on Information Systems Security and Privacy	
176	Senpy: A Framework for Semantic Sentiment and Emotion Analysis Services	2020	Knowledge-Based Systems	JCR 2019 Q1 (5.921)
38	An Approach for Radicalization Detection Based on Emotion Signals and Semantic Similarity	2020	IEEE Access	JCR 2019 Q1 (3.745)

Table A.1: Publications not directly related to the core of the thesis.

55	MoralStrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction	2019	Knowledge-Based Systems	JCR 2019 Q1 (5.921)
180	GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter	2019	Proceedings of the 13th International Workshop on Semantic Evaluation	
82	DepecheMood++: a Bilingual Emotion Lexicon Built Through Simple Yet Powerful Techniques	2019	IEEE Transactions on Affective Computing	JCR 2019 Q1 (7.512)
67	A Semantic Similarity-based Perspective of Affect Lexicons for Sentiment Analysis	2019	Knowledge-Based Systems	JCR 2019 Q1 (5.921)
189	How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining	2018	2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)	
198	Neural Domain Adaptation of Sentiment Lexicons	2017	2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACHIW)	
205	Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets	2017	Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN	
95	Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications	2017	Expert Systems with Applications	JCR 2019 Q1 (5.452)
212	Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis	2016	2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)	
221	Senpy: A Pragmatic Linked Sentiment Analysis Framework	2016	2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)	

Table A.1: Publications not directly related to the core of the thesis.

230	Aspect Based Sentiment Analysis of Spanish Tweets	2015	Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015
-----	---	------	---

A.2 Rest of publications

The following tables describe the rest of the publications that are related to the thesis, and have not been shown in Chapter 3. The following color scheme has been used: gray for

conference papers, and blue for journal papers.

A.2.1 Radical Text Detection based on Stylometry

Title	Radical Text Detection based on Stylometry
Authors	de Pablo, Álvaro and Araque, Oscar and Iglesias, Carlos A.
Proceedings	Proceedings of the 6th International Conference on Information Systems Security and Privacy
ISBN	978-989-758-399-5
Year	2020
Keywords	Radicalism, Terrorism, ISIS, Dabiq, Rumiyah, Stylometry, Machine Learning
Pages	524–531
DOI	https://doi.org/10.5220/0008971205240531
Abstract	The Internet has become an effective tool for terrorist and radical groups to spread their propaganda. One of the current problems is to detect these radical messages in order to block them or promote counter-narratives. In this work, we propose the use of stylometric methods for characterizing radical messages. We have used a machine learning approach to classify radical texts based on a corpus of news from radical sources such as the so-called ISIS online magazines Dabiq and Rumiyah, as well as news from general newspapers. The results show that stylometric features are effective for radical text classification.

Radical Text Detection based on Stylometry

Álvaro de Pablo^a, Óscar Araque^b and Carlos A. Iglesias^c

Intelligent Systems Group, Universidad Politécnica de Madrid, Madrid, Spain
alvaro.depablo@alumnos.upm.es, {o.araque, carlosangel.iglesias}@upm.es

Keywords: Radicalism, Terrorism, ISIS, Dabiq, Rumiya, Stylometry, Machine Learning.

Abstract: The Internet has become an effective tool for terrorist and radical groups to spread their propaganda. One of the current problems is to detect these radical messages in order to block them or promote counter-narratives. In this work, we propose the use of stylometric methods for characterizing radical messages. We have used a machine learning approach to classify radical texts based on a corpus of news from radical sources such as the so-called ISIS online magazines Dabiq and Rumiya, as well as news from general newspapers. The results show that stylometric features are effective for radical text classification.

1 INTRODUCTION

Recent terrorist attacks have revealed a vulnerability in modern society. As a natural consequence, countering and preventing violent terrorism and radicalization has become a major international security priority. Terrorism is nowadays a major threat to security in the EU Member States (Europol, 2016).

In this context, cyberspace is of special interest, since the effective usage of the Internet has been one of the differential aspects of modern terrorist groups. Their use range from psychological warfare, publicity, and propaganda to more instrumental actions such as recruitment, mobilization, fundraising, information sharing, coordination, and networking (Weimann, 2004).

Terrorist organizations have used different media strategies. According to New York Times (Shane and Hubbard, 2014), Islamic State of Iraq and Syria (ISIS) media strategy can be named Jihad 3.0 since they follow a sophisticated multidimensional strategy (Al-Rawi, 2018; Feakin and Wilkinson, 2015), including not only social media (e.g., Twitter) but also video games (Al-Rawi, 2018), online magazines (e.g., Inspire, Dabiq and Rumiya), high-quality videos, audio reports in SoundCloud or publication of battle summaries in JustPaste, to name a few.

Since understanding terrorist organizations is a requirement for detecting and countering them, a large

body of research has studied radical presence in social media (Correa and Sureka, 2013; Fernandez et al., 2018), including Twitter, YouTube and Facebook, online magazines (Ingram, 2017) and videogames (Al-Rawi, 2018).

In this work, we provide a first attempt to characterize the writing style of terrorist groups in online magazines. This has several important applications, such as the detection (Cohen et al., 2018) of online terrorist propaganda and misinformation (Wilner, 2018) as well as the authorship attribution for counter-terrorism purposes (Kijewski et al., 2016).

To address this research objective, we have used a rich source of data, the official ISIS online magazines Dabiq and Rumiya. The methodology followed has been to create a corpus of radical texts with the articles of these magazines. In addition, we have mined articles related to radicalism from the online magazine of the Arab satellite news network Al Jazeera, which is considered an “alternative” medium (Iskandar, 2006; Al-Sadi, 2012). Another corpus of non-radical nor alternative articles has been created with articles from international magazines (e.g., New York Times, CNN) related to radical events. Then, a stylometric analysis of the articles has been carried out, and a machine-learning classifier has been trained with stylometric features to evaluate its relevance.

The remainder of this paper is organized as follows. Sect. 2 reviews related work. Sect. 3 provides a background on the style metrics used in the analysis. Sect. 4 presents the methodology followed for collecting the dataset as well as a stylometric analysis of non-radical and radical news. Sect. 5 reports

^a  <https://orcid.org/0000-0002-7561-9997>

^b  <https://orcid.org/0000-0003-3224-0001>

^c  <https://orcid.org/0000-0002-1755-2712>

the experiments to classify texts based on stylometric features. Traditional approaches are compared to evaluate the increment in performance of using these features. Finally, Sect. 6 summarizes the results from our analysis and highlights the implications of our results.

2 RELATED WORK

Stylometry techniques¹ have been applied to authorship attribution, authorship verification, authorship profiling, stylochronometry, and adversarial stylometry (Neal et al., 2017).

In the context of radicalism detection, several works have already used stylometric features.

The approach most closely related to our approach is that of Ashcroft et al. (Ashcroft et al., 2015). They describe a machine learning system for classifying Jihadist messages on Twitter using stylometric, temporal, and sentiment features. They use the stylometric features described in (Narayanan et al., 2012), which include i) length of words/characters, vocabulary richness (based on Yule’s K^3 and frequency of *hapax legomena*, *dis legomena*, etc.), word shape (frequency of words with different combinations of upper and lower case letters), word length (frequency of words that have 1–20 characters), frequency of different elements (words, letters, digits, punctuation, hashtags, special characters, function words, and syntactic category pairs). Nevertheless, this study does not detail the impact of stylometric features on the classification. In addition, most stylometric metrics are based on word and hashtag frequencies.

Mencarini and Sensidoni (Mencarini and Sensidoni, 2017) analyzed some terrorist or cybercriminal statements belonging to groups like Anonymous, ISIS, Al Qaeda, or the Muslim Brotherhood with some style metrics which include readability, vocabulary, registers and slangs, grammatical tenses and more. The application of the stylometry to these kinds of texts showed differences between different terrorist groups in aspects like the difficulty of reading or lexical diversity.

Regarding the analysis of radical online magazines from ISIS, Dabiq, and Rumiya, as well as al Qaeda’s Inspire, they have been subject to a number of studies.

Bisgin et al. (Bisgin et al., 2019) analyze Dabiq’s propagandist elements by studying the entities mentioned in the articles.

¹The interested reader in stylometry techniques and applications should consult (Neal et al., 2017) for a recent survey of this topic.

Vergani and Bliuc (Vergani and Bliuc, 2018) have analyzed the language differences between ISIS and al-Qaeda in their journals Dabiq and Inspire, respectively. Their analysis is based on the tools Linguistic Inquiry and Word Count (LIWC) (Tausczik and Pennebaker, 2010) and Recursive Inspection of Text (RIOT) (Boyd, 2015) to calculate the frequency of texts in language categories (e.g., article, pronoun), psychological dimensions (e.g., affect, emotions, cognitive mechanisms) or moral foundations (e.g., harm, fairness, authority). The main insight of this study is that ISIS propaganda is more effective in mobilizing individuals who are more authoritarian and more religious than that of al-Qaeda.

Sikos et al. (Sikos et al., 2014) analyze an Arabic translation of al-Qaeda’s Inspire magazine since most of the magazines are published in English together with an Arabic translation. They use stylometry features using LIWC to analyze authorship of Inspire’s issues, concluding that the issues 1-9 were produced by one group of editors while a different group of editors produced issues 10 and 11.

Wignell et al. (Wignell et al., 2017) analyze Dabiq and Rumiya’s style in terms of the topics addressed and the images included, using a mixed-methods approach. They conclude that the distribution of topics and images is quite consistent in both magazines. In addition, they always include ISIS’s core values, intolerance and a hostile world.

Johnston and Weiss (Johnston and Weiss, 2017) develop a deep learning model for detecting Sunni propaganda. They use a dataset of Sunni propaganda with the journals Dabiq, Rumiya and Inspire as curated from the Clarion project², and another benign dataset that includes Wikipedia articles and news. In this case, their approach is similar to ours for building the dataset, but they do not address stylometry.

3 ON STYLOMETRY

In this section, the main metrics used for the stylometric analysis of the texts are presented.

Readability Index. (Kincaid et al., 1975) is a style metric that measures how easy or difficult reading a text is. We have used two popular metrics for texts written in English, Fog Count, and Flesch (Kincaid et al., 1975), both based on the number of syllables of the words within the text. These metrics relate the value obtained with the US Grade Level required to read a text.

²<https://clarionproject.org/>

Vocabulary Richness. Measures the lexical diversity of the texts. The most relevant metric is Type Token Ratio (TTR) (Šišková, 2012), which is calculated as shown in Eq. 1.

$$TTR = \frac{\text{types}}{\text{tokens}} \quad (1)$$

The main problem of TTR is that the longer a text is, the smaller TTR. To solve this, the metrics MTLT and HD-D have been proposed, as described below.

The **Measure of Textual Lexical Diversity (MTLD)** (McCarthy and Jarvis, 2010) is a sequential algorithm for measuring the Vocabulary Richness of a text. It is calculated as follows:

$$MTLD = \frac{\text{tokens}}{\text{Segments} + \text{Partial segment}} \quad (2)$$

where *Segments* is the number of segments of the text that have a TTR lower than a limit (usually 0.72), and *Partial Segment* is the last segment of the text.

Hypergeometric Distribution of the Diversity index (HD-D) (McCarthy and Jarvis, 2010). It uses the hypergeometric distribution to determine the probability of a particular word in a text occurring at least once in a random sample of a particular length. The probability is calculated as follows:

$$P(X = x) = \frac{\binom{d}{x} \binom{N-d}{n-x}}{\binom{N}{n}} \quad (3)$$

where N is the population size, n is the sample size, d is the number of the elements belonging to the request type and x is the number of elements in the sample that belongs to the request category.

HD-D algorithm is shown below.

$$HDD = \sum_{i=0}^n \frac{1}{42} * P(X = \text{type}_i) \quad (4)$$

where $P(X = \text{type}_i)$ is the probability given by the hypergeometric distribution for any type in the text.

Formality Metrics. (Heylighen and Dewaele, 1999) measure the degree of formality of a text. The main metrics are Adjective Score and F Score.

Adjective Score (Fang and Cao, 2009) analyzes the adjective density of a text. Its value is usually under 10%. The higher the Adjective Score, the higher the degree of formality. Its calculation is as follows:

$$\text{Adj Score} = \frac{\text{Number of Adjectives}}{\text{tokens}} * 100 \quad (5)$$

F Score (Heylighen and Dewaele, 1999) is another formality metric. It takes into account the POS

(Part Of Speech) tagging processes. The higher the F Score, the higher the degree of formality.

$$F = \frac{\frac{N + \text{Adj} + \text{Prep} + \text{Dt} - \text{Pro} - \text{Vb} - \text{Adv} - \text{In}}{\text{Tokens}} * 100 + 100}{2} \quad (6)$$

where N is *Nouns*, *Adj* is *Adjectives*, *Prep* is *Prepositions*, *Dt* is *Determiners*, *Pron* is *Pronouns*, *Vb* is *Verbs*, *Adv* is *Adverbs* and *In* is *Interjections*.

Coherence Measure. (Foltz et al., 1998) is an index that evaluates the coherence of a text. The Coherence index of a text is calculated by adding the semantic similarity of every sentence and the successive one, as shown in Eq. 7.

$$\text{Coherence} = \frac{\sum_{i=0}^{n-1} \text{Coh}(\text{sent}_i, \text{sent}_{i+1})}{\text{sentences}} * 100 \quad (7)$$

where *sentences* is the number of sentences within the text, sent_i is the sentence number i within the text and $\text{Coh}(\text{sent}_i, \text{sent}_{i+1})$ is the semantic similarity of a sentence with the following in the text.

4 ANALYSIS OF THE DATA

4.1 Dataset Collection

The field of radicalization detection has an inherent need for representative and reliable data with which it is possible to develop computational models. Nevertheless, annotated data is scarce, which makes it difficult for some research advancements. Additionally, many of the existing datasets have been extracted from Twitter (Fernandez et al., 2018). While Twitter is indeed a representative source of data, it also has some generalization issues that proper radicalization detection studies should take into account. In this way, intending to explore alternative data sources, we have collected a dataset based on magazines.

Following previous work that tackles the use of Dabiq for characterizing radical language (Nouh et al., 2019; Bisgin et al., 2019), we have considered both the Dabiq and Rumiyah magazines, which are used by ISIS radical organization. Original publications have been collected from an online resource dedicated to radicalization online³. The obtained dataset consists of the 15 issues of Dabiq and the 13 issues of Rumiyah (276 texts), and 349 articles from Al Jazeera, which can be considered as an alternative source (Iskandar, 2006; Al-Sadi, 2012).

³<http://www.jihadology.net>

Table 1: Statistics of the collected dataset.

	CNN	NYT	Al Jazeera	Dabiq	Rumiyah
Avg. no. of words/article	692.565	197.501	680.169	969.823	612.286
Avg. no. of sentences/article	31.025	9.04	28.263	43.0	41.720
Avg. no. of word appearances/article					
<i>Allah</i>	0.005%	0.001%	0.004%	1.799%	2.628%
<i>Jihad</i>	0.012%	0.012%	0.013%	0.318%	0.257%
<i>Khilafah/Caliphate</i>	0.037%	0.038%	0.033%	0.314%	0.325%
<i>Iran</i>	0.228%	0.221%	0.228%	0.029%	0.013%

As a balance of the previous radical data, we have selected two online newspapers that address ISIS-related issues using a neutral tone: CNN and The New York Times. These newspapers are freely accessible and can be obtained through their APIs. In total, 383 articles have been collected from CNN, and 765 from The New York Times. For the dataset compensation, 300 random texts from the CNN dataset and 300 random texts from The New York Times have been analyzed. The same preprocessing has been applied to all the obtained data: normalization of capital letters, numbering, and contraction tokenization (e.g., *I've*, *we'll*). The processed data contains documents with lower case tokens. In Table 1, some statistics of the obtained dataset are shown.

4.2 Analysis

This section focuses on the analysis of the dataset described previously. We follow a systematic approach to analyze different stylometry metrics for every issue of the magazines Dabiq and Rumiyah and provide a comparison with the average metrics of the news dataset. From now on, *D.n* refers to “*Dabiq: Issue n*”, while *R.n* refers to “*Rumiyah: Issue n*”.

4.2.1 Formality: Adjective Score

The Adjective Score remains constant in Rumiyah and Dabiq issues, although it is slightly slower in Rumiyah. The reason for this could be the differences between Rumiyah and Dabiq. Some authors have reported that Rumiyah replaced Dabiq because of a shift in ISIS’s priorities, and it is more focused on directly attacking western countries, rather than encouraging recruits to migrate to Iraq and Syria (Comerford, 2016). Other authors (McKernan, 2016) also report that Rumiyah can be considered an inferior product based on recycled material and significantly shorter.

It can be highlighted the difference between propaganda news and standard news, being the formality of standards news significantly higher (around 7%). In conclusion, propaganda texts are less formal than standard news. Another finding of this experiment is that Dabiq is more formal than Rumiyah.

These results are aligned with the conclusions of the authors of the adjective score, Fang and Cao (Fang and Cao, 2009). They concluded that the variations of adjectives seem to be a reliable indicator to categorize text categories. This is confirmed for our experiment for the two text categories, radical propaganda, and news.

4.2.2 Formality: F Score

In contrast with the previous analysis, the F Score shows a very stable value across all the issues. In addition, the F Score of news (72.49%) is slightly higher, and only one issue achieves this value (D.2, with 74.69%).

Based on the F Score, values above 70% imply high formality. Thus, the analyzed news are very formal, and the vast majority of the terrorist texts have a standard formality grade. According to (Heylighen and Dewaele, 1999), a formal style is characterized by objectivity and cognitive load, while an informal style is more direct, subjective, less accurate, and less informative. This confirms our results, where subjective propaganda texts have a lower formality index.

4.2.3 Coherence

For measuring coherence, a Word Embeddings model has been used (Sun et al., 2016) for English texts. The average of the coherence of the news is around 60.77% and only two issues (D.5 and D.7) are far from that value. The higher value is the coherence of R.12, with 74.41%.

The results show that Dabiq is less coherent than Rumiyah. Specifically, the average of the coherence

of the issues of Dabiq is 64.33% (1 percentage point over the coherence of the news), and the average of Rumiya is 69.84% (almost seven percentage points higher). Moreover, every issue of Rumiya exceeds the coherence value of the news, but some of the issues of Dabiq do not exceed it.

Thus, almost all issues (except D.5, with 51.27% and D.7, with 59.98%) are above 60% of coherence. In fact, most of the values are above 70%.

Therefore, it can be said that both magazines and news are coherent, being the coherence of Rumiya higher than the coherence of Dabiq. Also, ISIS magazines are a bit more coherent than news. In fact, every issue of Rumiya is more coherent than newspapers, while there are few issues of Dabiq that are less coherent than newspapers.

4.2.4 Vocabulary Richness

Two different metrics have been used to measure the lexical diversity of the texts.

Firstly, the average of the application of the HD-D metric to the news texts is 74.21%, while the average of Dabiq and Rumiya texts is around 71.70%, close to news average but below. Moreover, while the HD-D of Dabiq (with an average of 72.93%) is closer to the HD-D of the news in each of its issues (in fact, D.3 surpasses it), the HD-D of Rumiya (with an average of 70.28 %) is always under this limit, and it does not get that close. This finding is aligned with some observations about the lower quality of Rumiya, as commented previously. Even so, all the HD-D values are around 70%, which implies a high lexical diversity.

Regarding the metrics MTLD, it provides better discrimination between news and propaganda, since its value for news is around 92% while its average value for propaganda is around 67%. In addition, its value is lower in Rumiya than in Dabiq.

In conclusion, both news and magazines have good lexical diversity, but news values are higher in this aspect. Moreover, Dabiq has a better vocabulary richness than Rumiya. These conclusions are supported by both metrics, although MTLD is more discriminative.

4.2.5 Readability Index

Two different metrics have been used too for measuring the Readability Index of the texts.

The results of the magazines vary greatly throughout the issues, and all the metrics applied to the magazines follow the same path, having approximately the maximums and the minimums in the same issues.

As for news, the Fog Count is 14.27, and Flesch-Kincaid is 13.05, which implies in both cases that news require a college level of education (Kincaid et al., 1975).

The Flesch results of the magazines have the lowest values. In Dabiq's issues, only D.11 exceeds the Flesch of the news. On the other hand, most of Rumiya issues exceed this value. In fact, most of the Dabiq values are between 8 and 12, and only a High School Grade Level is required for reading them. Nevertheless, Rumiya values are higher than 12, and a College Grade Level is needed.

Fog Count results show similar values but higher than Flesch. In fact, a College Grade Level is required in almost all issues.

Moreover, for both metrics, Dabiq is easier to read than Rumiya. Also, most Flesch values are under the values of the news, but in the Fog Count measure, there are more values above the values of the news.

4.2.6 Use of Words

Also, it has been measured how many common and uncommon words are used in the different analyzed texts. Please note that common words are those who are common in the original language (in this case, English). If a word is common or uncommon is given by an algorithm based on the Zipf Law (Montemurro, 2001).

The results show that terrorism news have the highest proportion of common words (98.32%). The common words of Rumiya and Dabiq are 92.49% and 92.83%, respectively. On average, terrorist texts have 92.66%, more than five percentage points under news texts values.

The most frequent uncommon words of the news texts (1.68%) are words like "*Poway*", "*Haftar*" or "*Yisroel*". In the case of the two magazines, the most frequent uncommon words (7.34%) are "*Mujahidin*", "*Khilafah*" and "*Kufr*".

5 EXPERIMENTATION

In order to assess the effectiveness of the style features in radicalization detection, we have designed an experimental study that makes use of a machine learning model. In this way, we postulate the problem as a classification task that learns from the provided features. We quantify the performance using style features and compare such results over a baseline, which we define with bigrams. Following, we experiment with the fusion of both types of features, with the aim of complementing a typical textual representation (bi-

grams) with style-specific features. For delimiting the problem, we have cross-validated the number of bigram tokens, considering both the final accuracy of the method, as well as the time needed to train the models. With these constraints, we set the bigram approach to use the 100 most-common tokens.

Regarding the importance analysis of style metrics for radicalization detection, we measured the correlation between each style metric with the annotation of the documents (neutral or radical). The obtained results can be seen in Figure 1.

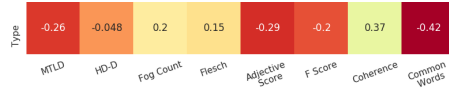


Figure 1: Correlation between Style Metrics and Type of texts.

As can be seen, the style metrics with higher correlations to the text annotations are MTL (Vocabulary Richness), Common Words, F Score, Adjective Score (Formality), and Coherence. It is worth to note that in the field of Readability Index, the computed correlations are not large. Even so, the Fog Count metric stands out over the Flesch metric, for which it has been included as a feature. Furthermore, Common Words can be stressed over the rest of the metrics in terms of the obtained correlation. This may indicate that such a metric represents a useful feature for radicalization detection.

In summary, the style metrics selected as features by the classifier were Fog Count, Adjective Score, Common Words, F Score, Coherence, and MTL. It should be noted that these results fit well with what was stated in the previous section.

Table 2 shows the results obtained in the experiments. We evaluate the performance of bigrams and style metrics, as well as their combination. As a learning algorithm, we evaluate a popular model: Random Forest. We report the accuracy, recall, precision and f-score, averaged through cross-validation in a 5-fold fashion.

Table 2: Classifier results with Random Forest algorithm.

Random Forest Classifier			
Features	Style	Bigrams	Bigrams + Style
Accuracy	78.3	89.6	92.2
Precision	78.5	90.1	92.8
Recall	78.2	89.6	92.3
F-Score	78.3	89.6	92.3

As seen in the evaluation, Random Forest learner improves its f-score over the bigrams baseline when

combines with style features. It is noteworthy that the performance obtained when using uniquely bigrams is higher than using style metrics uniquely. This is to be expected, as the bigrams features are more expressive than style features when the aim is to represent text. Also, we need to consider that the dimensionality of the bigrams representations (100) is much higher than the style features, which amount to a selection of 6.

Considering the improvement with the combination of bigrams and style features, it can be seen that the obtained improvement is 2.7% using the Random Forest, reaching 92.3%. This positive result indicates the effectiveness of the proposed style features when used with a standard text representation approach, obtaining a much more robust classification system.

With all this, we can conclude that the style metrics presented in this paper can result in a benefit for machine learning models applied to the task of radicalization detection.

6 DISCUSSION

This paper presents a feature extraction method for detecting radical content in texts. As part of this, a comprehensive analysis of the presented texts' stylometry is done. Both radical magazines and international news are compared, drawing some interesting differences among them.

In the first place, it has been seen that the captured radical texts magazines are formal, characteristic that is shared with the international news. Even so, considering the two different formality metrics applied, the neutral news achieve higher formality scores than their radical equivalents. Interestingly enough, Dabiq issues tend to have more formal texts than the Rumiyaah ones, regardless of the metric used for the study.

Attending to the coherence, both radical and neutral texts have obtained a score that lies in the 60%-70% range, indicating their coherence. Still, some differences can be observed when comparing radical and neutral texts. This is clearer when looking at the correlation with the annotation obtained by this style metric, which amounts to 0.35.

Continuing, both of the used metrics of Vocabulary Richness show that news have more lexical diversity than the terrorist texts analyzed. Again, such differences can explain the obtained correlations in Figure 1. As for the Readability Index, used metrics show that in many times terrorist texts are, in general, more comfortable to read than news texts. Consequently, we can conclude that the readability of the

texts can also offer information over the distinction between radical and non-radical content.

Lastly, the analysis of the use of common and uncommon words has interesting results. Non-radical news have a more significant proportion of common words in their texts in comparison to radical texts. In addition, the analyzed magazines have an equivalent proportion of these words, widely surpassed by news texts.

When performing a comparison between Dabiq and Rumiyah, we have found that the first one is more formal, has a larger lexical diversity, and is easier to read. Alternatively, Rumiyah has a higher coherence score.

7 CONCLUSIONS

The field of radicalization detection is nowadays facing multiple challenges. Regarding the available data, the scarcity of resources is relevant since limited datasets do not allow to develop computational models. In this sense, it is essential to gather more resources that represent reliable and robust sources of data. In light of this, this work includes the extraction of a dataset that is posteriorly used in the evaluation. This data combines radical content, coming from sources like Dabiq, Rumiyah, and Al Jazeera, as well as neutral content that has been extracted from international news outlets, CNN and The New York Times.

Considering the existing radicalization detection models, the relevant information for detecting radical content has not been exhaustively studied. While there are proposals that aim at exploiting different sources of information to detect radicalization (Saif et al., 2017), the application of the stylometry field in radicalization detection has not been thoroughly studied. This paper proposes the use of several stylometry metrics with the aim of characterizing radical and non-radical texts using this kind of information. To the extent of our knowledge, such an approach is novel in the field of radicalization detection.

Attending to the results obtained in the evaluation, it is safe to assume that by themselves, the style features can obtain fairly reasonable performances in this task. When compared to a more detailed representation of a text (e.g., bigrams), as expected, the mentioned style features perform poorly. This is due to their low dimensionality and, consequently, their lower representative power in comparison to an established technique such as bigrams. Nevertheless, as mentioned above, the trend of the field lies in combining several sources of knowledge in order to obtain

better, more robust representations on top of which learning algorithms can obtain better results. Thus, when evaluating the combination of these two types of features, we find that the overall performance is improved. These results indicate that the proposed features can enhance the performance of a classification system, complementing basic textual representations.

We believe that exploiting stylometric cues from the text is indeed an interesting research path. Previous work has also studied the viability of this kind of information source (Sikos et al., 2014).

As future work, we would like to focus on evaluating the style features utility on other domains, like radical content on social networks. To do so, several methods for extracting such metrics need to improve, since many texts in social networks are not as long as in magazines. This may difficult a successful analysis of style. Additionally, we would like to expand the textual representations by incorporating recent Natural Language Processing techniques such as word embeddings (Mikolov et al., 2013; Araque et al., 2017) and language models (Devlin et al., 2018), adapting them to the domain of radicalization detection.

ACKNOWLEDGEMENTS

This work has been supported by the H2020 project Trivalent, grant agreement 740934, under the call SEC-06-FCT-2016.

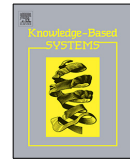
REFERENCES

- Al-Rawi, A. (2018). Video games, terrorism, and isis's jihad 3.0. *Terrorism and Political Violence*, 30(4):740–760.
- Al-Sadi, M. R. (2012). Al Jazeera television: Rhetoric of deflection. *Arab Media & Society*, 15.
- Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., and Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246.
- Ashcroft, M., Fisher, A., Kaati, L., Omer, E., and Prucha, N. (2015). Detecting jihadist messages on twitter. In *2015 European Intelligence and Security Informatics Conference*, pages 161–164. IEEE.
- Bisgin, H., Arslan, H., and Korkmaz, Y. (2019). Analyzing the dabiq magazine: The language and the propaganda structure of isis. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 1–11. Springer.
- Boyd, R. (2015). Recursive inspection of text scanner [internet].

- Cohen, S. J., Kruglanski, A., Gelfand, M. J., Webber, D., and Gunaratna, R. (2018). Al-qaeda's propaganda decoded: A psycholinguistic system for detecting variations in terrorism ideology. *Terrorism and political violence*, 30(1):142–171.
- Comerford, M. (2016). What is lost in dabiQ. *New Statesman*.
- Correa, D. and Sureka, A. (2013). Solutions to detect and analyze online radicalization: a survey. *arXiv preprint arXiv:1301.4916*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Europol, T. (2016). European union terrorism situation and trend report 2016. *Europol*.
- Fang, A. C. and Cao, J. (2009). Adjective density as a text formality characteristic for automatic text classification: A study based on the british national corpus. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*, volume 1.
- Feakin, T. and Wilkinson, B. (2015). *The Future of Jihad: What Next for ISIL and Al-Qaeda?* Australian Strategic Policy Institute.
- Fernandez, M., Asif, M., and Alani, H. (2018). Understanding the roots of radicalisation on twitter. In *Proceedings of the 10th ACM Conference on Web Science*, pages 1–10. ACM.
- Foltz, P. W., Kintsch, W., and Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. *Discourse processes*, 25(2-3):285–307.
- Heylighen, F. and Dewaele, J.-M. (1999). Formality of language: definition, measurement and behavioral determinants. *Interne Bericht, Center "Leo Apostel", Vrije Universiteit Brussel*.
- Ingram, H. J. (2017). An analysis of inspire and dabiQ: Lessons from aqap and islamic state's propaganda war. *Studies in Conflict & Terrorism*, 40(5):357–375.
- Iskandar, A. (2006). Is al Jazeera alternative? *The Real (Arab) World: Is Reality TV Democratizing the Middle East?*, 1(2):249.
- Johnston, A. H. and Weiss, G. M. (2017). Identifying sunni extremist propaganda with deep learning. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE.
- Kijewski, P., Jaroszewski, P., Urbanowicz, J. A., and Armin, J. (2016). The never-ending game of cyberattack attribution. In *Combating Cybercrime and Cyberterrorism*, pages 175–192. Springer.
- Kincaid, J. P., Fishburne Jr, R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Institute for Simulation and Training, University of Central Florida.
- McCarthy, P. M. and Jarvis, S. (2010). Mtd, vocd-d, and hd-d: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior research methods*, 42(2):381–392.
- McKernan, B. (2016). Isis' new magazine rumiyah shows the terror group is 'struggling to adjust to losses'. *The Independent*.
- Mencarini, M. and Sensidoni, G. (2017). Multilanguage semantic behavioural algorithms to discover terrorist related online. In *Proceedings of the First Italian Conference on Cybersecurity (ITASEC17)*, volume 1816. CEUR.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Montemurro, M. A. (2001). Beyond the zipf-mandelbrot law in quantitative linguistics. *Physica A: Statistical Mechanics and its Applications*, 300(3-4):567–578.
- Narayanan, A., Paskov, H., Gong, N. Z., Bethencourt, J., Stefanov, E., Shin, E. C. R., and Song, D. (2012). On the feasibility of internet-scale author identification. In *2012 IEEE Symposium on Security and Privacy*, pages 300–314. IEEE.
- Neal, T., Sundararajan, K., Fatima, A., Yan, Y., Xiang, Y., and Woodard, D. (2017). Surveying stylometry techniques and applications. *ACM Comput. Surv.*, 50(6):86:1–86:36.
- Nouh, M., Nurse, J. R., and Goldsmith, M. (2019). Understanding the radical mind: Identifying signals to detect extremist content on twitter. *arXiv preprint arXiv:1905.08067*.
- Saif, H., Dickinson, T., Kastler, L., Fernandez, M., and Alani, H. (2017). A semantic graph-based approach for radicalisation detection on social media. In *European semantic web conference*, pages 571–587. Springer.
- Shane, S. and Hubbard, B. (2014). Isis displaying a deft command of varied media. *New York Times*, 30.
- Sikos, J., David, P., Habash, N., and Faraj, R. (2014). Authorship analysis of inspire magazine through stylistic and psychological features. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 33–40. IEEE.
- Šišková, Z. (2012). Lexical richness in efl students' narratives. *Language Studies Working Papers*, 4:26–36.
- Sun, F., Guo, J., Lan, Y., Xu, J., and Cheng, X. (2016). Sparse word embeddings using l1 regularized online learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2915–2921. AAAI Press.
- Tausczik, Y. R. and Pennebaker, J. W. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.
- Vergani, M. and Bliuc, A.-M. (2018). The language of new terrorism: differences in psychological dimensions of communication in dabiQ and inspire. *Journal of Language and Social Psychology*, 37(5):523–540.
- Weimann, G. (2004). *www.terror.net: How modern terrorism uses the Internet*, volume 116. DIANE Publishing.
- Wignell, P., Tan, S., O'Halloran, K., and Lange, R. (2017). A mixed methods empirical examination of changes in emphasis and style in the extremist magazines dabiQ and rumiyah. *Perspectives on Terrorism*, 11(2):2–20.
- Wilner, A. S. (2018). Cybersecurity and its discontents: Artificial intelligence, the internet of things, and digital misinformation. *International Journal*, 73(2):308–316.

A.2.2 Senpy: A Framework for Semantic Sentiment and Emotion Analysis Services

Title	Senpy: A Framework for Semantic Sentiment and Emotion Analysis Services
Authors	Sánchez-Rada, J. Fernando and Araque, Oscar and Iglesias, Carlos A.
Journal	Knowledge-Based Systems
Impact factor	5.921
ISSN	0950-7051
Volume	190
Year	2020
Keywords	Classification, Emotion analysis, Evaluation, Linked data, Sentiment analysis
Pages	105193
DOI	https://doi.org/10.1016/j.knosys.2019.105193
Online	http://www.sciencedirect.com/science/article/pii/S0950705119305313
Abstract	Senpy is a framework to develop, evaluate and publish web services for sentiment and emotion analysis in text. The framework is aimed towards both developers and users. For developers, it is a means to evaluate their classifiers and easily publish them as web services. For users, it is a way to consume Sentiment Analysis from different providers through the same interface. This is achieved through a combination of an API aligned with the NLP Interchange Format (NIF) service specification, the use of semantic formats and a series of well established vocabularies. The framework is Open Source, and has been used extensively in several projects. As a result, several Senpy Open Source services are available for use and download.



Original software publication

Senpy: A framework for semantic sentiment and emotion analysis services[☆]

J. Fernando Sánchez-Rada^{*}, Oscar Araque, Carlos A. Iglesias

Grupo de Sistemas Inteligentes, Universidad Politécnica de Madrid, Spain



ARTICLE INFO

Article history:

Received 10 April 2019

Received in revised form 2 September 2019

Accepted 3 November 2019

Available online 7 November 2019

Keywords:

Sentiment analysis

Emotion analysis

Linked data

Classification

Evaluation

ABSTRACT

Senpy is a framework to develop, evaluate and publish web services for sentiment and emotion analysis in text. The framework is aimed towards both developers and users. For developers, it is a means to evaluate their classifiers and easily publish them as web services. For users, it is a way to consume sentiment analysis from different providers through the same interface. This is achieved through a combination of an API aligned with the NLP Interchange Format (NIF) service specification, the use of semantic formats and a series of well established vocabularies. The framework is Open Source, and has been used extensively in several projects. As a result, several Senpy Open Source services are available for use and download.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Sentiment analysis is a field of research and application in full expansion, driven by the popularity of social media and the need to give meaning to the concept of collective opinions [1]. A large number of sentiment analysis tools and services have been developed in recent years. Unfortunately, the lack of standard tools and APIs makes development costly, and it makes it harder to use or evaluate several services.

Senpy is an opinionated framework to publish sentiment and emotion analysis algorithms as web services. Its ultimate aim is to facilitate the growth of sentiment analysis by providing services that can be interchanged and easily evaluated. This will contribute to the quality and quantity of sentiment analysis services available to the research community.

2. Problems and background

Senpy addresses three challenges of sentiment analysis services: model heterogeneity, interoperability of APIs and formats, and service evaluation.

There is a wide range of models for sentiment and emotion. A sentiment model may include several classes or levels (e.g. positive, negative, and neutral), and a polarity value within a given

range (e.g., from -1 to $+1$). Emotion models are more complicated, and they are usually grouped into categorical models (e.g., Ekman's model uses 6 categories of emotion), and dimensional models (e.g., the VAD model represents emotions as vectors in a 3-dimensional space). Model heterogeneity may cause ambiguity and confusion, unless the model used is explicit and available to every user. For that reason, (EmotionML) [2] provides the means to link annotations to a model that can be defined and re-used.

A second issue is that most analysis tools and services use ad-hoc APIs, representation formats (e.g., JSON and XML) and schemas. This is a burden for consumers, who need to study and adapt to each service. It also hinders research, as it makes it harder to compare different approaches and solutions.

Lastly, evaluating services is crucial for two main reasons: (1) to compare different approaches in the state of the art; and (2) to estimate the performance of a given service in a specific domain (e.g., evaluating over a dataset of posts from a niche social network in the electronics domain). The lack of evaluation APIs or options to classify a dataset in bulk means that researchers need to learn to manually consume a service before they know it is appropriate for their use case.

Earlier works propose a linked data approach to solve these aspects [3,4], but the concepts and tooling behind ontologies and linked data publishing are unfamiliar to both the linguistic community and developers. Senpy aims to bridge this gap by presenting a tool that creates full-fledged services that benefit from the power of semantic technologies without requiring any prior knowledge of web or semantic technologies.

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.knosys.2019.105193>.

^{*} Corresponding author.

E-mail addresses: jf.sanchez@upm.es (J.F. Sánchez-Rada), o.araque@upm.es (O. Araque), carlosangel.iglesias@upm.es (C.A. Iglesias).

<https://doi.org/10.1016/j.knosys.2019.105193>

0950-7051/© 2019 Elsevier B.V. All rights reserved.

Plugin	Dataset	Accuracy	Precision_macro	Recall_macro	F1_macro	F1_weighted	F1_micro	F1
endpoint:plugins/sentiment-vader_0.1.1	vader	0.6907	0.3454	0.5	0.4085	0.5644	0.6907	0.4085
endpoint:plugins/sentiment-vader_0.1.1	sts	0.3107	0.1554	0.5	0.2371	0.1473	0.3107	0.2371

Fig. 1. Evaluating a service in two datasets through the Web UI.

3. Senpy framework

3.1. Architecture

At a high level, Senpy consists of three main modules: (1) a core that provides the HTTP service layer to consume and evaluate services; (2) a module to develop plug-ins that provide specific features (e.g., a sentiment analysis algorithm); and (3) a Web User Interface (Web UI) that can be used to consume and evaluate services in a user-friendly way, instead of directly interacting with the HTTP layer.

A plugin may be defined in a python module or a YAML file. When the Senpy tool is launched, it detects all plugins and plugin code available to it (i.e., working directory and configurable folders), and launches the Web UI and the HTTP server.

3.2. Functionalities

The main features of Senpy are:

- Development of sentiment and emotion classifiers that can be exposed as semantic multi-format HTTP services.
- Sentiment and emotion analysis from different providers (i.e., a service that uses Vader or one that uses SenticNet [5]) using the same interface (including a NIF-based [6] API and vocabularies [7]). In this way, applications do not depend on the API offered for these services. The API and vocabularies follow the draft guidelines for developing semantic services by the The Linked Data Models for Emotion and Sentiment Analysis W3 Community Group.¹
- Combination of services even if they use different models for sentiment (e.g., converting polarity levels from a “-1 to 1” interval to a “0 to 10” interval) or emotion (e.g., Ekman or VAD). Within a server, users can create processing pipelines that combine several analyses within a single request. For instance, asking for sentiment and emotion analysis in the same request, or for two complementary sentiment analyses.
- Evaluation of sentiment algorithms with well-known datasets. The included evaluation API allows users to compare specific combinations of services and datasets.
- A Web User-Interface for users of any skill level to consume and evaluate services. The interface visualizes the results in a user-friendly way, including tabular and multi-format visualization.

¹ <https://www.w3.org/community/sentiment/>.

4. Implementation

The framework is implemented with Python 3.5 using open source libraries. It can be installed manually or from pip/PyPI. The HTTP layer is built on Flask, and its Web UI uses bootstrap and other Javascript libraries. RDFLib is used to convert to and from semantic formats (XML-RDF, JSON-LD, and Turtle). Scikit-Learn provides machine learning primitives.

A docker image is also provided, which contains all necessary dependencies and can be used as a standalone server or to develop new services. Moreover, the main repository contains the configuration necessary to deploy services to a Kubernetes cluster.

5. Illustrative examples

Users may explore the main features of Senpy on the online demo.² It includes several Open Source services³ for sentiment and emotion analysis. The project's documentation⁴ contains a list of features as well as instructions for users and developers. It explains different ways in which these and services can be consumed, and how to evaluate their performance in a series of public datasets (Fig. 1). It also covers how to develop new sentiment analysis services.

6. Conclusions

The framework presented in this paper has the potential to ease adoption, development, integration and evaluation of sentiment and emotion analysis services.

Acknowledgments

This work has been partially funded by the SEMOLA project, funded by the Ministry of Economy and Competitiveness of Spain (TEC2015-68284-R).

Appendix. Required metadata

Current executable software version

See Table 1

Current code version

See Table 2

² <http://senpy.gsi.upm.es/>.

³ <http://github.com/gsi-upm/senpy-plugins-community>.

⁴ <https://senpy.readthedocs.io>.

Table 1

Software metadata.

Nr.	Software metadata description	Please fill in this column
S1	Current software version	1.0.1
S2	Permanent link to executables of this version	https://github.com/gsi-upm/senpy/tree/1.0.1
S3	Legal software license	Apache license, Version 2.0
S4	Computing platform/Operating system	Linux, OS X, Microsoft windows, Unix-like
S5	Installation requirements & dependencies	Python 3.5+, NetworkX, NLTK, RDFLib, Scipy, Scikit-learn, Numpy
S6	Link to user manual	https://senpy.readthedocs.io/
S7	Support email for questions	jf.sanchez@upm.es

Table 2

Code metadata.

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1.0.1
C2	Permanent link to code/repository used of this code version	https://github.com/gsi-upm/senpy/tree/1.0.1
C3	Legal code license	Apache license, Version 2.0
C4	Code versioning system used	git
C5	Software code languages, tools, and services used	Python
C6	Compilation requirements, operating environments & dependencies	Python 3.5+, NetworkX, NLTK, RDFLib, Scipy, Scikit-learn, Numpy
C7	Link to developer documentation/manual	https://senpy.readthedocs.io/
C8	Support email for questions	jf.sanchez@upm.es

References

- [1] B. Liu, Sentiment analysis and opinion mining, *Synth. Lect. Hum. Lang. Technol.* 5 (1) (2012) 1–167.
- [2] M. Schröder, et al., EmotionML—an upcoming standard for representing emotions and related states, in: *Affective Computing and Intelligent Interaction*, Vol. 6974, Springer Berlin Heidelberg, 2011, pp. 316–325.
- [3] S. Hellmann, J. Lehmann, S. Auer, M. Brümmer, Integrating NLP using linked data, in: *The Semantic Web—ISWC*, Springer, 2013, pp. 98–113.
- [4] J.F. Sánchez-Rada, et al., Towards a common linked data model for sentiment and emotion analysis, in: *Proceedings of the LREC 2016 Workshop Emotion and Sentiment Analysis*, 2016, pp. 48–54.
- [5] E. Cambria, R. Speer, C. Havasi, A. Hussain, Senticnet: A publicly available semantic resource for opinion mining, in: *AAAI Fall Symposium Series*, 2010.
- [6] S. Hellmann, J. Lehmann, S. Auer, M. Nitzschke, NIF combinator: Combining nlp tool output, in: *Knowledge Engineering and Knowledge Management*, Springer, 2012, pp. 446–449.
- [7] J.F. Sánchez-Rada, C.A. Iglesias, Onyx: A linked data approach to emotion representation, *Inf. Proc. Manag.* 52 (1) (2016) 99–114.

A.2.3 GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter

Title	GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter
Authors	Benito, Diego and Araque, Oscar and Iglesias, Carlos A.
Proceedings	Proceedings of the 13th International Workshop on Semantic Evaluation
ISBN	-
Year	2019
Keywords	-
Pages	396–403
DOI	http://dx.doi.org/10.18653/v1/S19-2070
Online	https://www.aclweb.org/anthology/S19-2070
Abstract	This paper describes the GSI-UPM system for SemEval-2019 Task 5, which tackles multilingual detection of hate speech on Twitter. The main contribution of the paper is the use of a method based on word embeddings and semantic similarity combined with traditional paradigms, such as n-grams, TF-IDF and POS. This combination of several features is finetuned through ablation tests, demonstrating the usefulness of different features. While our approach outperforms baseline classifiers on different sub-tasks, the best of our submitted runs reached the 5th position on the Spanish subtask A.

GSI-UPM at SemEval-2019 Task 5: Semantic Similarity and Word Embeddings for Multilingual Detection of Hate Speech Against Immigrants and Women on Twitter

Diego Benito, Oscar Araque, and Carlos A. Iglesias

Intelligent Systems Group

Universidad Politécnica de Madrid

Madrid, Spain

Avenida Complutense, 30

{d.benito,o.araque,carlosangel.iglesias}@upm.es

Abstract

This paper describes the GSI-UPM system for SemEval-2019 Task 5, which tackles multilingual detection of hate speech on Twitter. The main contribution of the paper is the use of a method based on word embeddings and semantic similarity combined with traditional paradigms, such as n-grams, TF-IDF and POS. This combination of several features is fine-tuned through ablation tests, demonstrating the usefulness of different features. While our approach outperforms baseline classifiers on different sub-tasks, the best of our submitted runs reached the 5th position on the Spanish sub-task A.

1 Introduction

Information available in social networks is the result of many interactions between users and their activity on the net. Unfortunately, hate speech and other misuses are proliferating on the Internet. Hate speech authors justify their conduct based on the freedom of speech argument. Thus, a debate over hate speech legislation and freedom of speech has been generated (Herz and Molnar, 2012).

The task to decide if a piece of text contains hate speech is not trivial, even for humans. Being subject to different interpretations and opinions, the manifestations of hate speech become difficult to define. Based on previous hate speech statements (Fortuna and Nunes, 2018; Schmidt and Wiegand, 2017), this phenomenon could be defined as offensive or humorist content in form of text, video, or images that attacks, diminishes, incites violence or hate against groups or individuals, based on actual or perceived specific characteristics such as physical appearance, religion, descent, national or ethnic origin, sexual orientation, gender identity, or any other.

Hate speech topic has gained impact and popularity in recent years, which is reflected not only

by the increased media coverage but also by the growing political attention. Regarding the specific forms of hate speech that we deal with, sexism and racism victims increased during 2017 according to the FBI hate crime statistics¹.

For this reason, participating in SemEval2019 Task 5 (Basile et al., 2019) is such an interesting challenge. The proposed task consists in Hate Speech detection in Twitter messages featured by two specific different targets intrinsically related to the phenomena mentioned above, immigrants and women. The task is enriched by adding a multilingual perspective fostering the research for both English and Spanish messages.

The system proposed relies on a supervised classifier using different text features combined with several strategies with the aim of finding an optimal performance. The remainder of this paper is structured as follows. After this introductory section, Section 2 reviews related work. Following, the proposed classification model is described in Section 3. Then, Section 4 presents the experimental results, and finally, Section 5 concludes the paper with a final discussion.

2 Related Work

Most of our literature review from the field is referenced by previous survey research (Fortuna and Nunes, 2018; Schmidt and Wiegand, 2017).

Multiple procedures have been implemented, since more traditional feature engineering, such as n-grams (Waseem and Hovy, 2016) or Part-of-Speech (POS) (Davidson et al., 2017) to more complex deep learning architectures (Yuan et al., 2016; Badjatiya et al., 2017).

According to the analyzed bias which motivates hate speech, general hate speech (Silva et al., 2016) is considered by the majority, however,

¹<https://ucr.fbi.gov/hate-crime/>

there is large research that focuses particularly on racism (Kwok and Wang, 2013) and sexism (Hewitt et al., 2016). Though it is not exactly a form of hate speech, cyberbullying is a very related problem with some study research (Cortis and Handschuh, 2015).

3 System Overview

The system relies on a supervised machine learning algorithm. This final classification step is fed by a data processing pipeline formed by the preprocessing and the feature extraction modules. Regarding the implementation, Python has been used, with the additional capabilities provided by the libraries scikit-learn (Pedregosa et al., 2011), NLTK (Bird and Loper, 2004), and GSITK (Araque et al., 2017)². Figure 1 illustrates the system architecture from a general perspective.

3.1 Preprocessing

In this phase, the raw text is taken and cleaned using common NLP techniques (Manning et al., 1999): removal of punctuation marks, special characters, URLs, and stop-words. Tweet preprocessing relies on tokenization, user mentions normalization, the appearance of hashtags, URLs, and all caps words flagged supported by the tools provided by GSITK. In addition, tokens are lemmatized using the Porter stemmer (Porter, 1980).

3.2 Feature Engineering

Different features have been taken into account during the feature engineering stage. Such features are divided into subcategories: statistical features, content analysis, word embeddings, semantic features, and linguistic features.

3.2.1 Statistical Features

We collected word and character n-grams evaluating both approaches, Bag-of-Words (BOW) and Term Frequency - Inverse Document Frequency (TF-IDF). The reason to include character n-grams comes from the Twitter domain, where texts are short and misspelling may occur; this can be attenuated at the character level (Schmidt and Wiegand, 2017). Apart from the mentioned reasoning, previous research (Mehdad and Tetreault, 2016) has shown the effectiveness of character n-grams in the problem of offensive language.

²<https://github.com/gsi-upm/semEval2019-hatespeech>

Besides tokens included within the text corpus, the system also includes frequencies from external lexicons that are thought for hate speech³, sentiment analysis (Hu and Liu, 2004; Liu et al., 2005), and subjectivity analysis (Pang and Lee, 2004).

3.2.2 Content Analysis

As seen, sentiment and subjectivity information has been included. Hate speech can be considered as subjective content, and a relation between subjectivity, sentiments, and emotions can occur. Besides, hate speech is expected to have a negative polarity, so text subjectivity and polarity provided by the TextBlob (Loria et al., 2014) library were included in the analysis.

Topic modeling methods were added to the study, particularly, Latent Dirichlet Allocation (LDA) (Blei et al., 2003) in order to extract the topic of each tweet in combination with the appearance of hashtags (topics) inside the corpus.

3.2.3 Word Embeddings

In order to solve the lack of semantic of words in n-grams features, word distributed representations based on word embeddings models are evaluated. Pre-trained word vectors convert words into vector space where semantically similar words tend to appear close by each other. In this system, a vector is extracted for each word in the input text; then, as done in (Araque et al., 2017), the average pooling operation is performed on all word vectors, resulting in a vector of the same dimensions as the original word vectors.

3.2.4 Semantic Features

A central part of the system consists of a method (Araque et al., 2019) that exploits the semantic similarity measure that a word embedding model provides, via cosine similarity. In general lines, this approach uses a lexicon to which the input text is projected, employing the similarity measure obtained from an embedding model.

The method considers a selection of words S that constitutes a lexicon vocabulary to which the input documents are projected. Given a text document (e.g., tweet), a similarity value between the input word vectors of that document and each of the words in S is computed. After iterating over all input words and all lexicon words, a matrix $m \times |S|$ is obtained, where m is the number of input words in a particular document. Following,

³<https://hatebase.org/>

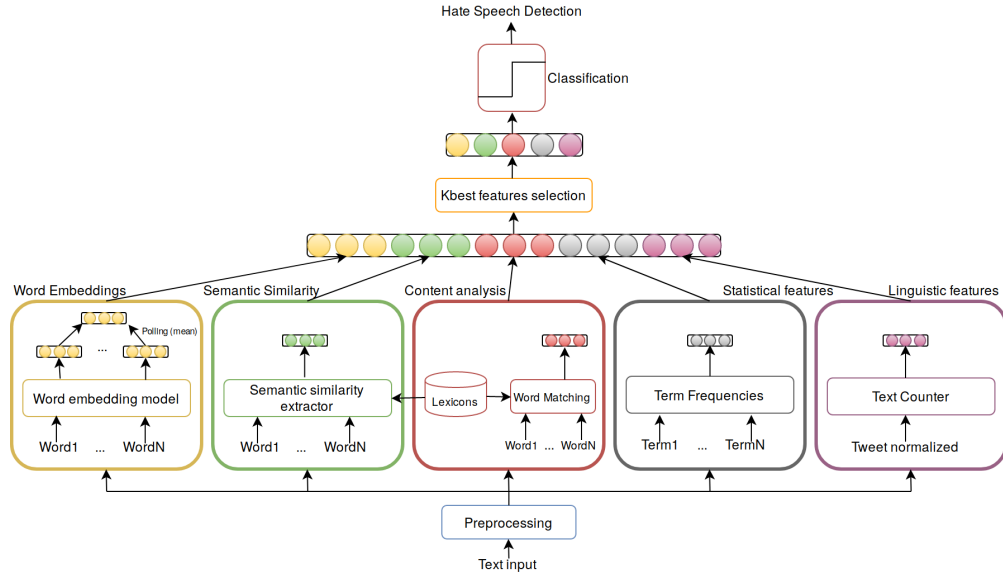


Figure 1: System Overview

the maximum pooling function is applied column-wise, obtaining the semantic similarity feature vector of dimensionality $|S|$.

In this work, the previously mentioned lexicons have been used, as well as a domain-oriented word selection, which have been extracted from the dataset. In this last approach, words were filtered by its frequency of appearance considering the document annotation, being the cutoff frequency an adjustable parameter.

3.2.5 Linguistic Features

The last set of features used are related to linguistic aspects. The proposed system considers the number of sentences, length from the tweet, POS stats, as well as some Twitter-related features such as the count of hashtags, URLs, mentions, all caps words, emojis, and exclamations.

3.3 Classification

Finally, the furthest step in the data processing pipeline makes use of a machine learning classifier. There are many options among machine learning models that can be used. In this project, we have evaluated the performance of three different types of algorithms: Logistic Regression, Support Vector Machines (SVM) with linear kernel, and Random Forest.

4 Experiments

This section presents the results obtained by the proposed system in the competition, considering both test and development phase submissions. Firstly, a data exploration has been carried out in order to analyze the data distribution, possible features to feed the classifier, and deficiencies in the data source. The evaluation of the different feature extraction approaches and the hyperparameter tuning has been done by using a cross-validation grid search. Special attention has been paid in the regularization parameter of the algorithms: “C” parameter in the Logistic Regression and Linear SVM case and “maximum depth” of the trees in the Random Forest case. Finally, the system is trained, and the evaluation metrics are computed. This workflow has been repeated several times from the feature extraction step, changing the set of features in every iteration.

4.1 Sub-task A

The goal of this task is to classify both Spanish and English tweets as hateful or not hateful. Systems are evaluated using standard evaluation metrics, including accuracy, precision, recall, and F1-score, but predictions are ranked by F1-score metric alone.

Team	Accuracy	Precision	Recall	F-score
English				
Best	0.506	0.65	0.566	0.457
SVM baseline	0.492	0.595	0.549	0.451
GSI-UPM	0.483	0.643	0.549	0.42
MFC baseline	0.579	0.289	0.5	0.367
Spanish				
Best	0.731	0.734	0.741	0.73
GSI-UPM	0.728	0.726	0.733	0.725
SVM baseline	0.705	0.701	0.707	0.701
MFC baseline	0.58	0.294	0.5	0.37

Table 1: Official test set results for Task A

Feature combination	Accuracy	Precision	Recall	F-score
English				
Official submission combination	0.777	0.774	0.780	0.775
Lexical, similarity, embeddings, and n-grams (1)	0.757	0.754	0.758	0.754
Bigrams, trigrams, similarity, and embeddings (2)	0.75	0.747	0.752	0.748
Embeddings, similarity, twitter stats, and LDA (3)	0.736	0.731	0.733	0.732
Spanish				
Official submission combination	0.856	0.856	0.852	0.853
Lexical, similarity, embeddings, and n-grams (1)	0.812	0.811	0.807	0.808
Bigrams, trigrams, similarity, and embeddings (2)	0.796	0.794	0.791	0.792
Embeddings, similarity, Twitter stats, and LDA (3)	0.784	0.781	0.782	0.782

Table 2: Development set results for Task A

Task A data was partitioned into train, development, and test sets. Train and development sets were used to obtain the best feature combination by training over the train set and testing over the development one. Finally, for the final submission, the predictions for the test set were obtained with a system trained over both train and development sets.

Test results, which represent the official submission, as well as development phase results are presented in Tables 1 and 2 respectively. Task organizers included two baselines (Basile et al., 2019) in the competition, a linear SVM based on a TF-IDF representation and a trivial model that assigns the most frequent label from the training set to all instances in the test set.

The Spanish-oriented system relies on linguis-

tic features (excepting POS), semantic similarity with a domain-oriented lexicon, sentiments (using the sentiment vocabulary weighted by the TF-IDF measure), word embeddings, topic modeling (both LDA and hashtags), and word and character TF-IDF n-grams. These features are filtered according to the ANOVA F-test, selecting the best 3,000. Linear SVM has been the selected machine learning algorithm for classification. On the other hand, the English-oriented system considers the same feature set excluding word embedding representation; the number of selected features has been set at 17,500. In contrast to the previous system, a Logistic Regression model was used to perform the classification.

Team	F-score(HS)	F-score(TR)	F-score(AG)	F-score (Avg)	EMR
English					
MFC baseline (Best)	0.367	0.452	0.445	0.421	0.58
GSI-UPM	0.421	0.686	0.556	0.555	0.312
SVM baseline	0.45	0.697	0.587	0.578	0.308
Spanish					
Best	0.729	0.798	0.737	0.755	0.705
GSI-UPM	0.725	0.79	0.735	0.75	0.624
SVM baseline	0.701	0.781	0.726	0.736	0.605
MFC baseline	0.37	0.424	0.413	0.402	0.588

Table 3: Official Results for Task B

Feature Combination	F-score(HS)	F-score(TR)	F-score(AG)	F-score (Avg)	EMR
English					
Official submission	0.775	0.811	0.723	0.770	0.665
(1)	0.754	0.797	0.712	0.755	0.641
(2)	0.748	0.788	0.699	0.745	0.628
(3)	0.731	0.767	0.687	0.728	0.611
Spanish					
Official submission	0.853	0.876	0.824	0.851	0.78
(1)	0.808	0.839	0.777	0.808	0.732
(2)	0.792	0.843	0.776	0.804	0.718
(3)	0.782	0.836	0.783	0.800	0.714

Table 4: Development Results for Task B

4.2 Sub-task B

The goal of this task is firstly to classify hateful tweets (i.e., tweets identified as hate speech against women or immigrants) as aggressive or not aggressive, and secondly to identify the target harassed as individual or generic (i.e., single person or group). Systems are evaluated by two criteria: partial match and exact match (Basile et al., 2019), but predictions are ranked by exact match metric alone.

For this task, the data has been delivered in the same way than sub-task A, so we emulated the same workflow than before, but in this case, considering solely hateful tweets. In this case, there are different distributions (Basile et al., 2019) along languages and sets, but different labels show a similar layout. This result goes in line with the work presented in (ElSherief et al., 2018), which states that directed hate speech is more informal, angrier, and often explicitly attacks the victim. Regarding the language, Spanish-speaking people tend to be more aggressive and more direct towards specific individuals. Seeing this skewed distribution, we outlined the idea to balance aggres-

siveness and directed messages by oversampling hateful tweets with not hateful ones, assuming that not hateful tweets are not aggressive nor directed.

As done previously, Tables 3 and 4 present official and development results, respectively. The Spanish-oriented system in this task is identical to that from Task A, but finally selecting 2,500 features. For the English case, in light of aggressiveness and target tweets, a different combination of features have been chosen. In order to detect aggressive tweets, all features except semantic similarity have been used, filtering the 32,500 best. Otherwise, for target messages, the complete set of features (sentiments and subjectivity were included by means of TF-IDF and semantic similarity) are used just considering the 2,500 best. Finally, different models were applied for each label, Logistic Regression for Target label and Linear SVM for the Aggressive one. The same algorithm selection was made in the Spanish case.

4.3 Discussion

In general terms, the results obtained are auspicious: the best submitted system achieved the 5th

position in the Spanish Task A, 0.5% points under the best result obtained in the same task. For the Spanish Task B, the proposed model outperforms the baseline. In contrast to this, results in English Task A are lower than expected, where there was not any team that surpassed the 50% threshold in terms of F-score. As a general trend, test set results are worse than development results, which may indicate that our systems suffer over-fitting, and cannot generalize properly. This observation is enforced by attending to the English Task B, where no system has surpassed the baseline.

Since the data distribution is equal along languages in Task A (Basile et al., 2019), the difference in performance across languages may be due to Spanish speaking people are more explicit when typing any utterance with hate speech goals. As previously mentioned, we have observed that this type of hate speech messages show more aggressiveness. Language characteristics may be involved since the Spanish language has a morphologically-rich nature than English.

The presented results constitute the outcome of exhaustive experimentation of a variety of feature combination tests. In contrast with earlier work, semantic similarity and word embeddings representations do not produce such high performance results when compared to other domains such as sentiment analysis (Araque et al., 2019) and sleep disorder detection (Suarez et al., 2018) tasks. This circumstance suggests that hate speech detection is still an open challenge and more research must be done into the specific characteristics of such an exciting task.

Attending to the Spanish case, sentiment information and character n-grams were features that helped in a meaningful manner, confirming the issues raised in Sect. 3. For the English case, the improvement of the proposed features was incremental. While subjectivity and emojis had a relevant role in the results, this improvement was not as high as in the Spanish case. In light of the complexity of the hate speech domain, it could be argued that attending to word context instead of isolated words could help in the analysis. Indeed, n-grams include this type of information to some extent, but capturing the grammatical dependencies within a sentence (Chen, 2011) or template based strategies (Warner and Hirschberg, 2012) could enhance the performance.

5 Conclusions

This paper described the GSI-UPM hate speech detection system presented to participate in SemEval-2019 Task 5, which revolves around analyzing text messages from Twitter. In order to tackle this, a machine learning based approach has been developed. The different features that feed this system have been thoroughly evaluated, considering its suitability in the field of hate speech detection. It has been seen that both novel and traditional approaches do not yield so promising when used separately. Nevertheless, properly combining several types of features, as well as with content analysis features (e.g., sentiments and subjectivity) can improve the system to the point of reaching a reasonably good performance.

Concerning the achieved goals, the highest ranking was 5th place on the Spanish sub-task A, being 0.5% apart from the best performing system. This is, undoubtedly, a promising result that highlights the capacity of the proposed method to obtain nearly state-of-the-art performance in this task. When comparing with the same sub-task in the English case, in which we scored lower, it is necessary to study further the applicability of the system to different languages.

As future work, several lines of work could be addressed. Firstly, we plan to implement deep learning architectures which have shown to obtain better results in previous research (Zhang and Luo, 2018; Zhang et al., 2018). In addition, in order to afford imbalanced distributions, data augmentation (Hemker, 2018) techniques could be explored. Also, context-aware approaches could represent an improvement (Dinakar et al., 2012), since having general knowledge of hate speech (e.g., anti-LGBT or racism) may boost the performance of learning systems.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness under the R&D project SEMOLA (TEC2015-68284-R) and the European Union with Trivalent (H2020 Action Grant No. 740934, SEC-06-FCT-2016).

References

Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Snchez-Rada, and Carlos A. Iglesias. 2017. *Enhanc-*

- ing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236 – 246.
- Oscar Araque, Ganggao Zhu, and Carlos A. Iglesias. 2019. A semantic similarity-based perspective of affect lexicons for sentiment analysis. *Knowledge-Based Systems*, 165:346 – 359.
- Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 759–760, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.
- Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*. Association for Computational Linguistics.
- Steven Bird and Edward Loper. 2004. Nltk: The natural language toolkit. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*, ACLdemo '04, Stroudsburg, PA, USA. Association for Computational Linguistics.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.
- Ying Chen. 2011. Detecting offensive language in social medias for protection of adolescent online safety.
- Keith Cortis and Siegfried Handschuh. 2015. Analysis of cyberbullying tweets in trending world events. In *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*, i-KNOW '15, pages 7:1–7:8, New York, NY, USA. ACM.
- Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.
- Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Trans. Interact. Intell. Syst.*, 2(3):18:1–18:30.
- Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth M. Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media. *CoRR*, abs/1804.04257.
- Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):85.
- Konstantin Hemker. 2018. Data augmentation and deep learning for hate speech detection. Master's thesis, Imperial College London.
- Michael Herz and Peter Molnar. 2012. *The content and context of hate speech*. Cambridge University Press.
- Sarah Hewitt, T. Tiropanis, and C. Bokhove. 2016. The problem of identifying misogynist language on twitter (and other online social spaces). In *Proceedings of the 8th ACM Conference on Web Science*, WebSci '16, pages 333–335, New York, NY, USA. ACM.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.
- Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.
- Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, pages 342–351, New York, NY, USA. ACM.
- Steven Loria, P Keen, M Honnibal, R Yankovsky, D Karesh, E Dempsey, et al. 2014. Textblob: simplified text processing. *Secondary TextBlob: Simplified Text Processing*.
- C.D. Manning, C.D. Manning, H. Schütze, and H.A. SCHUTZE. 1999. *Foundations of Statistical Natural Language Processing*. Mit Press. MIT Press.
- Yashar Mehdad and Joel Tetreault. 2016. Do characters abuse more than words? In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 299–303.
- Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.
- M.F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.
- Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10.

- Leandro Araújo Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *ICWSM*, pages 687–690.
- D. Suarez, O. Araque, and C. A. Iglesias. 2018. [How well do spaniards sleep? analysis of sleep disorders based on twitter mining](#). In *2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 11–18.
- William Warner and Julia Hirschberg. 2012. [Detecting hate speech on the world wide web](#). In *Proceedings of the Second Workshop on Language in Social Media, LSM '12*, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Zeeraq Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.
- Shuhan Yuan, Xintao Wu, and Yang Xiang. 2016. A two phase deep learning model for identifying discrimination from tweets. In *EDBT*, pages 696–697.
- Ziqi Zhang and Lei Luo. 2018. [Hate speech detection: A solved problem? the challenging case of long tail on twitter](#). *CoRR*, abs/1803.03662.
- Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *The Semantic Web*, pages 745–760, Cham. Springer International Publishing.

A.2.4 How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining

Title	How Well Do Spaniards Sleep? Analysis of Sleep Disorders Based on Twitter Mining
Authors	Suarez, Daniel and Araque, Oscar and Iglesias, Carlos A.
Proceedings	2018 Fifth International Conference on Social Networks Analysis, Management and Security (SNAMS)
ISBN	978-1-5386-9588-3
Year	2018
Keywords	Machine learning, data mining, Feature extraction, natural language processing, pattern classification, social networking (online), machine learning, Task analysis, Twitter, learning (artificial intelligence), common sleep disorder, feature extraction, feature extraction method, Geology, insomnia, insomnia treat, machine learning classifiers, medical computing, medical disorders, numerous research studies, Security, sleep, sleep disorder, sleep disorders, social mining, social network, Spanish, Twitter mining, word embedding models, word insomnia
Pages	11-18
DOI	https://doi.org/10.1109/SNAMS.2018.8554488
Online	https://ieeexplore.ieee.org/document/8554488
Abstract	Twitter is a social network that allows its users to exchange messages of 280 characters with the possibility of accompanying them with a photo, video and/or link. This social network has been used as a source of data for numerous research studies on the human being. This study aims to analyse and characterize the messages coming from Spanish-speaking users related to the most common sleep disorder in our society, insomnia. For this purpose, this study provides two machine learning classifiers that enable the classification of users with insomnia together with the self-reported cause. In this context, this paper proposes a novel feature extraction method that exploits the similarity measure that can be computed in word embeddings models. For training these classifiers, a dataset of tweets in Spanish containing the word insomnia has been manually annotated to draw conclusions about the geographical distribution, symptoms and the different topics that users with insomnia treat. In addition, a second dataset has been collected formed by two groups of users from Spain with insomnia and without insomnia. Analysing the timeline of both groups we have been able to extract the differences in the patterns of activity on Twitter of each of these groups.

How well do Spaniards sleep? Analysis of Sleep Disorders based on Twitter mining

Daniel Suárez*, Oscar Araque† and Carlos A. Iglesias†

Intelligent Systems Group

Universidad Politécnica de Madrid

Madrid, Spain

Avenida Complutense, 30

Email: *daniel.suarez.souto@alumnos.upm.es, †{o.araque, carlosangel.iglesias}@upm.es,

Abstract—Twitter is a social network that allows its users to exchange messages of 280 characters with the possibility of accompanying them with a photo, video and/or link. This social network has been used as a source of data for numerous research studies on the human being. This study aims to analyse and characterize the messages coming from Spanish-speaking users related to the most common sleep disorder in our society, insomnia. For this purpose, this study provides two machine learning classifiers that enable the classification of users with insomnia together with the self-reported cause. In this context, this paper proposes a novel feature extraction method that exploits the similarity measure that can be computed in word embeddings models. For training these classifiers, a dataset of tweets in Spanish containing the word insomnia has been manually annotated to draw conclusions about the geographical distribution, symptoms and the different topics that users with insomnia treat. In addition, a second dataset has been collected formed by two groups of users from Spain with insomnia and without insomnia. Analysing the timeline of both groups we have been able to extract the differences in the patterns of activity on Twitter of each of these groups.

Index Terms—insomnia, sleep disorder, Twitter, social mining, natural language processing, machine learning, feature extraction

I. INTRODUCTION

The catalogue of different sleep disorders is part of the main problems that must be faced by medicine today. The percentage of people suffering from some of these disorders is 31% in Western Europe, 56% in the USA and 23% in Japan [1].

A study by DOPPS (Dialysis Outcomes and Practice Patterns) concluded that there are a number of common clinical features associated with people who sleep a few hours of sleep such as increased body mass index, pain, coronary artery disease, congestive heart failure, diabetes, lung disease,

psychiatric disorders, peripheral arterial disease, depression, and pruritus [2].

However, it is estimated that only some of these people follow some kind of medical treatment for these disorders. This fact makes necessary to create techniques for medical professionals to predict a sleep disorders in an early way and to be able to advise healthy sleep routines and therapies.

This article aims to analyse sleep disorders based on Twitter mining. This social network allows its users to publish messages of a maximum of 280 characters with the possibility of accompanying them with a photo, video, and/or link.

In our case, we are interested in determining if it is possible to characterize potential people to suffer insomnia through the self-reported messages posted on a social network. Specifically, the research will be based on Spanish-speaking messages and users in the social network Twitter.

We are also interested in determining the proportion of users who suffer from each of the different symptoms of insomnia, allowing us to carry out a more extensive study of the phenomenon of insomnia. According to the third edition of the International Criteria for Sleep Disorders (ICSD-3) [3], the main symptoms are: difficulty beginning sleep, short sleep duration and poor energy during the day. Also, another objective consists in measuring the proportion of users that undergo medical treatments of insomnia.

To achieve these goals we will develop two classifiers through machine learning techniques responsible for automating the tasks of characterizing users and tweets on the topic of insomnia. Also, with the aim of fostering research in this field, we make public the generated datasets ¹.

The reminder of this paper is organized as follows. Sect. II reviews related work. Sect. III elaborates the methodology for collecting and annotating data. Sect. IV describes the techniques uses for tweet classification. Sect. V presents the analysis of the results of the data. Finally Sect. VI discusses the conclusions of this work.

¹<https://github.com/gsi-upm/dreampy-resources>

ORCID info:

D. Suárez: <https://orcid.org/0000-0002-6225-3324>

O. Araque: <https://orcid.org/0000-0003-3224-0001>

C. A. Iglesias: <https://orcid.org/0000-0002-1755-2712>

Corresponding author: Oscar Araque. TIF: +34 910672166.

II. RELATED WORK

Nowadays, social networks have become spaces where people communicate their daily activities, concerns and problems to others [4]. Recently, many studies have emerged in which social networks were used as a data source produced by users to attempt to draw conclusions on health issues [5].

One of the social networks that has experienced this growth in recent years has been Twitter. Because of this, recently a large body of research has been devoted to use Twitter to analyse users behaviour and personality, including possible suicidal tendencies [6], people on diet [7], influence of political candidates on voters [8], users characteristics for engaging customers [9], and many others. Many of these works make use of common text-based features [10] such as keyword extraction, topic classification, Part-of-Speech and Name Entity Recognition.

In addition to the different fields we have mentioned in the previous section, research on sleep disorders has also been carried out using social networks prior to this study. These researches are based on comparisons between a group of users who suffer from insomnia and another group of users who do not suffer from it.

Jamison-Powell et al. [11] analysed a corpus of English Tweets in order to explore the role of social media in the discussion of mental health issues, and with particular reference to insomnia and sleep disorders. They present a content analysis which revealed that tweets that contained the word “insomnia” contained significantly more negative health information than a random sample, strongly suggesting that individuals were making disclosures about their sleep disorder. In addition, they also reveal the existence of two themes present in the content of this type of tweets: (i) describing the experience, in which they catalogue the way in which users transmit their insomnia and (ii) coping with insomnia, in which they catalogue how users manage their insomnia. They determine a tendency of users with insomnia to release their frustration of not being able to sleep through social networks. Based on this conclusion, we claim that it is possible to carry out studies on insomnia through the information provided by users suffering from this disorder on twitter.

Another study has also been carried out on the Chinese social network Sina Weibo, but based on the same context [12] of insomnia. Their objective was to evaluate the feasibility of using social media to understand whether and how sleep complaints are discussed online and the feasibility of using their data to inform the detection and prevention of insomnia. To do this, they annotated a corpus in terms of themes and symptoms present in the tweets. For the task of this labelling they developed a text classifier machine learning. They also conducted a demographic study of the tweets and a study of activity on the social network of users. Their results have allowed us to know the most common topics among users suffering from insomnia.

Finally, another relevant research has been done to characterize sleep issues on Twitter [13]. Its objective was to

determine whether social media can be used as a method for carrying out research focused on sleep-related issues. To do this they built a corpus of tweets in English containing words related to insomnia. Their conclusions were that users with insomnia present a tendency of negative sentiment in their publications. They also carried out a study of the characteristics of these users on Twitter, concluding that they are less active and have fewer followers.

III. INSOMNIA CORPUS COLLECTION AND ANNOTATION

A. Dataset and methodology

The capture of tweets was done through the Streaming Twitter API² that allows us to capture tweets in real time. For this task we used an open-source Python library called Tweepy [14] that allows us to communicate with the Twitter API in a simple and intuitive way.

The following criteria were considered for collecting the tweets:

- They must have been in Spanish.
- They had to contain the word insomnia, because it is the phenomenon we want to study.
- They could not be re-tweets (a re-post of a tweet originally posted by a different user) because we are only interested in experiences expressed by people who claim to have insomnia.

In addition to the tweet text, we also store metadata such as the date and time the tweet was published, the user ID and his location, and the tweet ID.

The data collection process has been carried out in two different periods. The first data capture (*General Insomnia dataset*) was made between December 14, 2017 and January 4, 2018. It produced a sample of 54432 tweets following the aforementioned criteria. This sample has been used in the geographical study of insomnia explained in the Sect. III-B. The second capture (*Spanish Insomnia dataset*) was made between April 2 and 10, restricted to Spain and produced a sample of 2361 tweets.

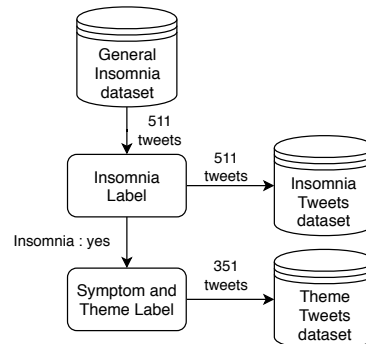


Fig. 1. Methodology for the labelling and creation of datasets.

²<https://developer.twitter.com/en/docs>

From this data, several datasets have been created for our experiments (see Fig. 1).

- *Insomnia Tweets dataset*. This dataset is formed by 511 random tweets of the *General Insomnia dataset*. In it we labelled the *Insomnia* label explained in the Sect. III-C, and it is the one we used to train the *Insomnia Classifier*, explained in the Sect. V-A.
- *Theme Tweets dataset*. This dataset is made up of the 351 tweets of *Insomnia Tweets dataset* tagged with the *Insomnia* label as 'yes' (i.e. tweets in which the author claims to suffer insomnia). In this dataset we labelled the labels *Themes* and *Symptoms* explained in the Sect III-C, and we used it to train the *Theme Classifier*, explained in the Sect V-B.
- *Spanish Insomnia Users dataset*. This dataset has been formed with the aim of drawing conclusions about the activity of users with insomnia on a social network such as Twitter. This dataset is formed by 103 users with insomnia based on the *Spanish Insomnia dataset* using the methodology described in Sect. V-C.

B. Geographic Research

The first study we did was a geographical distribution of the 54432 tweets we collected in the first capture. The objective of this study is to understand the importance of insomnia in Spanish-speaking countries.

The location information of a tweet can be extracted in two different ways: (i) the tweet can be geolocated directly by the platform (users can optionally choose to allow their tweets to be geolocated using a GPS system) (ii) using the "Location" field associated with the tweet (Twitter API also provides information about the user who has written the tweet).

When analysing our *General Insomnia dataset*, we determined that only 1% of the tweets contained information about the location from which they were published. For this reason, we carried out the analysis of the location fields of the users who had written the rest of the tweets. To do this we use the Google Maps Geocoding API³ which allows us to convert this field in text to geographic coordinates, and so be able to work with the locations in a more effective way than with texts [15].

With this procedure, we obtained 32404 geolocated tweets that is 59% of our sample.

For the part of visualizing the geolocation of the tweets we have used the Python GeoPandas [16] library that allows us to represent the geolocalized tweets in their corresponding country, as shown in Fig. 2.

TABLE I
TOP COUNTRIES IN TWEETS OF THE GENERAL INSOMNIA DATASET.

Country	Tweets	Percentage
Argentina	11132	34.53%
Mexico	5349	16.59%
Spain	5112	15.85%
United States	3052	9.46%
Colombia	1884	5.84%

³<https://cloud.google.com/maps-platform/>

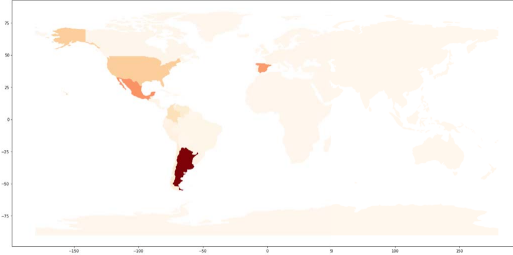


Fig. 2. Origin countries of the General Insomnia dataset.

We have also highlighted the countries with more tweets with the word insomnia as shown in Table I, in which it is important to highlight that Argentina, Mexico and Spain have a much greater presence than the rest of the Spanish-speaking countries.

Specifically, in the case of Spain, we want to determine the importance of insomnia in the Twitter user community.

To do this we have determined the percentage of users who have posted at least one tweet in a given period with the word insomnia. This percentage is given by the number of different users that we have obtained from our dataset from Spain among the total number of users that have written in that country during the time of capture that has been three weeks.

The number of different users who have posted in the three weeks of capture at least one tweet about insomnia is 1982. This number has to be compared with the number of active Spanish users in the period of three weeks who have the location field filled in, as our sample only includes users who have an insomnia tweet and the location field filled in. The number of users in Spain who have a written localization field is 2.9% [17].

With this number and the data that there are 7.52 million active Twitter users (they write at least one tweet) for a month in Spain⁴, it can be obtained that there are 164124 active users with the location filled in during the time of capture.

With these three facts, we can determine that approximately 1.21% of users in Spain have ever written about insomnia.

C. Labelling

Our goal is to detect and characterize tweets from users who claim to have insomnia as a supervised classification task and two text classifiers was trained to perform this task.

The *Insomnia Tweets* dataset has been labelled with the *Insomnia* label and the *Theme Tweets* dataset has been labelled with the *Themes* and *Symptoms* labels.

1) *Insomnia*: The first labelling we did was in *Insomnia Tweets* dataset and it consisted of differentiating between tweets that contain the word insomnia and come from a person who claims to have insomnia and those who do not.

⁴<https://es.statista.com/>

Specifically, this tag is added to tweets that match these characteristics:

- They do not come from a corporate account, that is, the user who has written the tweet is a natural person.
- In the tweet the user informs that it is the user who suffers from insomnia, so there are no valid tweets where it is transmitted that an acquaintance suffers from insomnia for example.

It is important to remember that none of the tweets that form the dataset are re-tweets, a filter that was implemented in the capture process. The Insomnia Tweets dataset distribution is shown in Table. II

TABLE II
DISTRIBUTION OF INSOMNIA TWEETS DATASET AND AVERAGE NUMBER OF WORDS

Insomnia	N (%)	Avg. no. words	Example
yes	351 (69)	12.57	I have insomnia again
no	160 (31)	11.61	Do you have insomnia?

The average number of words has been included in order to provide information on the size of the dataset. For this average we have considered all the words contained in a tweet, including stop words

2) *Themes*: The second labelling was done in the *Theme Tweets* dataset and it aims to group tweets by type of information, allowing us to better characterize the messages and thus draw better conclusions. We remember that this dataset is the one we will use to train the *Theme Classifier*. The topics we defined were:

- User expresses only his sleep disorder.
- User expresses his sleep disorder and gives the causes of his problem.
- User expresses his or her disorder and requests help.
- User expresses his sleep disorder and transmits what he does at night.
- User expresses his/her sleep disorder and takes some action to try to solve it.

The distribution of this label in *Theme Tweets* dataset is shown in Table III.

TABLE III
DISTRIBUTION OF THEME LABELS IN THEME TWEETS DATASET AND AVERAGE NUMBER OF WORDS

Theme	N (%)	Avg. no. words
Only sleep disorder	214 (61)	10.6
Sleep disorder and cause	34 (9.7)	17.11
Sleep disorder and requests help	28 (8)	11.9
Sleep disorder and current activity	41 (11.6)	16.22
Sleep disorder and current therapy	34 (9.7)	16.6

The most worrying fact is that only 9.7% say they take some action to solve it. This percentage shows that there is a need to develop techniques for the detection of sleep disorders such as insomnia in order to help and diagnose those affected and thus increase the number of people who take some remedy.

3) *Symptoms*: As mentioned above, we are interested in knowing what are the different symptoms of tweets for users with insomnia. In the third edition of International Criteria for Sleep Disorders (ICSD-3), the main symptoms of insomnia were defined as: difficulty in starting sleep; short sleep duration; difficulty in sleeping and low energy during the day.

As in the case of *Theme* labels, we have labelled *Theme Tweets* dataset in each of the three defined symptoms. The distribution of this label in *Theme Tweets* dataset is shown in Table IV.

TABLE IV
DISTRIBUTION OF SYMPTOMS LABEL AND AVERAGE NUMBER OF WORDS IN THEME TWEETS DATASET

Symptom	N (%)	Avg. no. words
Difficulty in starting sleep	32 (66.7)	15
Short sleep duration	9 (18.7)	13.5
Difficulty in sleeping and low energy during the next day	7 (14.6)	20.7

The first conclusion we have drawn from of this labelling was that most of the tweets did not express any of the defined symptoms. Specifically, only 14% (48) of the posts commented on any of the symptoms. The small amount of tweets with this information suggests that the population does not analyse how they suffer these kinds of disorders.

IV. TEXT MINING TECHNIQUES

With the aim of automating the annotation task, two text classifiers have been developed: (i) one that performs the task of tagging tweets that meet the characteristics of the insomnia tag; and another (ii) that classifies the tweets among the defined themes. Regarding the implementation, Python libraries *scikit-learn* [18] and *NLTK* [19] have been used; for orchestration and providing the system with scalability capabilities, we have used Luigi ⁵.

As for feature extraction methods, we separate these into two categories: statistical and semantic methods. In this section, the developed steps –preprocessing, feature extraction (both statistical and semantic), and classification– are described.

A. Preprocessing

In this phase the raw text is taken and cleaned using common NLP techniques [20]: elimination of punctuation marks, stop-words, rare characters, URLs, etc. Besides, the tokenization has been done oriented to the domain, since Twitter has specific characteristics [21] that need to be addressed: user mentions are normalized (e.g., @potus is transformed to *user*), and appearance of hashtags is flagged. Finally, tokens are stemmed using the Porter method.

B. Statistical feature extraction

Regarding the dataset-specific statistic features, we consider TF-IDF weighting over n-grams. In this way, these features encode information regarding the presence of n-grams, adding

⁵<https://github.com/spotify/luigi/>

frequency statistics of those same n-grams. After performing a preliminary evaluation, we have found that using n-grams with n ranging from 1 to 3 (1-grams, 2-grams and 3-grams) yields the better results. To a more extent description of the TF-IDF measure, we refer to the reader to [22].

We also consider the use of Latent Dirichlet Allocation (LDA) features as a baseline. This algorithm considers each document as a mixture of several topics, and each word in the document belongs to one of the topics in the document. LDA works by calculating the probability that a document belongs to a topic, based on the probability of each of the words that make up the document [23].

In addition, the Labeled LDA (L-LDA) feature extraction algorithm is used. In contrast to LDA [23], which is a modeling algorithm that discovers textual topics in an unsupervised manner, L-LDA is a variation of the LDA algorithm that models topics in a supervised setting. It does not only indicate the number of topics that are latent in a set of documents, but it does so by using the topic labels [24].

The L-LDA method works as follows. Firstly, we used the Stanford Topic Modelling Toolbox (STMT) [25] tool which generates a dictionary for each tagged topic composed of all the words in the dataset and the probability that each of those words belongs to that particular topic. As a result, the algorithm is able of estimating the probabilities that describe if a certain document belongs to each of the topics.

C. Semantic feature extraction

This paper presents a feature extraction method that uses the information contained in a word embedding model and a selection of relevant words to represent a document. This method makes use of the similarity measure that can be done in a word embedding model via cosine similarity.

In this way, we consider a selection of words S that serves as a “template” to which project input documents. Given an input document (e.g., tweet), the similarity between the input word vectors of that document and each of the words in S is computed. This can be done because the embedding model maps each word to a vector in a n -dimensional vector space. For an input word the method computes a similarity vector; when done for the whole document, a matrix of dimensions $m \times |S|$ is obtained, where m is the number of input words in a certain document. In order to reduce the dimensionality to a feature vector, the maximum is computed column-wise. The final dimensionality of the extracted feature vector is $|S|$, that is, the number of selected words.

The selection of relevant words is independent to the method, and can be implemented in a number of ways. This work tackles two different approaches for this word selection.

- Firstly, the words of a sentiment lexicon are used. The idea is that insomnia users probably show a bias towards negative sentiment when complaining about insomnia, while users that do not suffer insomnia may not complain, and thus do not use negative sentiment. For this, we use the Spanish sentiment lexicon *ElhPolar* [26].

- Secondly, domain-wise words have been extracted. For this, most frequent words have been collected from our dataset. Please note that these words have been extracted after the preprocessing steps (Sect. IV-A).

The proposed feature extraction method can be seen as a projection of the input text to a selection of words. Generated feature vectors are dense, and its dimensionality depends on the number of selected words; this value is set to 100 words with a parameter exploration. This method is, to the best of our knowledge, a novel work.

D. Classification

Finally, as a last step in our data processing pipeline, a machine learning classifier is used. This work tackles two different classification tasks, namely insomnia detection and theme classification. Thus, we use two different machine learning systems to address these tasks.

With respect to the methodology of the evaluation, all experiments have been done with cross validation. More specifically, we have used the k -fold configuration with a $k = 5$. Besides, accuracy and F1-score metrics have been used to measure the performance of the learners.

There are many options among machine learning models that can be used. In this project, as done in [12], we have evaluated the use of the following classification algorithms: Multinomial Naive Bayes, Support Vector Classification (SVC), K-Nearest Neighbors (KNN), Logistic Regression, Random Forest Classifier. The hyperparameters of each of these algorithms have been tuned using a cross-validation grid search. In a preliminary evaluation, we have found that Logistic Regression yields better results in the two classification tasks at hand.

V. RESULTS

A. Insomnia Classifier

For the task of determining whether a tweet has been written by an user that suffers insomnia or not (sect. III-C), a machine learning classifier has been trained. As we have already mentioned, we have used the *Insomnia Tweets* dataset to train this classifier. Consequently, to facilitate this training, we have balanced the dataset with under-sampling, resulting in a total of 320 tweets. In this task, the feature extraction methods used are: TF-IDF combined with LDA, and semantic similarity (Sect. IV-C) with the two variations considered: using the sentiment lexicon or a domain-wise word selection.

TABLE V
EVALUATION RESULTS IN INSOMNIA DETECTION.

Method	Accuracy	F1-Score
TF-IDF + LDA	0.84	0.82
Semantic Similarity (lexicon words)	0.85	0.85
Semantic Similarity (domain words)	0.89	0.90

Table V shows the results for the evaluation in this task. It can be seen that the semantic similarity features perform, by a large margin, better than the ones extracted by the TF-IDF + LDA combination. Also, using a domain-adapted word

selection leads to further improvement, reaching a 90% of F1-score. This result indicates that the proposed feature extraction methods yields good performances. Also, the fact that using a domain-adapted selection increases the performance suggests that our method can be easily used for other text categorization tasks just by changing the selection of words.

B. Theme Classifier

Following, the task of determining the *Theme* label (Sect. III-C) of a message is performed similarly as in Sect. V-A. As described in Table III, the messages are distributed along five categories that represent the themes. From this distribution, it can be seen that there is a considerable disproportion of tweets among the aforementioned themes. Therefore, and to avoid problems of unbalanced data, a sample subset of 172 tweets from the *Theme Tweets* dataset has been selected, so that the categories distribution was approximately uniform for each theme.

Regarding the feature extraction, we have evaluated the use of TF-IDF representations and, due to the nature of the problem, the L-LDA feature extraction method (Sect. IV). As described, L-LDA is able to effectively learn topics by means of annotations, characteristic that is exploited in this task.

TABLE VI
EVALUATION RESULTS IN THEME DETECTION.

Method	Accuracy	F1-Score
TF-IDF	0.57	0.58
TF-IDF + LDA	0.51	0.52
TF-IDF + L-LDA	0.78	0.78

The obtained results are shown in Table VI, where different feature extraction methods are evaluated. It can be seen that TF-IDF and LDA features do not lead to a good performance in this task. This suggests that the information contained in such features is not useful for the task of theme detection. Nevertheless, the evaluation shows that using L-LDA method highly improves the performance of the system, reaching a 0.78 in the F-score. In light of these results, it is safe to assume that such improvement is due to the use of the L-LDA method. Since the task of theme detection involves, at least to some extent, topic detection, we hypothesize that the features extracted by a supervised topic detection method are of use.

C. Social Network activity

As we have commented in the Sect. III-A, we have made a capture of tweets with the word insomnia from Spain. The aim of this capture was to form the *Spanish Insomnia Users* dataset to be able to determine, through their timeline, a pattern of activity on a social network such as Twitter of people suffering from insomnia, and compare it with the activity of a set of normal users.

The process of selecting a user for the Sleep Group (set of users with insomnia) is described in the Fig. 3.

Basically, this process consisted of evaluating each of the tweets captured with the *Insomnia Classifier*, so that only the tweets of users who claim to have insomnia would follow

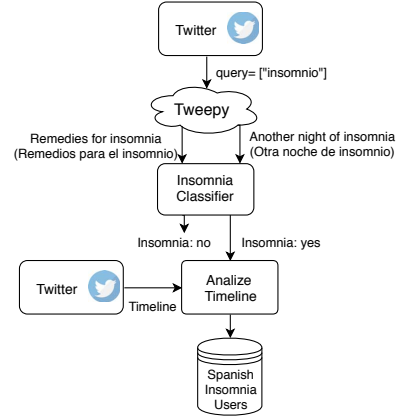


Fig. 3. Process to elaborate the dataset of users.

in the process. After this first filter, we analysed the last 200 tweets of the account associated with each tweet; if in these 200 tweets there were 2 or more tweets in which the user mentioned the word "insomnia"(insomnio) or "I can't sleep"(no puedo dormir), that user became part of the *Spanish Insomnia Users* dataset.

As we have already mentioned we want to determine a pattern of activity of users with insomnia and normal users, so we also perform a user dataset without insomnia (Normal Group). To form this group we made a random capture of tweets from Spain, and hand filtered to have only tweets from people and thus not have in the study tweets from accounts from some kind of organisation. We also analysed the timeline of these users to make sure they didn't have any insomnia-related tweets.

The first feature that we focused on was the number of tweets related to insomnia that each of the *Spanish Insomnia Users* had (Fig. 5). We can observe that most of the users have two relatively recent messages related to insomnia (72%).

After that, we analyse the time associated to all the tweets of the timeline of each one of the users of the two groups. In the Fig. 4 we have plotted the percentage of tweets per hour of average that the users of each of the groups post.

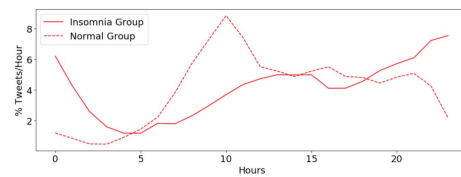


Fig. 4. Activity on Twitter per hour in users with and without insomnia.

It can be seen that users who belong to the *Spanish Insomnia Users* dataset have a greater activity throughout the night due

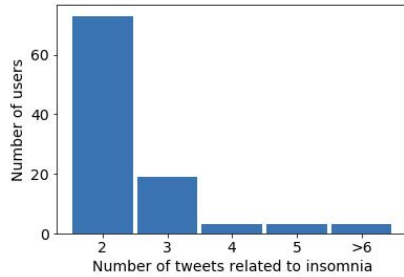


Fig. 5. Distribution of the Spanish Insomnia Users dataset according to number of Insomnia tweets on their timeline.

to the inability to fall asleep. During the morning, the activity of these users may decrease due to nighttime fatigue and in exchange the Normal Group has its highest point of activity around 10 a.m. In the afternoon, the activity is balanced, until the evening, when the users with insomnia again show more activity.

An online service ⁶ has been developed with the aim of being able to analyse Twitter messages about insomnia. The interface of this service is shown in Fig. 6. In addition to the analysis we have carried out in this study, this system also allows us to analyse both the sentiments and the emotions transmitted by users with insomnia through the Senpy [27] tool.

VI. CONCLUSIONS AND OUTLOOK

This research has been carried out through messages related to insomnia published on the social network Twitter with the aim of obtaining information on how this disorder was represented in the users of this social network. Specifically, it has been done through Spanish-speaking posts and users that, as far as we know, is something that has not been done previously and that, due to the large number of people who speak this language, in our opinion was very necessary.

In relation to the proposed objectives, we have characterized the presence of insomnia in the main Spanish-speaking countries and, specifically in Spain, we have been able to quantify the presence of insomnia in its community of users in which we have determined that approximately 1.21% (corresponding to 54450 users) write at least one tweet per month related to insomnia. This conclusion determines that insomnia is a disease that is very present in our society today.

Through the analysis of the posts related to insomnia we have been able to characterize the way in which users transmit this disorder, however the conclusions we have been able to draw are limited by the size of our dataset and by the number of tweets we have from some of the themes. As we have mentioned, 61% of tweets only give information that the user suffers from insomnia, this makes it difficult for us to

⁶<http://dreampy.cluster.gsi.dit.upm.es/>

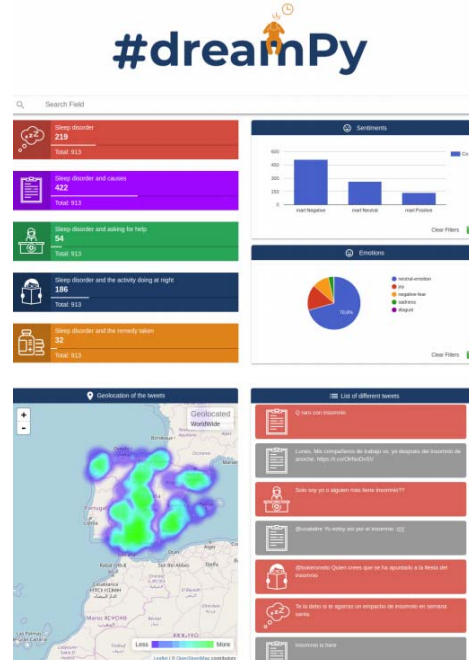


Fig. 6. Dashboard of the monitoring service

determine a characteristic that is common to the context in which all tweets about insomnia are found.

However, the most worrying thing is that only 9.6% of the tweets analysed contained information about some kind of measure or treatment that users perform to try to solve their problem with insomnia. The conclusion we draw from this fact is that the majority of users do not give the necessary importance to insomnia, which supports our idea that studies and tools are needed to understand and diagnose this disorder more effectively.

Regarding the users that form the *Spanish Insomnia Users* dataset, we have been able to determine that these users have a significantly higher activity posting tweets during the night hours regarding the activity of a normal user group. In addition, this dataset has been captured using the *Insomnia Classifier* developed in the project, so seeing the results we have just commented on the night activity, we can determine the correct functioning of this classifier when determining tweets about insomnia.

In addition to this, this paper explores the possibilities of applying machine learning and NLP techniques to the task of determining whether a Twitter message has been written

by an user that suffers insomnia or not. In this sense, this work proposes a feature extraction mechanism that exploits the similarity measure that can be computed in a word embedding model. Moreover, this approach can be adapted to a certain domain by means of tuning word selection processes. This feature extraction method has been evaluated using the captured dataset. Obtained results highly indicate that this method can yield high performance in this task, and that domain adaptation can further improve these results. This suggest that the proposed method can be generalized to different domains or even NLP tasks.

As we have already mentioned, from the analysis of the tweets on insomnia we have been able to conclude that the number of tweets that contain relevant information for the study of this phenomenon is small. Therefore, we consider that the Insomnia Monitoring System developed is another great contribution to the research and development of systems for the analysis of insomnia as it allows the capture and analysis of only the data that interest us in an automated way.

Our future work focuses on a more detailed study of the users that form the *Spanish Insomnia Users* dataset. So far, we have focused the entire study on posts from users who already claim to have insomnia. Therefore, our goal will be to conduct a more detailed study on the timeline of all Spanish insomnia users. Specifically, we want to study the change of emotions and feelings that these users have about time and try to determine patterns that allow us to investigate for the prediction of users who may suffer insomnia in the future.

ACKNOWLEDGEMENTS

This research work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D projects SEMOLA (TEC2015-68284-R) and ITEA3 EmoSpaces (ITEA3-14012/RTC-2016-5053-7) and for the project ITEA3 SoMeDi (ITEA3-15011).

REFERENCES

- [1] D Leger, B Poursain, D Neubauer, and M Uchiyama. An international survey of sleeping problems in the general population. *Current medical research and opinion*, 24(1):307–317, 2008.
- [2] Stacey J Elder, Ronald L Pisoni, Tadao Akizawa, Rachel Fissell, Vittorio E Andreucci, Shunichi Fukuhara, Kiyoshi Kurokawa, Hugh C Rayner, Anna L Furniss, Friedrich K Port, et al. Sleep quality predicts quality of life and mortality risk in haemodialysis patients: results from the dialysis outcomes and practice patterns study (DOPPS). *Nephrology Dialysis Transplantation*, 23(3):998–1004, 2007.
- [3] Michael J Sateia. International classification of sleep disorders. *Chest*, 146(5):1387–1394, 2014.
- [4] Lucia Falzon, Caitlin McCurrie, and John Dunn. Representation and analysis of twitter activity: A dynamic network perspective. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, pages 1183–1190. ACM, 2017.
- [5] Elizabeth M Seabrook, Margaret L Kern, Ben D Fulcher, and Nikki S Rickard. Predicting depression from language-based emotion dynamics: Longitudinal analysis of facebook and twitter status updates. *Journal of medical Internet research*, 20(5):e168, 2018.
- [6] Amayas Abboute, Yasser Boudjeriou, Gilles Entringer, Jérôme Azé, Sandra Bringay, and Pascal Poncelet. Mining twitter for suicide prevention. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 250–253. Springer, 2014.
- [7] Sherry Pagoto, Kristin L Schneider, Martinus Evans, Molly E Waring, Brad Appelhans, Andrew M Busch, Matthew C Whited, Herpreet Thind, and Michelle Ziedonis. Tweeting it off: characteristics of adults who tweet about a weight loss attempt. *Journal of the American Medical Informatics Association*, 21(6):1032–1037, 2014.
- [8] Sanne Kruikemeier. How political candidates use twitter and the impact on votes. *Computers in Human Behavior*, 34:131–139, 2014.
- [9] Shintaro Okazaki, Ana M Díaz-Martín, Mercedes Rozano, and Héctor David Menéndez-Benito. Using twitter to engage with customers: A data mining approach. *Internet Research*, 25(3):416–434, 2015.
- [10] Seyed-Mehdi-Reza Beheshti, Alireza Tabebordbar, Boualem Benatallah, and Reza Nouri. On automating basic data curation tasks. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 165–169. International World Wide Web Conferences Steering Committee, 2017.
- [11] Sue Jamison-Powell, Conor Linehan, Laura Daley, Andrew Garbett, and Shaun Lawson. I can't get no sleep: discussing# insomnia on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1501–1510. ACM, 2012.
- [12] Xianyun Tian, Guang Yu, and Fang He. An analysis of sleep complaints on Sina Weibo. *Computers in Human Behavior*, 62:230–235, 2016.
- [13] David J McIver, Jared B Hawkins, Rumi Chunara, Arnaub K Chatterjee, Aman Bhandari, Timothy P Fitzgerald, Sachin H Jain, and John S Brownstein. Characterizing sleep issues using twitter. *Journal of medical Internet research*, 17(6), 2015.
- [14] Joshua Roesslein. tweepy documentation. [Online] <http://tweepy.readthedocs.io/en/v3.5.0/>, 5, 2009.
- [15] Gabriel Svennerberg. *Beginning Google Maps API 3*. Apress, 2010.
- [16] K Jordahl. Geopandas: Python tools for geographic data. URL: <https://github.com/geopandas/geopandas>, 2014.
- [17] Alejandro Llorente, Manuel Garcia-Herranz, Manuel Cebrian, and Esteban Moro. Social media fingerprints of unemployment. *PloS one*, 10(5):e0128692, 2015.
- [18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.
- [19] Steven Bird and Edward Loper. Nltk: the natural language toolkit. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 31. Association for Computational Linguistics, 2004.
- [20] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [21] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsn*, 11(538-541):164, 2011.
- [22] Ray R Larson. Introduction to information retrieval. *Journal of the American Society for Information Science and Technology*, 61(4):852–853, 2010.
- [23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [24] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.
- [25] D Ramage and E Rosen. Stanford topic modeling toolbox (sttm), 2009.
- [26] X Saralegi Urizar and I San Vicente Roncal. Elhuyar at tass 2013. In *Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013)*, pages 143–150, 2013.
- [27] J Fernando Sánchez-Rada, Carlos A Iglesias, Ignacio Corcuera, and Oscar Araque. Senpy: A pragmatic linked sentiment analysis framework. In *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*, pages 735–742. IEEE, 2016.

A.2.5 Neural Domain Adaptation of Sentiment Lexicons

Title	Neural Domain Adaptation of Sentiment Lexicons
Authors	Araque, Oscar and Guerini, Marco and Strapparava, Carlo and Iglesias, Carlos A.
Proceedings	2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)
ISBN	978-1-5386-0680-3
Year	2017
Keywords	Sentiment analysis, natural language processing, Semantics, Task analysis, Adaptation models, completely data-driven domain characterization metric, computational linguistics, domain context, domain-adapted sentiment lexicons, domain-specific traits, learning (artificial intelligence), Measurement, neural domain adaptation, posterior sentiment analysis tasks, semantic domain, sentiment lexica, sentiment lexicon, subjective sentimental knowledge, Training, Transforms
Pages	105–110
DOI	https://doi.org/10.1109/ACIIW.2017.8272598
Online	https://ieeexplore.ieee.org/document/8272598
Abstract	Sentiment lexicons are widely used in computational linguistics, as they represent a resource that directly contains subjective sentimental knowledge. Usually these sentiment lexica are generic and developed without any specific semantic domain in mind. Nonetheless, the domain context can be highly relevant for Sentiment Analysis, as it is known that word polarities can be influenced by domain-specific traits. This paper studies the problem of automatically generating domain-adapted sentiment lexicons that can be used in posterior Sentiment Analysis tasks. We propose a neural network approach that modifies a sentiment lexicon using distantly annotated text of a certain domain. Additionally, we present a completely data-driven domain characterization metric that measures the centrality of a set of documents. Experimental work shows that this metric offers a measure of the generated lexicons' quality. Also, it is shown that the generated lexicons yield higher performance on domain-oriented Sentiment Analysis than a generic lexicon and other known baselines. Finally, it is also discussed that these extracted lexicons can be used for Sentiment Analysis even for approaches with no learning capabilities.

Neural Domain Adaptation of Sentiment Lexicons

Oscar Araque*, Marco Guerini†, Carlo Strapparava† and Carlos A. Iglesias*

*Grupo de Sistemas Inteligentes, Departamento de Ingeniería Telemática
Universidad Politécnica de Madrid, E.T.S.I. de Telecomunicación, Avda. Complutense, 30, Madrid, Spain

o.araque@upm.es, carlosangel.iglesias@upm.es

†FBK-Irst, Via Sommarive 18, Trento, Italy
guerini@fbk.eu, strappa@fbk.eu

Abstract—Sentiment lexicons are widely used in computational linguistics, as they represent a resource that directly contains subjective sentimental knowledge. Usually these sentiment lexica are generic and developed without any specific semantic domain in mind. Nonetheless, the domain context can be highly relevant for sentiment analysis, as it is known that word polarities can be influenced by domain-specific traits. This paper studies the problem of automatically generating domain-adapted sentiment lexicons that can be used in posterior sentiment analysis tasks. We propose a neural network approach that modifies a sentiment lexicon using distantly annotated text of a certain domain. Additionally, we present a completely data-driven domain characterization metric that measures the centrality of a set of documents. Experimental work shows that this metric offers a measure of the generated lexicons' quality. Also, it is shown that the generated lexicons yield higher performance on domain-oriented sentiment analysis than a generic lexicon and other known baselines. Finally, it is also discussed that these extracted lexicons can be used for sentiment analysis even for approaches with no learning capabilities.

1. Introduction

Sentiment Analysis (SA) has become a popular research area in recent years due to its applicability to a wide range of problems, the growth and accessibility of on-line opinions and its commercial interest [1]. For instance, automatic analysis of product reviews are of special interests to a great variety of companies, as it allows them to have a measure on their customers' opinions towards their stocks. Other areas where Sentiment Analysis has been applied are movie reviews [2], congressional floor debates [3], news and blogs [4], among others. This variety of data sources and target domains make it difficult to construct a generic sentiment classifier. In this context, it is specially relevant the role that the domain plays in the Sentiment Analysis process. The sentiment associated with, for example, a product review can be greatly affected by the domain to which it belongs. In this sense, sentiment associated to different words can vary in its intensity or even shift its polarity value altogether (e.g. 'big' when describing a monitor screen usually associates

with a positive orientation, while can have a negative sense when referring to clothes).

In relation to this, many sentiment classification tasks make use of opinion lexicons [5]. An opinion lexicon is a collection of words that have associated sentiment polarity values that can be continuous range of numbers or simpler polarity labels. There are many prior polarity lexica (where generic sentiment scores are associated out of context), such as [6], [7] and [8]. However, using a generic lexicon for a domain-specific sentiment task does not yield the best results because, as discussed, words can vary in their sentiment values. On top of this, it is known that words sentiment is domain dependent [9]. For these reasons, the task of automatically generating posterior polarity (domain-adapted) lexicons in a flexible manner is an important research problem.

In this work, we propose a neural network based method for automatically extracting domain-adapted sentiment lexicons. This model makes use of distantly annotated data belonging to a certain domain in order to compile a human-readable sentiment lexicon that can be used for posterior sentiment analysis on that domain. Also, this work introduces a novel metric that characterizes certain features of a set of domain-oriented documents. We show that this metric offers a way of predicting the effectiveness of the proposed model.

The basic idea for the proposed metric is to estimate the centrality of a set of documents, so one can have a measure of the performance of the proposed model before any training. We define the centrality as the measure of how specific is a given set of documents to a variety of topics. The experiments show that the introduced metric gives a clear sense of the quality of the extracted lexicon by means of our model.

As for the proposed lexicon adaptation model, its main characteristic is that it uses the back-propagation algorithm [10] to change the lexicon values. This model makes use of the loss signal of a sentiment classification task to better adjust those values, finally generating the desired domain-adapted sentiment lexicon, which we call SEDLex (SEntiment adapted to the Domain Lexicon).

The rest of the paper is organized as follows. Section 2 shows previous work on lexicon domain adaptation. In Sec-

tion 3, the proposed domain characterization metric is described, as well its relation with the proposed model, which is explained in Section 4. After this, the paper continues with an overview on the experimental setup in Section 5. Section 6 follows, illustrating the results obtained in the experiments, as well as the validation of the proposed techniques. Finally, Section 7 draws the conclusions obtained, and outlines the future work.

2. Related Work

Many works have been oriented to domain adaptation of sentiment lexicons. As depicted in [5], three main techniques have been explored for generating opinion lexicons: (i) manual approach, (ii) dictionary-based approach and (iii) corpus-based approach. The manual approach is very time-consuming, and thus it is normally combined with the other two, that rely on automatic methods. Many dictionary-based methods consist in bootstrapping from a small set of opinionated words, and using them as seed for the search of similar terms in a known dictionary such as WordNet [11] or SentiWordNet [7]. In order to improve the generated lexicons researchers have used machine learning techniques [12], as well as some external sources of knowledge, such as the definition of WordNet words (glosses) [13]. Nevertheless, as explained in [5], the dictionary-based approaches are not able to obtain domain specific orientations. In this regard, corpus-based techniques can alleviate the problem. This last type of lexicon generation technique relies on co-occurrence patterns detected on a large corpus, as well as a seed set of opinion words to locate opinionated words in the corpus [5].

There is a great variety of statistical information extraction methods for generation of domain-oriented lexicons. Nevertheless, some works share some common methodology, as is the case of *optimization-based* approaches. For instance, the work shown in [14] proposes an information bottleneck problem that blends cross and within-domain knowledge. This problem is then solved with an iterative optimization method that, upon convergence, obtains the desired lexicon. A different approach is explored in [15], where a linear programming technique is aimed to directly reflect the characteristics of a certain domain. This method exploits the relation among words and opinion expressions to compute the most likely polarity value in that domain. Another explored technique is presented in [16], in which several opinion signals are extracted from an unlabeled opinionated text collection. These signals are combined into an optimization problem, and then transformed to a linear programming formulation by means of a mathematical transformation. In this way, the method can learn the sentiment of words and aspects.

Another common technique used when generating domain-specific lexicons is *label propagation*. In this line of work, the technique explained in [17] introduces a method called double propagation. This method consists in several extraction rules that are designed based on dependency trees, and exploit the relations between modifier sentiment words

and their associated topics, as well as the sentiment and topics words. A similar research [18] proposes an automatic construction strategy based on constrained label propagation. The strategy firstly extracts candidate sentiment terms, then it uses domain and morphological constraints to spread to the entire set of candidates, improving the lexicon. Label propagation can be also applied onto different structures, such as a lexical graph. This approach is taken in [19], where a set of high-quality word embeddings are transformed into a lexical graph by connecting each word with its nearest neighbors in the semantic space. Then, the sentiment labels are propagated from a set of seed words through the graph using a random walk algorithm.

Lastly, there are other methods that directly extract statistical information from corpora, and transform this knowledge to a domain-adapted sentiment lexicon. The method described in [20] uses frequency information to assign different weights to positive and negative words in the domain of movie reviews. Another example of such approaches is [21], which uses the *TF-IDF* score for each sentiment polarity to adapt an existing general lexicon to a specific domain. Also, the works described in [22] and [23] describe the *Compositional Semantics* methods that, parting from a frequency measure of the words in the corpus and a set of emotion labels (this could be used in sentiment as well), builds a lexicon from a word-emotion matrix.

3. Domain characterization

Understanding what a domain is in the context of sentiment analysis can be beneficial for domain adaptation applications. In the context of this work, properly defining a domain can substantiate the lexicon adaptation process. Considering an approach that tries to adapt a certain lexicon to a specific domain, we would like to know some characteristics of the data before any training, so we can have a measure of how well the adaptation is going to perform.

In this work, we tackle this problem by proposing a data-driven metric that aims to characterize a set of documents D . The proposed metric aims to estimate the semantic centrality of D . That is, how specific to a certain set of topics T the dataset is. As an example, we can consider a dataset that describes some products of the electronics domain. If the majority of the text in that dataset refers to certain recurrent topics (phones, batteries, screens, computers, ...), then it is possibly a centered domain. On the contrary, a dataset that includes a large set of topics has a lower centrality.

Following this, we propose a metric m that measures the centrality of D using a semantic structure. In this work, a word embeddings model is used. It is known that word embeddings retain a semantic structure that can be exploited in a variety of applications [24]. The proposed method aims to leverage the information embedded in this representation for domain characterization. The idea is to measure the distances between a set of selected topics W from D . These measures retain information of how disperse W is on the vector space, and thus the extent of centrality of D . Figure 1

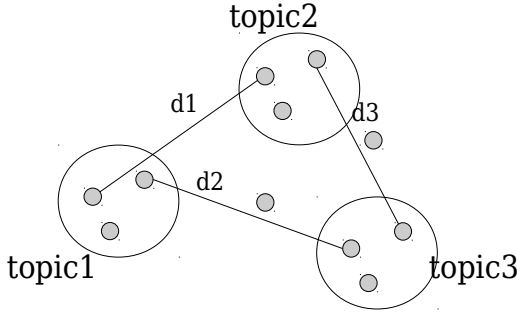


Figure 1. Distances measures between words from the set W .

shows a graphical representation of the measures in relation to a set of topics. We define the metric as:

$$m = \frac{1}{|W|} \sum_{i,j}^{i,j \in W} d(v_i, v_j)$$

being $|W|$ the number of elements in W . The distance between word vectors v_i and v_j can be expressed as follows, $\|\cdot\|_2$ being the euclidean norm.

$$d(v_i, v_j) = \frac{v_i v_j}{\|v_i\|_2 \|v_j\|_2}$$

For extracting the set W from the dataset, a number of most common words is first selected, and filtered with the stopwords set from [25]. Then, the embedding vectors for those words are extracted, forming the V set. The metric is computed for all the possible pairs from V . If $|V|$ is a moderate amount (e.g. 300-600), the cost of computing all the possible combinations of two word vectors is not too high¹.

4. Lexicon Adaptation Model

4.1. Lexicon Generation

Automatically generating a domain-centered sentiment lexicon is an open research problem. This work considers the situation where we have access to distantly labeled textual data. We refer to distantly labeled data as the one that is not manually nor automatically annotated, but has some meta-data associated that can be used as sentiment label. In this work, we consider product reviews that have a meta-data value belonging to a 5-score system. From this score value, we extract a sentiment signal with which we train the proposed model. Section 5 further explains this process.

The proposed approach consists in a feedforward neural network (also called multilayer perceptron). The network is

¹. Note that the number of combinations is given by the expression $\binom{|V|}{2} = \frac{|V|!}{2!(|V|-2)!}$

trained from representations of a set of documents that have associated distant labels. These inputs are the lexicon values of the words contained in the document representation. That is, if we consider a set of k words, the values from a certain sentiment lexicon are extracted and used as representation for the network. Regarding the initialization of the sentiment lexicon values, the network uses an existing generic lexicon from which starts modifying.

Using the distant labels, the network is trained with a cross-entropy loss function on the task of classifying the sentiment of a certain document using the lexicon values. The key idea is that, at training time, the back-propagation algorithm can be used to let the sentiment error signal flow to the network weights, as well as to the lexicon values. In this way, by training the network to differentiate the sentiment of the documents, it is also modifying the lexicon values, which are a continue range of numeric values. Considering that the training documents are domain-oriented we can expect that, after training, the modified lexicon contains useful sentiment information with respect to the domain.

4.2. Lexicon Transformation

When training is finished, the modified values are extracted, forming a new domain-adapted sentiment lexicon. Nevertheless, the resulting lexicon does not normally have the expected distribution of a human-readable lexicon. That is, we would expect a sentiment lexicon to be bounded in a range such as $[-1, 1]$. Also, positive orientations should be represented by positive numbers, and the contrary for negative sentiment orientations. The modification of the lexicon values relies on the back-propagation algorithm, and this method has no inherent constraints on the distribution of a certain set of parameters. In order to transform the modified lexicon to one that exhibits a “human-readable” distribution, we define a transformation operation. More formally, let $\mathbf{a}$ be the vector that contains the lexicon values so that the value a_i represents the lexicon value for the word i of the vocabulary. We define the transformation operation as $g((\mathbf{w} * \mathbf{a}) + \mathbf{b})$, $*$ being the element-wise product and g the element-wise function $g(x) = \frac{x-a}{b-a} + c$ that transforms the values from one range $[a, b]$ to another $[c, d]$. Figure 2 illustrates how the transformation process operates.

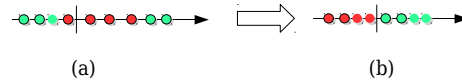


Figure 2. Graphical representation of polarity distribution of the modified lexicon (a) before the transformation process, and (b) after the transformation.

In order to obtain the parameters $\mathbf{w}$ and $\mathbf{b}$ we train a logistic regressor on a subset of the training data. The parameters of the defined operation and those of the input to the sigmoid function of the logistic regression are equal [26], so by training the learner we directly obtain the desired parameters. The extracted lexicon, that is, the one

TABLE 1. DATASETS STATISTICS

Name	#Reviews	#Windows
Toys and Games	2,252,771	8,060,106
Health and Personal Care	2,982,326	11,718,861
Movies and TV	4,607,047	27,042,309
Electronics	7,824,482	38,032,004
Books	22,507,155	138,422,556

resulting from the training of the neural network and the transformation process, can be directly used for sentiment analysis on the trained domain.

5. Experimental Setup

In order to evaluate the proposed approach, several experiments have been designed. We evaluate the effectiveness of the model through the sentiment analysis performance of the generated lexicons for each domain. For this reason, various known domain-oriented datasets have been used for training and evaluating the model.

5.1. Datasets

The datasets used for training are the Amazon Product Data [27], [28], which contain millions of product reviews extracted from Amazon. From these datasets, we have used the reviews text and their corresponding ratings, which are ranging from 1 to 5. For extracting the polarity labels the ratings are mapped to positive and negative values, being 1 and 2 a negative value; while 4 and 5 are transformed as a positive polarity. Documents with ratings of 3 are discarded. The selected domains and their basic statistics are shown in Table 1. The test dataset is the Multi Domain Sentiment Dataset v2 [29], which contains a set of product reviews already annotated. For the evaluation, we selected the same domain for the training and test sets, for each domain, as the test set is also divided domain-wise.

5.2. Training Instances Generation

The training examples that are fed to the proposed model are not directly text documents, but rather fixed windows of tokens. For obtaining these windows, each document is processed independently. Firstly, the set of words that constitute each document are intersected with the defined vocabulary (e.g. the vocabulary of the original lexicon). In this way, the words that contain sentiment information are selected. This intersection of words is then downsampled by randomly selecting a fraction of the length of the intersection set. Then, for each target word w_t in the downsampled set we extract a symmetric fixed window containing the words around w_t . If the window boundaries lie outside of the document, zero-padding is applied. As for the sentiment label, the window is assigned a label that is equal to that of the original document. Finally, the set of windows obtained are the training instances that correspond to the given document. In this way, we aggregate all the training instances from all

the documents, obtaining more instances than reviews in the original dataset. The number of reviews and training instances are described in Table 1.

5.3. Evaluation Workflow

For the model evaluation, we use both the training and test sets. First, the proposed model is trained with the training windows extracted from the training data for a specific domain. The training dataset is divided into batches, and the network is allowed to update its weights when each batch has been propagated forward (Sec. 4.1). Following, the lexicon is transformed as explained in Sec. 4.2.

Periodically, the model's modified lexicon is extracted and its individual performance is evaluated on the test set of that same domain via 3-fold cross-validation. To this end, a logistic regression model is trained on sparse representations of the documents. These features are created by generating high-dimensionality sparse vectors, where each index element corresponds to a word in the vocabulary. The value of a element is 0 if the word does not appear in the document, while the value is equal to the sentiment lexicon value for that word if it does appear. When the training is finished, the extracted lexicon is the one that has achieved better performance. In this way, the whole model is not evaluated, only the sentiment lexicon that has been generated by the model.

Apart from the accuracy, we also use the correlation between the values of the lexicon and the labels of the test set. This metric is used in order to have a fair comparison of the generated lexicons performance (i.e. without any syntactic or compositional reasoning that can boost the performance), as done in [8]. That is, for the words in a document of the test set, the lexicon values are extracted for those words, and the average is applied. These values are then compared to the labels of each document through the Pearson correlation.

5.4. Baselines

We use several baselines that are evaluated in the same way that the lexicons generated by our proposed model for comparison purposes. All methods have the same vocabulary for each domain. The most simple is the *random* baseline, in which the lexicon word values are randomly generated number from a normal distribution. The random baseline does not yield an accuracy of 50% due to that vocabulary information is introduced in the document representations. That is, if a word appears in a document the baseline has a random value in the corresponding index of the representation vector. If a word does not appear, a zero is used. In this way, the representation vectors are sparse and not completely random, which yields slightly better results than a full random approach.

Another baseline used in this work is the compositional semantics (CS) method [22], [23]. We evaluate three variants of this method: raw frequency counts (CS raw), normalized counts (CS norm), and TF-IDF frequency values (CS tfidf). Finally, the last baseline is the sentiment lexicon Sentiwords,

TABLE 2. ACCURACY ON THE TEST SET FOR BASELINES AND PROPOSED MODEL.

Model	Domain				
	Movies	Electronics	Health	Books	Games
Random	65.20	66.40	66.95	61.60	67.91
CS raw	79.75	80.20	81.45	79.35	80.65
CS norm	80.15	80.50	81.45	79.70	81.55
CS tfidf	70.75	80.80	81.85	80.75	80.90
Sentiwords	80.05	78.75	78.95	78.10	80.30
SEDLex-mod	81.60	80.65	81.71	79.40	84.05
SEDLex-trans	81.60	83.09	83.55	80.20	84.15

proposed in [8]. This represents a generic sentiment lexicon in our experiments, as it is not adapted to any domain.

6. Results

For the experiments, several hyper-parameters have been tested, and we report the parameters that yield the best results on a randomly selected dev set, drawn from the train set. For the neural model, a two-layer feedforward network is used with 52 and 25 units in each layer. Good results have been achieved with learning rates ranging from 0.0005 to 0.0001 with the Adam algorithm [30] as optimizer. During training the window is set to be symmetric with size 5, and the downsampling ratio used is 0.1. Interestingly, it has been observed that the transformation operation (Sec. 4.2) can be simplified by setting $b = 0$. This simplification further improves the performance results.

The first metric on which we evaluate the proposed model is the Accuracy on the test set. Table 2 shows the performance of the baselines and the proposed model. Our model is evaluated separately for the lexicon without transformation (SEDLex-mod, Sec. 4.1) and with transformation (SEDLex-trans, Sec. 4.2).

It can be seen that the best performing lexicon in almost all domains is the one generated by our full model. This result confirms the idea that the training process of a neural architecture on domain data can generate domain adapted lexicons. Also, as it is shown in the last two rows of Table 2, performing the proposed linear transformation further improves the effectiveness of the generated lexicon. We have confirmed that, after the transformation, positive word polarities are represented by positive values, and vice versa.

For the second evaluation metric, the correlation, Table 3 presents the associated results. It is interesting to see that after the transformation process the correlation is highly improved in comparison to that of the generated lexicon before the transformation. This is an indicative that the formulated transformation can correctly distribute the generated lexicon values as desired, positioning the positive words and the negative words in their corresponding numeric polarity.

Following, we evaluate how well the proposed metric m indicates the centrality of the datasets. For this end, we compute the Pearson correlation between the difference of accuracy score between the Sentiwords baseline and the

TABLE 3. CORRELATION BETWEEN SUM OF LEXICON VALUES AND TEST LABELS.

Model	Domain				
	Movies	Electronics	Health	Books	Games
Random	0.067	-0.060	-0.046	-0.016	0.031
CS raw	0.062	0.106	0.055	0.048	0.154
CS norm	0.069	0.127	0.074	0.051	0.172
CS tfidf	0.076	0.168	0.132	0.041	0.287
Sentiwords	0.153	0.147	0.199	0.083	0.152
SEDLex-mod	-0.085	0.167	0.119	0.029	0.207
SEDLex-trans	0.688	0.707	0.707	0.671	0.733

TABLE 4. DIFFERENCE IN ACCURACY METRIC BETWEEN PROPOSED MODEL AND BASELINE COMPARED TO m VALUES.

Domain	AccSEDLex-trans - AccChaseline	m value
Movies	1.55	0.1380
Electronics	4.34	0.1186
Health	4.60	0.1251
Books	2.10	0.1302
Games	3.85	0.1303

transformed lexicon for each domain, as Sentiwords is used for initializing the lexicon at training time. In this sense, the measure is understood as the gain in information for sentiment analysis. Table 4 shows the metric values computed from the domain datasets compared to that difference in accuracy. The correlation between these two columns is of **-0.8207**. This result indicates that there is a relation between the value of the proposed metric and the performance improvement of the generated sentiment lexicons.

7. Conclusions and Future Work

This paper proposes a sentiment lexicon domain adaptation framework based on a neural network architecture that, through the process of training for a straightforward sentiment classification task on domain data, modifies a lexicon in order to better contain domain information. The generation process is enhanced by a posterior transformation that provides the generated lexicon with a human-understandable distribution and range, following the majority of known sentiment lexicons. Besides, a domain characterization metric is presented that measures the centrality of a set of documents that relate to a given domain. It is shown that this metric strongly correlates with the improvement of the lexicons generated by the proposed model on a benchmark dataset in comparison to a known generic sentiment lexicon. It is argued that this metric can be used for measuring the effectiveness of the framework and the lexicons it generates provided a set of domain documents with a distant sentiment annotation.

Indeed, the experiments have shown that this model effectively adapts a sentiment lexicon to a certain domain, outperforming the performance of a linear classifier trained with only the lexicon as features with respect to a generic sentiment lexicon. Furthermore, the generated lexicons show

a strong correlation between their values and the sentiment labels of the test set. This allows the use of the generated lexicons for simple sentiment classification, without the need for any learning process.

As future work, we believe that extending this work to the emotion paradigm can be effectively done. Also, adding new sources of information to the model could further improve the lexicons quality. Finally, the proposed metric could be improved to a greater extent by adding new semantic structures to the measurement process.

Acknowledgments

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used in this research. This research work is partially supported through the projects Semola (TEC2015-68284-R), EmoSpaces (RTC-2016-5053-7), MOSI-AGIL (S2013/ICE-3019), Somedi (ITEA3 15011). Researcher Oscar Araque has been financed partially by the project Trivalent (H2020 Action Grant No. 740934, SEC-06-FCT-2016).

References

- [1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.
- [2] L. Zhuang, F. Jing, and X.-Y. Zhu, "Movie review mining and summarization," in *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006, pp. 43–50.
- [3] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from congressional floor-debate transcripts," in *Proceedings of the 2006 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2006, pp. 327–335.
- [4] N. Godbole, M. Srinivasiah, and S. Skiena, "Large-scale sentiment analysis for news and blogs," *ICWSM*, vol. 7, no. 21, pp. 219–222, 2007.
- [5] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," in *Mining text data*. Springer, 2012, pp. 415–463.
- [6] M. M. Bradley and P. J. Lang, "Affective norms for english words (anew): Instruction manual and affective ratings," Technical report C-1, the center for research in psychophysiology, University of Florida, Tech. Rep., 1999.
- [7] A. Esuli and F. Sebastiani, "Sentiwordnet: A high-coverage lexical resource for opinion mining," *Evaluation*, pp. 1–26, 2007.
- [8] L. Gatti, M. Guerini, and M. Turchi, "Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 7, no. 4, pp. 409–421, 2016.
- [9] P. D. Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 417–424.
- [10] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.
- [11] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [12] A. Esuli and F. Sebastiani, "Determining the semantic orientation of terms through gloss classification," in *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005, pp. 617–624.
- [13] A. Andreevskaia and S. Bergler, "Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses," in *EACL*, vol. 6, 2006, pp. 209–216.
- [14] W. Du, S. Tan, X. Cheng, and X. Yun, "Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon," in *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 2010, pp. 111–120.
- [15] Y. Choi and C. Cardie, "Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*. Association for Computational Linguistics, 2009, pp. 590–598.
- [16] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai, "Automatic construction of a context-aware sentiment lexicon: an optimization approach," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 347–356.
- [17] G. Qiu, B. Liu, J. Bu, and C. Chen, "Expanding domain sentiment lexicon through double propagation," in *IJCAI*, vol. 9, 2009, pp. 1199–1204.
- [18] S. Huang, Z. Niu, and C. Shi, "Automatic construction of domain-specific sentiment lexicon based on constrained label propagation," *Knowledge-Based Systems*, vol. 56, pp. 191–200, 2014.
- [19] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," *arXiv preprint arXiv:1606.02820*, 2016.
- [20] S. Almatarnah and P. Gamallo, "Automatic construction of domain-specific sentiment lexicons for polarity classification," in *15th International Conference on Practical Applications of Agents and Multi-Agent Systems*, Springer, Ed. Fernando De la Prieta, Zita Vale, Luis Antunes, Tiago Pinto, Andrew Campbell, Vicente Julián, Antonio J.R. Neves, María N. Moreno, 2017.
- [21] G. Demiroz, B. Yanikoglu, D. Tapucu, and Y. Saygin, "Learning domain-specific polarity lexicons," in *Data mining workshops (ICDMW), 2012 IEEE 12th international conference on*. IEEE, 2012, pp. 674–679.
- [22] J. Staiano and M. Guerini, "Depechemood: A lexicon for emotion analysis from crowd-annotated news," *arXiv preprint arXiv:1405.1605*, 2014.
- [23] K. Song, W. Gao, L. Chen, S. Feng, D. Wang, and C. Zhang, "Build emotion lexicon from the mood of crowd via topic-assisted joint non-negative matrix factorization," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 773–776.
- [24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [25] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [26] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Machine Learning*, vol. 85, no. 1, pp. 41–75, 2011.
- [27] J. McAuley, C. Targett, Q. Shi, and A. Van Den Hengel, "Image-based recommendations on styles and substitutes," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2015, pp. 43–52.
- [28] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2016, pp. 507–517.
- [29] J. Blitzer, M. Dredze, F. Pereira *et al.*, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *ACL*, vol. 7, 2007, pp. 440–447.
- [30] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

A.2.6 Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets

Title	Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets
Authors	Araque, Oscar and Barbado, Rodrigo and Sánchez-Rada, JF and Iglesias, CA
Proceedings	Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN
ISSN	1663-0073
Year	2017
Keywords	Deep Learning, Natural Language Processing, Sentiment Analysis, Recurrent Neural Network, TensorFlow
Pages	71-76
Online	http://ceur-ws.org/Vol-1896/p8_gsi_tass2017.pdf
Abstract	This article presents the participation of the Intelligent Systems Group (GSI) at Universidad Politécnica de Madrid (UPM) in the Sentiment Analysis work-shop focused in Spanish tweets, TASS2017. We have worked on Task 1, aiming to classify sentiment polarity of Spanish tweets. For this task we propose a Recurrent Neural Network (RNN) architecture composed of Long Short-Term Memory (LSTM) cells followed by a feedforward network. The architecture makes use of two different types of features: word embeddings and sentiment lexicon values. The recurrent architecture allows us to process text sequences of different lengths, while the lexicon inserts directly into the system sentiment information. The results indicate that this feature combination leads to enhanced Sentiment Analysis performances.
Volume	1896
Online	http://ceur-ws.org/Vol-1896/p8_gsi_tass2017.pdf

Applying Recurrent Neural Networks to Sentiment Analysis of Spanish Tweets

Aplicación de Redes Neuronales Recurrentes al Análisis de Sentimientos sobre Tweets en Español

Oscar Araque, Rodrigo Barbado, J. Fernando Sánchez-Rada y
Carlos A. Iglesias

Intelligent Systems Group, Universidad Politécnica de Madrid
Av. Complutense 30, 28040 Madrid
o.araque@upm.es, rodrigo.barbado.esteban@alumnos.upm.es
{jfernando, carlosangel.iglesias}@upm.es

Abstract: This article presents the participation of the Intelligent Systems Group (GSI) at Universidad Politécnica de Madrid (UPM) in the Sentiment Analysis workshop focused in Spanish tweets, TASS2017. We have worked on Task 1, aiming to classify sentiment polarity of Spanish tweets. For this task we propose a Recurrent Neural Network (RNN) architecture composed of Long Short-Term Memory (LSTM) cells followed by a feedforward network. The architecture makes use of two different types of features: word embeddings and sentiment lexicon values. The recurrent architecture allows us to process text sequences of different lengths, while the lexicon inserts directly into the system sentiment information. The results indicate that this feature combination leads to enhanced sentiment analysis performances.

Keywords: Deep Learning, Natural Language Processing, Sentiment Analysis, Recurrent Neural Network, TensorFlow

Resumen: En este artículo se presenta la participación del Grupo de Sistemas Inteligentes (GSI) de la Universidad Politécnica de Madrid (UPM) en el taller de Análisis de Sentimientos centrado en tweets en Español: el TASS2017. Hemos trabajado en la Tarea 1, tratando de predecir correctamente la polaridad del sentimiento de tweets en español. Para esta tarea hemos propuesto una arquitectura consistente en una Red Neuronal Recurrente (RNN) compuesta de celdas *Long Short-Term Memory* (LSTM) seguida por una red neuronal prealimentada. La arquitectura hace uso de dos tipos distintos de características: *word embeddings* y los valores de un diccionario de sentimientos. La recurrencia de la arquitectura permite procesar secuencias de texto de distintas longitudes, mientras que el diccionario inserta información de sentimiento directamente en el sistema. Los resultados obtenidos indican que esta combinación de características lleva a mejorar los resultados en análisis de sentimientos.

Palabras clave: Aprendizaje Profundo, Procesamiento de Lenguaje Natural, Análisis de Sentimientos, Red Neuronal Recurrente, TensorFlow

1 Introduction

Recent developments in the area of deep learning are strongly impacting sentiment analysis techniques. While traditional methods based on feature engineering are still prevalent, new deep learning approaches are succeeding and reduce the need of labeled corpus and feature definition. Moreover, traditional and deep learning approaches can be combined obtaining improved re-

sults (Araque et al., 2017).

This paper describes our participation in TASS 2017 (Martínez-Cámara et al., 2017). Taller de Análisis de Sentimientos en la SEPLN (TASS) is a workshop that fosters the research of sentiment analysis in Spanish for short text such as tweets. The first task of this challenge, Task 1, consists in determining the global polarity at a message level. The dataset for the evaluation of this task

considers annotated tweets with 4 polarity labels (P, N, NEU, NONE). P stands for positive, while N means negative and NEU is neutral. It is considered that NONE means absence of sentiment polarity. This task provides a corpus, which contains a total of 1514 tweets written in Spanish, describing a diversity of subjects.

We have faced this challenge as an opportunity to evaluate how these techniques could be applied in the TASS domain, and their results compared with the traditional techniques we used in a previous participation in this challenge (Araque et al., 2015).

The reminder of this paper is organized as follows. Sect. 2 introduces related work. Then Sect.3 describes the proposed polarity classification model and its implementation, which is evaluated in Sect. 4. Finally, conclusions are drawn in Sect. 6.

2 Related work

Many works in the last years involve the use of neural architectures to learn text classification problems and, more specifically, to perform Sentiment Analysis. A relevant example of this are Recursive Neural Tensor Networks (Socher et al., 2013). This architecture makes use of the structure of parse trees to effectively capture the negation phenomena and its scope. A similar work (Tai, Socher, and Manning, 2015) introduces the use of LSTM in tree structures, leveraging both the information contained in these trees and the representation capabilities of gated units. Although parse trees can result very useful in sentiment analysis, many works do not make use of them, as they introduce an additional computation overhead. In (Wang et al., 2015) a data-driven approach is described that learns from noisy annotated data also making use of LSTM units and a error signal processing to avoid the problem of vanishing gradient. Another useful technique is *attention* (Bahdanau, Cho, and Bengio, 2014), that enables weighting the importance of the different words in a given piece of text. Attention has been used in Sentiment Analysis successfully in a recurrent architecture, as presented in (Wang et al., 2016).

In the context of the TASS challenge, it has not been the first time that neural architectures have been proposed for solving the different tasks. In (Vilares et al., 2015), the authors propose a LSTM architecture that is

compared to linear classifiers. Also, word embeddings have been leveraged in previous versions, as shown in (Martinez-Cámara et al., 2015). Nevertheless, neural networks have not been thoroughly studied in TASS, and many potentially interesting techniques remain unused.

3 Sentiment analysis Task

3.1 Model architecture

The approach followed for the Sentiment Analysis at Tweet level Task consists in a RNN composed of LSTM cells that parse the input into a fixed-size vector representation. The representation of the text is used to perform the sentiment classification. Two variations of this architecture are used: (i) a LSTM that iterates over the input word vectors or (ii) over a combination of the input word vectors and polarity values from a sentiment lexicon.

The general architecture of the model takes as inputs the words vectors and the lexicon values for each word from an input tweet. Then, the inputs are passed through a one-layer LSTM with a tunnable number of hidden units. The generated representation is then used to determine the polarity of the input text using a feedforward layer with softmax activation as output function. The output of this last layer encodes the probability that the input text belongs to each class. Fig. 2 shows this architecture, which is further described as follows:

1. The input vector is the word embedding of each word in a given tweet. It contains word-level information or sentiment word-level information. Each specific case will be described in more detail afterwards.
2. The RNN number of units is chosen during training for optimization purposes. In this work we use a one-layer LSTM to avoid overfitting of the network to the training data.
3. The weight matrix has as input dimension the RNN size, and the number of classes as output dimension. This means that, taking as inputs the last LSTM output, we obtain a vector whose length is the number of classes. This matrix is also optimized during the training process.

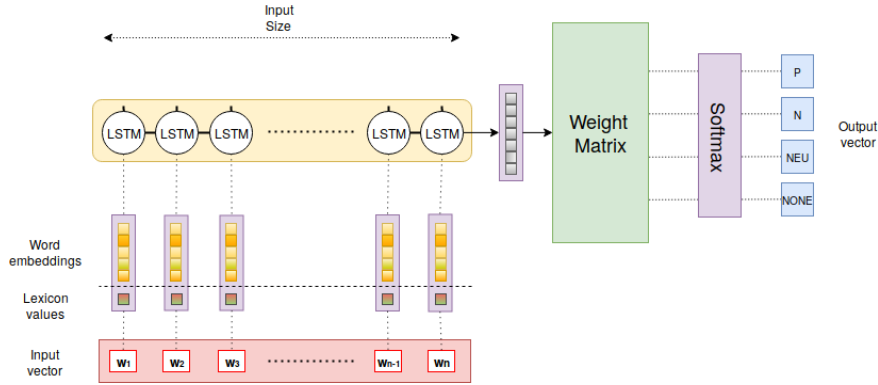


Figure 1: Recurrent Neural Network (RNN) architecture

4. The final probability vector is obtained by passing the result of the previous matrix multiplication through a softmax function, which converts the values of the components of this result vector into probabilities representation. Finally, the predicted label for the tweet is the component of the output vector with the highest probability.

Following, the two types of inputs used are described thoroughly.

3.2 Word-level RNN

For this input, the tweet text is tokenized into word tokens, which are then expressed in a one-hot representation. That is, each token is represented as a $\mathbb{R}^{|V| \times 1}$ vector with all 0s and one 1 at the index of that token in the sorted token vocabulary. For example, the representations for the tokens *a*, *antes* and *zebra* would appear as:

$$w_a = [100 \dots 0], w_{antes} = [010 \dots 0] \\ w_{zebra} = [000 \dots 1]$$

We limit the number of words to a certain vocabulary size in order to limit the computational cost of this preprocessing step. Before feeding this data to the network, each tweet is presented by the one-hot representation of all the tokens in the tweet.

3.3 Sentiment word-level RNN

Additionally, we include different sentiment information into the word representations by means of a sentiment lexicon. In this case, a similar approach as the word-level RNN

is followed, but instead of using information about the different words contained on each tweet, information about the sentiment of each word is used. In this case, the preprocessing process is modified:

1. First, each tweet is split into tokens.
2. Secondly, a sentiment dictionary is used to map words with sentiment polarity values. In this way, each word is mapped into a positive, neutral or negative value.
3. Finally, the representation of a word consists in its word vector concatenated with its sentiment polarity label.

3.4 Regularization

Given the reduced number of training examples that is available for this task (Sec. 4) a number of regularization techniques has been used in the experiments. Regularization is used in machine learning to control the complexity of a learning model so it does not overfit to the training data and generalized better to the test data.

It is known that Recurrent Neural Network tend to heavily overfit to the training set (Zaremba, Sutskever, and Vinyals, 2014). For this, we employ two regularization techniques to prevent this:

1. L2 regularization (Ng, 2004). This technique is applied on the weights of the feedforward layer of the network. Being W_{MLP} the weights of this layer, this regularization adds to the cost function the following value:

$$\lambda \|W_{MLP}^T W_{MLP}\|$$

where λ is a parameter that represents the importance that is assigned to this regularization in the overall cost function.

2. Dropout (Srivastava et al., 2014; Gal and Ghahramani, 2016b). This strategy consists in randomly setting a fraction of units to 0 at each step of the training process to prevent overfitting. During test time, the outputs are averaged by this fraction. Dropout has been recently found to be theoretically similar to applying a bayesian prior to the network weights (Gal and Ghahramani, 2016a).

3.5 TensorFlow implementation

TensorFlow is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms (Abadi, Agarwal, and et al., 2015). For implementing the model previously described, first we had to define a computation graph composed by the RNN architecture, matrices and operations needed. Once the graph is defined, the training process consists in iteratively adjusting numerical values in order to reach the best results. This task was done following those ideas:

- The values to optimize are the internal parameters and matrices that form the network. Those are: the word embedding representations, the LSTM internal weights and the feedforward weight matrix used as last layer. At the beginning of training, those values are initialized in a random way using a normal distribution $\sim \mathcal{N}(\mu, \sigma)$, with $\mu = 0$, and are considered variables to be optimized at each training step by TensorFlow.
- Having those variables defined, the training process iteratively modifies them in order to reach the better results. In order to obtain a error signal that can be used to modify the learning parameters we use a cost function which has to be minimized. That minimization problem is solved by applying the gradient descent method via backpropagation (LeCun et al., 2012). In this work we employ the Adam algorithm (Kingma and Ba, 2014).
- In each iteration of the training process, which are known as epochs, data from

the training set flows through all the computation graph yielding to a prediction result. The cost metric is computed by comparing the obtained result with the true training labels. When the backpropagation is finished, the variable values are updated and the following iteration proceeds.

- In order to enhance performance, we use early stopping on the accuracy on the development set. That is, for each epoch we monitor the performance of the network in the development set. If it has not improved for a number of epochs (in this work, 3 epochs) the training process is stopped and the model weights are freezed.
- The number of iterations can be chosen as well as other parameters such as the RNN size. For testing new examples, we use as input the test data, passing all the tweets through our model having as a result the vector of probabilities of the class each tweet belongs to, choosing the class with a higher probability value for each tweet.

4 Experimental setup

For the development of Task 1 a training and development dataset is made public, containing 1,514 labeled tweets which belong to the InterTASS corpus. Additionally, we use the TASS2015 edition training dataset that was extracted from the general corpus (García Cumberas et al., 2016). We train the system with the InterTASS and the TASS General Corpus training datasets, and adjust the hyper-parameters with the InterTASS development set. For the lexicon, we used ElhPolar dictionary (Urizar and Roncal, 2013), as it has been previously used in TASS competitions.

There are three test datasets, one belonging to the InterTASS corpus and two belonging to the General Corpus of TASS: the full version, with all the 60,798 tweets; and the 1k version, that contains a subset of 1,000 tweets.

In order to enhance the classification performance several hyper-parameters have been explored, and the values that yield better performance are selected to be used in the testing phase. The vocabulary size is set to 20,000, with a batch size of 256 and the num-

Model	Corpus	Accuracy	Macro-F1
LSTM + MLP	InterTASS	53.70	37.1
LSTM + MLP	TASS (1k)	60.1	45.6
LSTM + MLP	TASS (Full)	63.1	50.9
LSTM + MLP + Lexicon	InterTASS	56.2	38.7
LSTM + MLP + Lexicon	TASS (1k)	63.6	46.8
LSTM + MLP + Lexicon	TASS (Full)	63.1	49.7

Table 1: Results in TASS 2017

ber of epochs being 20. With this value, the early stopping mechanism stopped the training before its completion. Regarding the size of the layers, the number of dimensions of the word embeddings is set to 16, as well as it is done with the number of units in the LSTM layer. The dimensionality of the feedforward layer is given by the output of the LSTM, which is 16, and the number of classes of the classification task (in this case, 4). Note that these values are smaller than in the usual neural architectures in order to further prevent overfitting. Also, we select the λ parameter to 0.05, and the dropout rate to 0.7.

5 Experimental Results

Table 1 shows the results of the two variations of the proposed model: LSTM+MLP stack with or without lexicon values. In light of this results it is possible to affirm that the used architecture shows promising performances in the task of sentiment analysis of tweets. Although, the achieved performances are below the best in this year challenge. This indicates that further work should be done in order to improve the results.

The experimental results confirm the idea that the introduction of a sentiment lexicon into the word presentations results, in general, beneficial for the final performance. We see this improvement in the InterTASS and 1k corpora. Nevertheless, when attending to the Full corpus, a performance decrease in the Macro-F1 is observed.

6 Conclusions and Future Work

In this paper we have described the participation of the GSI in the TASS 2017 challenge. Our proposal relies on a Recurrent Neural Network architecture for Sentiment Analysis with Long Short-Term Memory cells. This network can be fed with both word vectors and sentiment lexicon values. This approach is able to represent a arbitrarily long se-

quence of text due to the dynamic recurrent structure of the architecture. Also, several techniques have been used for avoiding overfitting. From the experiments, it is seen that adding a sentiment lexicon can enhance the classification performance.

However, the proposed model does not compare with the best results in the TASS competition. This can be due to a number of reasons, but the training process suggests that overfitting is a relevant issue. Although benefit comes from the use of regularization techniques, the network is not able to largely generalize. To address this, we think that future work in this direction should include the expansion of the training set.

Other possible improvement for future work is doing a better preprocessing of input texts at word level. In addition, Convolutional Neural Networks could be used for feature extraction in combination with the Recurrent Neural Network architecture. This could lead to the computation of most complex features, which could also yield better results.

Acknowledgement

The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used in this research. This research work is partially supported through the projects Semola (TEC2015-68284-R), EmoSpaces (RTC-2016-5053-7), MOSI-AGIL (S2013/ICE-3019), Somedi (ITEA3 15011) and Trivalent (H2020 Action Grant No. 740934, SEC-06-FCT-2016).

Bibliografía

- Abadi, M., A. Agarwal, and P. B. et al. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Araque, O., I. Corcuera, C. Román, C. A.

- Iglesias, and J. F. Sánchez-Rada. 2015. Aspect based sentiment analysis of spanish tweets. In *TASS@ SEPLN*, pages 29–34.
- Araque, O., I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias. 2017. Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications. *Expert Systems with Applications*, June.
- Bahdanau, D., K. Cho, and Y. Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Gal, Y. and Z. Ghahramani. 2016a. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.
- Gal, Y. and Z. Ghahramani. 2016b. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.
- García Cumberras, M. Á., E. Martínez Cámara, J. Villena Román, and J. García Morera. 2016. Tass 2015—the evolution of the spanish opinion mining systems. *Procesamiento del Lenguaje Natural*, 56:33–40.
- Kingma, D. and J. Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- LeCun, Y. A., L. Bottou, G. B. Orr, and K.-R. Müller. 2012. Efficient backprop. In *Neural networks: Tricks of the trade*. Springer, pages 9–48.
- Martínez-Cámara, E., M. C. Díaz-Galiano, M. A. García-Cumberras, M. García-Vega, and J. Villena-Román. 2017. Overview of tass 2017. In J. Villena Román, M. A. García Cumberras, D. G. M. C. Martínez-Cámara, Eugenio, and M. García Vega, editors, *Proceedings of TASS 2017: Workshop on Semantic Analysis at SEPLN (TASS 2017)*, volume 1896 of *CEUR Workshop Proceedings*, Murcia, Spain, September. CEUR-WS.
- Martínez-Cámara, E., Y. Gutiérrez-Vázquez, J. Fernández, A. Montejo-Ráez, and R. Muñoz-Guillena. 2015. Ensemble classifier for twitter sentiment analysis. In R. Izquierdo, editor, *Proceedings of the Workshop on NLP Applications: completing the puzzle*, number 1386 in *CEUR Workshop Proceedings*, Aachen.
- Ng, A. Y. 2004. Feature selection, l_1 vs. l_2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78. ACM.
- Socher, R., A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, C. Potts, et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642.
- Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.
- Tai, K. S., R. Socher, and C. D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.
- Urizar, X. S. and I. S. V. Roncal. 2013. Elhuyar at tass 2013. In *Proceedings of the Workshop on Sentiment Analysis at SEPLN (TASS 2013)*, pages 143–150.
- Vilares, D., Y. Doval, M. A. Alonso, and C. Gómez-Rodríguez. 2015. Lys at tass 2015: Deep learning experiments for sentiment analysis on spanish tweets. In *TASS@ SEPLN*, pages 47–52.
- Wang, X., Y. Liu, C. Sun, B. Wang, and X. Wang. 2015. Predicting polarities of tweets by composing word embeddings with long short-term memory. In *ACL (1)*, pages 1343–1353.
- Wang, Y., M. Huang, X. Zhu, and L. Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, pages 606–615.
- Zaremba, W., I. Sutskever, and O. Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

A.2.7 Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis

Title	Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis
Authors	Araque, Oscar and Zhu, Ganggao and García-Amado, Manuel and Iglesias, Carlos A.
Proceedings	2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)
ISBN	978-1-5090-5910-2
Year	2016
Keywords	Sentiment analysis, ABSA, aspect based sentiment analysis, aspect classifier module, aspect context classification, aspect context detection, Context, data mining, Data mining, deep learning, Estimation, Feature extraction, Internet, natural language processing, NLP, opinion mining, opinion target extraction, opinionated Web, OTE, pattern classification, pipeline processing, Semantics, SemEval2016 datasets, sentiment analysis, sentiment classification, social media, social networking (online)
Pages	900–907
DOI	https://doi.org/10.1109/ICDMW.2016.0132
Online	https://ieeexplore.ieee.org/document/7836763
Abstract	Aspect Based Sentiment Analysis (ABSA) provides further insight into the analysis of social media. Understanding user opinion about different aspects of products, services or policies can be used for improving and innovating in an effective way. Thus, it is becoming an increasingly important task in the Natural Language Processing (NLP) realm. The standard pipeline of aspect-based Sentiment Analysis consists of three phases: aspect category detection, Opinion Target Extraction (OTE) and sentiment polarity classification. In this article, we propose an alternative pipeline: OTE, aspect classification, aspect context detection and sentiment classification. As it can be observed, the opinionated words are first detected and then are classified into aspects. In addition, the opinionated fragment of every aspect is delimited before performing the Sentiment Analysis. This paper is focused on the aspect classification and aspect context detection phases and proposes a twofold contribution. First, we propose a hybrid model consisting of a word embeddings model used in conjunction with semantic similarity measures in order to develop an aspect classifier module. Second, we extend the context detection algorithm by Mukherjee et al. to improve its performance. The system has been evaluated using the SemEval2016 datasets. The evaluation shows through several experiments that the use of hybrid techniques that aggregate different sources of information improves the classification performance.

Mining the Opinionated Web: Classification and Detection of Aspect Contexts for Aspect Based Sentiment Analysis

Oscar Araque, Ganggao Zhu, Manuel García-Amado and Carlos A. Iglesias
Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación
Departamento de Ingeniería Telemática
Spain, Madrid
Email: oscar.aiborra@alumnos.upm.es, gzhu@dit.upm.es,
manuel.garcia-amado.sancho@alumnos.upm.es, cif@dit.upm.es

Abstract—Aspect Based Sentiment Analysis (ABSA) provides further insight into the analysis of social media. Understanding user opinion about different aspects of products, services or policies can be used for improving and innovating in an effective way. Thus, it is becoming an increasingly important task in the Natural Language Processing (NLP) realm. The standard pipeline of aspect-based sentiment analysis consists of three phases: aspect category detection, Opinion Target Extraction (OTE) and sentiment polarity classification. In this article, we propose an alternative pipeline: OTE, aspect classification, aspect context detection and sentiment classification. As it can be observed, the opinionated words are first detected and then are classified into aspects. In addition, the opinionated fragment of every aspect is delimited before performing the sentiment analysis. This paper is focused on the aspect classification and aspect context detection phases and proposes a twofold contribution. First, we propose a hybrid model consisting of a word embeddings model used in conjunction with semantic similarity measures in order to develop an aspect classifier module. Second, we extend the context detection algorithm by Mukherjee et al. to improve its performance. The system has been evaluated using the SemEval2016 datasets. The evaluation shows through several experiments that the use of hybrid techniques that aggregate different sources of information improves the classification performance.

I. INTRODUCTION

An increasing number of users utilizes web sites and social media to share their experiences and degree of satisfaction with products, services or places, among others. On-line opinionated reviews are an important source of customer feedback that companies can use in order to measure satisfaction and even improve their products and services. Also, user-generated content in web sites and social networks has experimented an important growth [1]. This has contributed largely to the development of the Sentiment Analysis (SA) field. More concretely, Aspect Based Sentiment Analysis (ABSA) is the problem of mining opinions from text about specific entities and their associated aspects [1]. ABSA techniques require a more granular vision of the opinion mining problem as not only sentiment polarity is estimated, but also requires the aspects are identified and analysed. For example, an ABSA system that is presented with the text “*The food was lousy - too*

sweet or too salty and the portions tiny.” should express that both aspects, *food* and *portion*, are associated with a negative polarity.

In this article we present a complete ABSA system that addresses the different parts of the problem through a modular architecture, where each piece tackles a single task. The system consists of four phases. First of all, the *aspect detection* module is in charge of detecting the words that are referring to an opinion. That is, that words (or word) that form an aspect. Secondly, the *aspect classification* module classifies the detected aspect into one of several possible topics in the domain. Next, the *context detection* module determines the aspect context boundary. Finally, the *sentiment estimation* module realizes a sentiment analysis of the opinion and its aspect, yielding the estimated polarity.

The rest of this paper is organized as follows. In Sect. II the related work to our proposal is summarized. Sect. III presents the architecture of the system as a whole, briefly describing the aim of each module. Sect. IV describes our proposal for combining knowledge and corpus sources for aspect classification. Following, our proposal for context detection is presented in Sect. V. Sect. VI presents the sentiment estimation module. In order to evaluate the proposed system, Sect. VII depicts the experimental results obtained. Finally, conclusions and future work are presented in Sect. VIII.

II. RELATED WORK

Many approaches aim to detect the global sentiment polarity of a document or a sentence, but efforts have been made to analyze the sentiment at the aspect level [2], [3]. In this context, our work presents a hybrid system that classifies both the aspect and its opinion. As indicated by [4], aspect based sentiment analysis has typically two steps: (i) identification and extraction of the aspects included in an opinion sentence, and (ii) estimating the sentiment polarity of said aspects. This work deals with these two problems. In this section, the related work of both problems is summarized.

A. Aspect Classification

In the context of topic classification, Latent Dirichlet Allocation (LDA) driven models can be used [4]. LDA is a generative probabilistic model that considers each document as a mixture of topics. Added to this, many variations of this topic modeling method have been presented, such as pLDA [5]. In similar lines, [6] proposes a multi-grain approach to extract opinion aspects (MG-LDA), extending the LDA approaches. The MG-LDA method extracts the opinion aspects and it also clusters them into topics.

Some additional methods for aspect classification are the semantic-based approaches. These techniques use the concept of *semantic relatedness* to aid in many Natural Language Processing tasks. Basically, the problem is to determine the relation between concepts or words. That is, it is aimed to create a way of measuring the distance between the aspect words and certain topics. In this context, some curate knowledge sources (lexical databases) can be useful, such as WordNet [7]. Also, more simple resources have been used, as the network-based dictionary approach proposed in [8].

B. Sentiment estimation

The dominant approaches on sentiment analysis are driven by *machine learning* methods [3], [9]. The most common approach consists on the Bag of Word (BOW) model, where each document is transformed into a feature vector that is then fed to a classification algorithm. Other types of features are usually used, such as Part of Speech (POS) tagging, which is a elemental model of syntactic analysis [10]. An statistical approach for representing documents is known as TF-IDF, where words are weighted depending on their frequency on the corpus [11]. Moreover, many sentiment analysis design involve the use of a *sentiment lexicon* as source of subjective information [12]. Nevertheless, lexicon-based approaches have many drawbacks: the necessity of labeled data that is reliable and consistent, the expression differences between domains and the fact that lexicons cannot be automatically translated for multilingual use [13]. Also, extracting non-simple features from text and figuring out which ones are relevant is a fundamental question in the machine learning driven techniques [14].

Alternatively, *deep learning* techniques have shown promising performance in many NLP tasks, including sentiment analysis [15]. One common use of deep learning is to learn complex features from the data with minimum external contribution through deep neural networks [16]. Continuous representations of words as vectors, also known as word embeddings have been used for sentiment analysis [17]. Besides, one interesting approach is to augment the knowledge contained in these word embeddings with other sources of information. This added information can be sentiment specific word vectors [17], or a concatenation of manually crafted features with word vectors [18]. Another approach that incorporates new information to the embeddings consists on extract sentiment features in conjunction with semantic features [19].

In addition to these approaches, ensemble methods can be used for improving sentiment classifications performance. Ensemble methods combine the predictions of various classifiers (base classifiers) and apply some function on them in order to yield a final prediction. Rule based ensemble, such as majority voting, can be quite effective in the task of sentiment classification [20]. Besides, extra subjective knowledge can be added with ensemble techniques, such as POS using a rule based ensemble model [21]. Also, a meta classifier ensemble model can be used, as in [20]. Meta learning models are based on the use of base classifiers predictions as features fed to an additional classifier that predicts the polarity.

III. SYSTEM OVERVIEW

The proposed system is divided into four modules, each one addressing a different dimension of the ABSA problem: (i) the *aspect detection module* that detects OTE in an opinionated text; (ii) the *aspect classification module* that classifies each detected target into one of several possible classes in the studied domain; (iii) the *context detection module* that determines the limits of the opinion text for the detected aspect; and (iv) the *sentiment analysis module* is in charge of computing the sentiment polarity of the target based on the detected context.

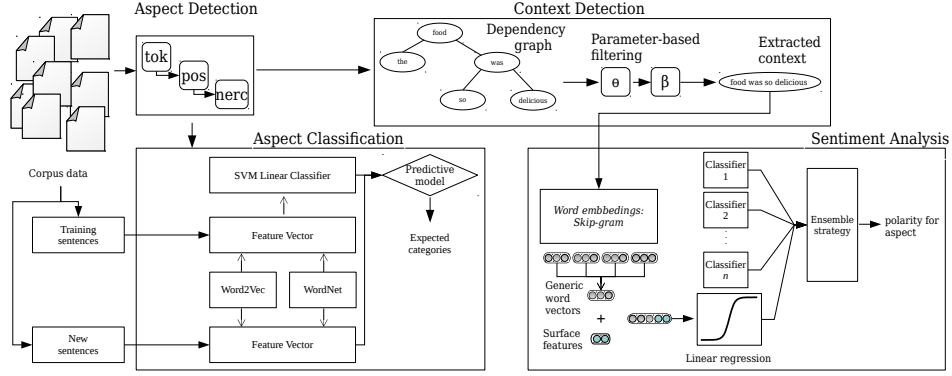
Figure 1 illustrates the data flow of the system and its main components. The process of Aspect Based Sentiment Analysis is as follows. Firstly, the text data is analyzed by the aspect detection module. This module is the one proposed in [22], composed of several sub-modules called *pipes*. Each pipe realizes a specific function and, when combined, they yield the set of words that represent an aspect in the opinion text. That is, the combination of these pipes allows us to detect the aspects. The *ixa-pipe-tok* tokenizes and segments the text, *ixa-pipe-pos* performs POS tagging and lemmatizes, and *ixa-pipe-nerc* performs the Opinion Target Extraction (OTE). We have used the already trained models offered by the tools, as they are prepared for the restaurant-reviews domain.

The, the detected aspects are passed to the aspect classification (Sect. IV) and context detection (Sect. V) modules. Finally, the context detection module requests analyses to the sentiment estimation (Sect. VI) module.

IV. ASPECT CLASSIFICATION

Aspect Category Detection is a sub-task of ABSA, aiming to identify every entity *E* and attribute *A* pair, towards which an opinion is expressed in the given text [23]. Specifically, given an input sentence such as “The food was delicious”, the aspect category detection extracts the E and A pair (e.g., category=FOOD#QUALITY) for the target word “food”. We have chosen the English restaurants domain of the ABSA of SemEval2016 [23]. In the restaurant domain, SemEval pre-defines a set of entity labels (SERVICE, RESTAURANT, FOOD, DRINKS, AMBIANCE, LOCATION) and a set of attribute labels (GENERAL, PRICE, QUALITY, STYLE_OPTION, MISCELLANEOUS). The entities and labels compose 12 categories. Our task of aspect category classification consists in assigning a aspect category to the opinion target words.

Fig. 1. Pipeline of the proposed Aspect-based Sentiment Analysis System



The baseline of this aspect category classification provided by SemEval employs a Support Vector Machine (SVM) with a linear kernel. Specifically, n unigram features are extracted from the training data, where the category value (e.g., FOOD#QUALITY) of the tuple is used as the correct label of the feature vector [23]. For each test sentence s , a feature vector is built and the trained SVM is used to predict the correct category. This unigram feature representation lacks of the ability to address those feature words that are not encountered in the training process. As reported in SemEval [23], word clusters learned from Yelp data are used to expand the features. However, those similar words of word clusters are added to feature vectors considering the same weight as the unigram features appearing in the training data, without concerning the different semantic distance between words.

With such concerns, we aim at combining knowledge (e.g. WordNet) and corpus (e.g. Yelp) sources in order to improve aspect classification. Our main contribution is the hybrid model that consists of a word embeddings model [24] and semantic similarity model using WordNet [25]. We propose to use similarity score as the weight of each vector dimension so that the semantic similarity between words computed by word2vec and semantic similarity measures are included for training. Specifically, we explicitly use the n unigrams as feature vector, in which the word similarity between target words and feature words are used to represent each dimension of feature vector. The idea is to train a semantic predictive model for each category based on the feature words and similarity models using SVM. Formally, let $F = \{f_1, f_2, \dots, f_n\}$ be the set of feature words, a feature vector is represented as $V \in [0, 1]^N$. For a set of target words $T = \{w_1, \dots, w_m\}$, the value of a dimension f_i is computed from $\max_{w_j \in T} \text{sim}(w_j, f_i)$, where the sim function denotes the word similarity between two words. The calculation of similarity scores is more computational intensive than counting the occurrence of words. Since

the target words are in the form of short text (several words), and the feature vector can be composed by most representative words (small vector dimensions), the intensive computation problem can be alleviated using word similarity matrix.

The sim function is implemented by word2vec [24] for training Yelp data and the semantic similarity measures based on WordNet [25]. For word2vec, we have obtained a continuous representation of words, where words that co-occur frequently are mapped to vectors close in vector space. Based on the distributional semantics hypothesis, the words co-occur in a same surrounding context are treated as relevant so that they have high similarity. Consequently, the $\text{sim}(w_j, f_i)$ function is implemented as cosine similarity between two word vectors. Using this word2vec similarity model, a first feature vector $V_{\text{word2vec}} \in [0, 1]^N$ is obtained.

The word2vec model considers the co-occurrence information of the same surrounding context, which would make a wide variety of words to be considered as related. This would challenge the word2vec model when discriminating words from different categories that are frequently collocated (e.g. food and drink). For instance, in restaurant domain, those target words such as fish and wine would appear in same surrounding contexts (e.g. “the fish is delicious and the wine is great”). If a word2vec model is trained from such corpus simply based on calculating co-occurrences of words, many words belonging to different categories would have similar similarity. In order to solve this problem, semantic similarity methods using WordNet [25] are useful to complement the word2vec model by including the structural knowledge from taxonomy. As illustrated in a fragment of WordNet in Fig. 2, lamb, beef, and seafood are sub-concepts of FOOD category, while coffee, tea and milk are sub-concepts of DRINKS category. Although WordNet based similarity model can retain taxonomical information from WordNet, it can only address limited words that are contained in WordNet. Combining

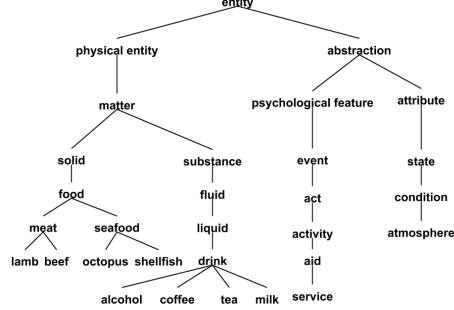


Fig. 2. A Fragment of WordNet Concept Taxonomy

word2vec and WordNet similarity models can enable the aspect classification model to have good ability in addressing large vocabularies and encoding hierarchical knowledge of common words from WordNet. In consequence, apart from Word2Vec, we also consider the semantic similarity methods using WordNet.

The semantic similarity methods exploit the hierarchical classification of all words via a relation, whose intuition is that two words are more similar if they are closer to each other in WordNet taxonomy. There have been many semantic similarity measures proposed in the literature [7]. To implement the WordNet based sim function, we study some of the most common ones. The simplest semantic similarity measure is counting the number of nodes or edges (shortest path) connecting two words in WordNet taxonomy. Let $path(w_i, w_j)$ be the shortest path between w_i and w_j , the Path [26] method defines semantic similarity as:

$$sim_{Path}(w_i, w_j) = \frac{1}{1 + path(w_i, w_j)} \quad (1)$$

The Leacock-Chod [27] method measures the semantic similarity between words based on their shortest path length using a non-linear function illustrated in Eq.(2):

$$sim_{Leacock-Chod}(w_i, w_j) = -\log\left(\frac{length(w_i, w_j)}{2D}\right) \quad (2)$$

where D is the maximum depth of the taxonomy. The idea of using depth information lies in the property of taxonomies that the upper-level words in a taxonomy are supposed to be more general. For example in Fig. 2, the word pair *lamb* and *beef* are more similar than the word pair *meat* and *seafood*. The Wu & Palmer method measures the depth of two words in a taxonomy with the Least Common Subsumer (LCS), which is the most specific word that is a shared ancestor of the two words. For example, the LCS of word *beef* and word *octopus* is the word *food*. Let w_{lcs} be the LCS of words w_i and w_j , then

the Wu & Palmer [28] method measures semantic similarity of given words using the following formula:

$$sim_{Wu\&Palmer}(w_i, w_j) = \frac{2depth(w_{lcs})}{depth(w_i) + depth(w_j)} \quad (3)$$

The above knowledge-based semantic similarity methods consider the structure of a taxonomy which have a common drawback of uniform distance between words in the taxonomy. Some other approaches consider the Information Content (IC) to solve the uniform distance drawback. The IC of a word is given by the probability of encountering the word in a corpus. Note that we use Brown Corpus [29] for WordNet to compute IC. The Resnik [30] method relies on the IC of LCS node of two words.

$$sim_{Resnik}(c_i, c_j) = IC_{corpus}(c_{lcs}) \quad (4)$$

The consequent works by Lin [31] and Jiang & Conrad [32] extend the IC-based method by including the IC of words.

$$sim_{Lin}(w_i, w_j) = \frac{2IC(w_{lcs})}{IC(w_i) + IC(w_j)} \quad (5)$$

$$sim_{Jiang\&Conrad}(w_i, w_j) = \frac{1}{1 + IC(w_i) + IC(w_j) - 2IC(w_{lcs})} \quad (6)$$

The IC-based methods lack of important information of path and depth. In order to choose the best WordNet based semantic similarity method for the aspect category classification, we will experiment with all the semantic similarity described above in evaluation.

As illustrated in Fig. 1, a list of feature words are extracted from training data. Apart from the word2vec based feature vector $V_{word2vec}$ mentioned previously, another feature vector $V_{wordnet} \in [0, 1]^N$ is composed by computing the semantic similarity between target words and feature words using the WordNet based semantic similarity methods. Consequently, a $2N$ dimension vector is composed for training and classifying new sentences by considering both word2vec similarity model and WordNet similarity model. The evaluation of this module is presented in Sect. VII-B. The main results show that combining word embedding and semantic similarity measures can improve the performance of aspect category classification.

V. ASPECT CONTEXT DETECTION

Aspect Context Detection is the task of detecting the text fragment in the original text that corresponds to the opinion about an attribute A of an entity E . For the aspect context detection, we have modified the algorithm proposed by Mukerjee et al. [33] in order to improve its performance. The original algorithm is based on computing the distance between words through dependency parsing. In this way, these distances can be represented in a graph, allowing the computation of the aspect context. This context detection technique is based in the assumption that the more closely associated words come together to express an opinion regarding a certain aspect. If n aspects $(a_1, a_2, \dots, a_n)$ have been detected in an opinion, the

algorithm for extracting the set of words w_i that express any opinion about the target aspect a_t proceeds as described in Algorithm 1.

Algorithm 1 Dependency extraction algorithm

- 1) Initialize n clusters $C_i \forall i = 1..n$
 - 2) Make each $a_i \in A$ the clusterhead of C_i . The target aspect a_t is the clusterhead of C_t . Initially, each cluster only consists of the clusterhead.
 - 3) Assign each word w_j to cluster C_k s.t.
 $k = \arg \min_{i \in n} d(w_j, a_i)$
 - 4) Merge any cluster C_i with C_t if $d(a_i, a_t) < \theta$, where θ is some threshold distance.
 - 5) The set of words $w_i \in C_t$ expresses the opinion regarding the target aspect a_t .
 - 6) If $\beta \neq 0$, add to (or remove from) C_t the words w_p so that

if $\beta > 0$:

$$\max d(w_i, a_t) < d(w_p, a_t)$$

$$d(w_p, a_t) \leq \max d(w_i, a_t) + \beta$$

if $\beta < 0$:

$$\max d(w_i, a_t) + \beta \leq d(w_p, a_t)$$

$$d(w_p, a_t) < \max d(w_i, a_t)$$
-

The original formulation of this algorithm includes a threshold parameter (θ) that controls the association of similar opinion contexts. Added to this, we have generalized this algorithm by adding one more parameter (β) that modifies the behaviour of the algorithm. This generalization intends to improve the sentiment analysis performance of the system by expanding or reducing the number of words that are included in aspect contexts. After the context is detected, this additional parameter controls the obtained context, adding or removing context words. Regarding this, we do such attending to the distances in the computed dependency graph. It consists on the value obtained after the generation of the dependency graph, the measure of the distance in it.

This β parameter controls the number of words that are either added ($\beta > 0$) or removed ($\beta < 0$) from the aspect context considering the distances on the dependency graph. Algorithm 1 shows the modified method with this type of distance. The associated parameter differs from the original formulation of the algorithm [33] when its value is not zero. That is, when $\beta = 0$, our proposal is identical to that of the original.

VI. ASPECT BASED SENTIMENT ANALYSIS

In this module, the detected context (Sec. V) is used for the sentiment estimation of the aspect. That is, we consider that the text that is contained in the detected context refers to the analyzed aspect, and thus that is what the sentiment estimation modules uses.

The sentiment classification has been addressed with the use of previously proposed models. These sentiment analysis

models aim to take advantage of different types of features, assuming that a sentiment classifier can yield a better performance in the sentiment analysis task when it is provided with a higher quantity and variety of information. For this end, two different combination techniques are used: ensemble of classifiers and ensemble of features.

On one hand, the ensemble of classifiers combines the predictions of the classifiers that form the ensemble (base classifiers). In this way, the augmented information is given through each classifier's sentiment prediction. On the other hand, the ensemble of features combines the word vectors or features that have been extracted differently. With this, different sources of information are inserted into a single classifier.

The features used in this work are: generic word vectors, representations obtained through a word embeddings algorithm; and surface features, such as sentiment lexicons and Part-of-Speech tagging. The models we use in this work are described next.

Generic word vectors model (M_G). This model combines the vectors from each word of the analyzed document, and aggregates them into a single vector. The aggregation functions that are used are the *average*, *max* and *min*. The generic word vectors are obtained using the *skip-gram* model [24]. Once the aggregated vectors have been composed, they are fed to a linear regression algorithm, that yields the sentiment polarity. Unlike the following models, M_G does not combine different sources of information.

Ensemble of classifiers (CEM_{SG}) model. This model ensembles the predictions of a number of classifiers that have been trained with both surface features and generic word vectors, as in the M_G model. The ensemble techniques used are two. First, a fixed rule strategy known as majority voting, where the polarity class is decided by the voting results of the base classifier. In case of tie, the positive polarity is selected. Second, a meta-learning technique, where the predictions of the base classifiers are used as features for a meta learner that yields the final sentiment prediction. In this work, the meta classifier is implemented using the Random Forest algorithm. The base classifiers of the ensemble are the same as in [34].

Ensemble of features (MSG), an ensemble of features model. In this model, the attempt to improve the M_G proposal is tackled by combining the previously used generic word vectors and a set of surface features. Both types of features are combined by concatenation, obtaining an enlarged vector. This vector, as in M_G model, is then fed to a linear regression classifier, that predicts the sentiment polarity. The surface features used in this work are: Wordnet-Affect lexicon values [35], number of exclamation and interrogation marks, number of positive, neutral and negative words, number of words that are in caps and number of elongated words.

VII. EVALUATION

In order to evaluate the aspect classification, context detection and sentiment analysis sub-modules, we have performed several experiments. In these experiments, we aim to evaluate

TABLE I
STATISTICS OF THE USED DATASETS.

Dataset	#Positive	#Negative
Yelp-extracted	1,492,558	450,540
SemEval16 train	1,696	773
SemEval16 test	609	204

the effectiveness of the proposed system and, also, optimize some of the described parameters. The metric used is the F-score.

A. Datasets

For this evaluation, we have extracted a dataset that is aligned with the restaurant reviews domain from the Yelp Challenge dataset¹. This dataset provides with a high quantity of data that can be used for the word embeddings training. Also, we labeled this dataset using a distant supervision strategy for the sentiment polarity. That is, we have taken advantage of the Yelp start-based rating, considering 1 or 2 stars as negative sentiment, and 4 or 5 start as positive polarity. In this work, we do not consider the role of the neutral polarity.

Also, we have used as development dataset (learning of hyper-parameters) the SemEval16 training data, and as test dataset (final validation of the sentiment performance) the SemEval16 test set. These two datasets, as well as the one extracted from Yelp, are summarized in Table I.

B. Aspect Category Classification Evaluation

We use the SemEval16 dataset of English Restaurant domain dataset. The training dataset consists of 1880 tuples and the test dataset consists of 650 tuples. We extracted most common 10 words of each category and composed into 76 feature words by removing duplicates. The small feature number is not a problem since the vocabularies are contained in word2vec and WordNet. Nevertheless, the quality of feature words should be considered because we use the word similarity scores as the value of feature vectors. We use the most frequent words for simplicity in this article. The word2vec similarity model and WordNet similarity model are used to compute word similarity between target words and feature words. We trained the aspect classification model using the linear kernel of SVM using the sklearn² package. The classification metrics accuracy, precision, recall and F-score are used as the performance metrics to evaluate the different models.

We have experimented with the classification model in different settings: simple feature, knowledge-based feature, dense vector feature and combined features. The experimental results are shown in Table II. In the simple feature, we use the simple word list features $V_{wordlist} \in \{0,1\}^N$, where the word list is the 76 feature words. In this setting, we use the unigram occurrence feature to train a classification module using SVM, and use this model as baseline. Note that the

¹https://www.yelp.com/dataset_challenge

²<http://scikit-learn.org>

TABLE II
ACCURACY, PRECISION, RECALL AND F-MEASURE OF ASPECT CATEGORY CLASSIFICATION USING DIFFERENT METHODS.

Method	Corpus & KB	Accuracy	Precision	Recall	F-measure
Simple Feature	Word List	.745	.72	.74	.71
Knowledge-based					
Path. [26]	WordNet	.78	.77	.78	.75
Leacock-Chod. [27]	WordNet	.757	.73	.76	.73
Wu & Palmer. [28]	WordNet	.751	.70	.75	.72
Resnik. [30]	WordNet	.646	.65	.65	.63
Lin. [31]	WordNet	.774	.73	.77	.74
Jiang & Conrad. [32]	WordNet	.768	.77	.77	.74
Dense Vectors					
Word2Vec. [24]	Yelp	.818	.79	.82	.78
Combination					
Word2Vec + Path	WordNet + Yelp	.82	.80	.82	.79
Word2Vec + Leacock-Chod	WordNet + Yelp	.81	.80	.81	.78
Word2Vec + Wu & Palmer	WordNet + Yelp	.813	.80	.81	.78
Word2Vec + Resnik	WordNet + Yelp	.814	.80	.81	.78
Word2Vec + Lin	WordNet + Yelp	.813	.80	.81	.78
Word2Vec + Jiang & Conrad.	WordNet + Yelp	.82	.80	.82	.79

different learning softwares and settings would influence the experimental results so that we implemented a simple baseline following the description of SemEval. In order to show that the similarity based feature is more effective than the simple word occurrence feature, we extended the simple feature model to the knowledge-based model and dense vector model. In the knowledge-based setting, we have trained and evaluated the classification model using the WordNet based similarity measures respectively. Table II shows that the Path [26] similarity measure is the best metric for aspect classification, and the most of similarity measures are more effective than the baseline except for the Resnik [30] method. In the dense vector setting, we have used word2vec embedding to learn the word vectors from Yelp comments data, and trained the aspect classification model only with the word2vec similarity model. The experimental result shows that the word2vec similarity model is more effective than knowledge-based methods and baseline. By looking at each category, we found that the knowledge-based features are more effective for food and drink categories while word2vec performs better in other categories. Since word2vec feature is trained from a domain corpus (Yelp comments), it has better coverage in vocabularies and the categories such as AMBIENCE, LOCATION are more concerned with relevant features rather than hierarchical feature. In the combined setting, we use both word2vec similarity model and WordNet similarity model to train and evaluate in order to select the best combination between word embedding and semantic similarity methods. Table II shows that both Path [26] and Jiang & Conrad [32] are the best in combining with word2vec, in terms of F measure (.79).

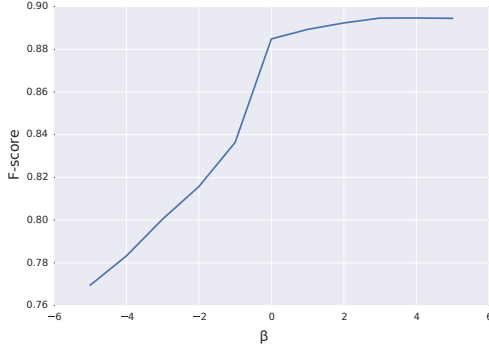
In summary, from the experimental results, we found that the similarity based features are effective in learning the aspect classification model. Furthermore, combining the word embedding model and semantic similarity measure is promising in training aspect classification model, since it has achieved best performance in our experiments, and it can combine the word cooccurrence information together with hierarchical knowledge from WordNet.

C. Context detection validation

In these context detection experiments, we have taken as performance metric the sentiment F-score on the development set with the M_G sentiment classification model. In this way, the different contexts that vary from the variation of several parameters are fed to this sentiment model. As for the training of the sentiment classifier, it is explained further in Sect. VII-D.

The parameter validated in the context detector module is the β value, that controls the words that added or removed to the context. As can be seen in Figure 3, the sentiment performance increases when $\beta > 0$. The improvement between $\beta = 0$ (original formulation) and $\beta = 6$ (maximum improvement of our proposal) is of 3.04% of the F-score.

Fig. 3. Variation of the F-score with the β parameter of the context detector.



D. Sentiment training and validation

Firstly, the *skip-gram* model has been trained with the Yelp-extracted dataset, setting the dimension of the resulting vectors to 400, and a minimum count of 5. As this is unsupervised training, polarity labels have not been used.

After the training of the word embeddings model, the M_G model requires that a linear regression algorithm is trained. For this, we used the aggregated vectors for each document of the Yelp-extracted dataset, and the corresponding distant sentiment labels. Also, the set of possible aggregation functions on the M_G model has been evaluated on the SemEval16 train data, obtaining the best performance with the *average* function a 91.79 % of F-score. The next high performance in this sense is the combination *max+avg*, with a 87.94 %.

The ensemble of classifiers model does not need a training process, as it is composed of already trained classifiers. Nevertheless, the meta-learning strategy does need training, as it learns from the predictions of its base classifiers. For this end, the meta classifier is trained with the development data.

Finally, Table III shows the F-scores for the different sentiment models in the SemEval16 test dataset. CEM_{SG}^{Vo} is the ensemble of classifier with the majority voting scheme, and CEM_{SG}^{MeL} with the meta learning strategy. The BOW baselines

TABLE III
FSCORES OBTAINED IN THE SEMEVAL16 TEST DATASET.

Model	F-Score (binary)
BOW + TF-IDF + SVM	81.29
BOW + SVM	81.78
M_G (<i>skipgram</i>)	90.76
M_G + TF-IDF	90.67
CEM_{SG}^{Vo} (<i>skipgram</i> + voting)	87.54
CEM_{SG}^{MeL} (<i>skipgram</i> + meta-learning)	81.75
M_{SG} (<i>skipgram</i> + surface features)	91.00

are tested and compared with the proposed models. It can be seen that the TF-IDF does not improve the sentiment performance in these experiments. Also, the best performing model is the M_{SG} model. The experiment result indicates that joining generic word vectors and surface features through a feature ensemble strategy improves the sentiment performance. Nevertheless, the two classifier ensemble methods do not result in a classification improvement, but a performance decrease.

To the extent of our knowledge, there is no public disaggregation of F-score for the predicted classes on the three best systems. Nevertheless, we can compare to the proposal described in [36], that claim their F-score for positive and negative classes are 88.26 and 76.21%, respectively. Our systems performs better in relation to the positive class, with a 91.00%, but it does not perform better on the negative class, yielding 73.91%. For both systems, the F-score metric for the neutral class is 0.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presents an Aspect Based Sentiment Analysis system divided into four modules, each one addressing a single step of the ABSA problem: aspect detection, aspect classification, aspect context detection and sentiment estimation.

For the aspect classification module we proposed a hybrid approach where both word embeddings and semantic similarity measures are used. The experiments show that the combination of these two types of features improves the classification when compared to these same techniques separately.

The aspect context detection module uses a modified dependency parsing tree algorithm whose assumption is that close words in the dependency tree express an opinion of the same aspect. A generalization parameter is introduced into the algorithm and evaluated on the data, finding that this addition on the detected context methods improves the sentiment classification performance.

The sentiment estimation module is composed by a hybrid system that uses a configurable combination of word embeddings, traditional sentiment features and an ensemble of classifiers. The combination of traditional sentiment features (e.g., sentiment lexicon values) and *skipgram* word embeddings is shown to improve the sentiment performance of the system.

All in all, we have addressed in this paper how the combination of knowledge and corpus sources can improve both aspect classification and polarity detection, being complemented by similarity metrics in aspect classification. Moreover, we have explored how modifying the scope of the aspect context

affects context detection, and have proposed a generalisation of Mukherjee et al. algorithm that can be used for its optimisation in other datasets.

As future work, we consider that the aspect classification module performance can be increased if using a domain adapted semantic knowledge base. In this way, the studied domain can be better characterized. Regarding the sentiment estimation, a neural approach can be taken to train specific word vectors adapted to the domain, instead of using generic word vectors. Also, we have intentions of expanding these systems to other domains, such as emotion analysis.

ACKNOWLEDGEMENT

This research work is supported by the EC through the H2020 project MixedEmotions (Grant Agreement no: 141111) and by the Spanish Ministry of Economy under the R&D project Semola (TEC2015-68284-R).

REFERENCES

- [1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.
- [2] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowledge-Based Systems*, vol. 108, pp. 42–49, 2016.
- [3] S. Poria, E. Cambria, L.-W. Ku, C. Gui, and A. Gelbukh, "A rule-based approach to aspect extraction from product reviews," in *Proceedings of the second workshop on natural language processing for social media (SocialNLP)*, 2014, pp. 28–37.
- [4] M. Cataldi, A. Ballatore, I. Tiddi, and M.-A. Aufaure, "Good location, terrible food: detecting feature sentiment in user-generated reviews," *Social Network Analysis and Mining*, vol. 3, no. 4, pp. 1149–1163, 2013.
- [5] Y. Wang, H. Bai, M. Stanton, W.-Y. Chen, and E. Y. Chang, "Plda: Parallel latent dirichlet allocation for large-scale applications," in *International Conference on Algorithmic Applications in Management*. Springer, 2009, pp. 301–314.
- [6] I. Titov and R. McDonald, "Modeling online reviews with multi-grain topic models," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 111–120.
- [7] A. Budanitsky and G. Hirst, "Evaluating wordnet-based measures of lexical semantic relatedness," *Computational Linguistics*, vol. 32, no. 1, pp. 13–47, 2006.
- [8] H. Kozima and T. Furugori, "Similarity between words computed by spreading activation on an english dictionary," in *Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, 1993, pp. 232–239.
- [9] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, 2012, pp. 90–94.
- [10] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, ser. HLT '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 42–47.
- [11] J. Martineau and T. Finin, "Delta tfidf: An improved feature space for sentiment analysis," *ICWSM*, vol. 9, p. 106, 2009.
- [12] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. New York, NY, USA: ACM, 2009, pp. 1275–1284.
- [13] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
- [14] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, "Sentiment analysis of twitter data," in *Proceedings of the workshop on languages in social media*. Association for Computational Linguistics, 2011, pp. 30–38.
- [15] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuska, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.
- [16] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [17] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification," in *ACL (1)*, 2014, pp. 1555–1565.
- [18] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Cocoonll: A deep learning system for twitter sentiment classification," in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 2014, pp. 208–212.
- [19] Z. Su, H. Xu, D. Zhang, and Y. Xu, "Chinese sentiment classification using a neural network tool word2vec," in *Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014 International Conference on, Sept 2014, pp. 1–6.
- [20] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," *Information Sciences*, vol. 181, no. 6, pp. 1138 – 1152, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025510005682>
- [21] R. Xia and C. Zong, "A pos-based ensemble model for cross-domain sentiment classification," in *IJCNLP*. Citeseer, 2011, pp. 614–622.
- [22] R. Agerri, J. Bermudez, and G. Rigau, "Ixa pipeline: Efficient and ready to use multilingual nlp tools," in *LREC*, vol. 2014, 2014, pp. 3823–3828.
- [23] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2016 task 5: Aspect based sentiment analysis," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*. Citeseer, 2016, pp. 27–35.
- [24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [25] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-based and knowledge-based measures of text semantic similarity," in *AAAI*, vol. 6, 2006, pp. 775–780.
- [26] R. Rada, H. Mili, E. Bicknell, and M. Blettner, "Development and application of a metric on semantic nets," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17–30, 1989.
- [27] C. Leacock and M. Chodorow, "Combining local context and wordnet similarity for word sense identification," *WordNet: An electronic lexical database*, vol. 49, no. 2, pp. 265–283, 1998.
- [28] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ser. ACL '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 133–138.
- [29] W. N. Francis and H. Kucera, "Brown corpus manual," *Brown University*, 1979.
- [30] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 448–453.
- [31] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 296–304.
- [32] J. J. Jiang and D. W. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," *Computational Linguistics*, vol. cmp-1g/970, no. Rocling X, p. 15, 1997.
- [33] S. Mukherjee and P. Bhattacharyya, "Feature specific sentiment analysis for product reviews," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2012, pp. 475–487.
- [34] O. Araque, "Prototype of a Sentiment Analysis System Based on Ensemble Algorithms for Combining Deep and Surface Machine Learning Techniques," Master's thesis, ETSI Telecomunicación, June 2016.
- [35] C. Strapparava, A. Valitutti et al., "WordNet Affect: an affective extension of wordnet," in *LREC*, vol. 4, 2004, pp. 1083–1086.
- [36] M. Chernyshevich, "Ihs-rd-belarus at semeval-2016 task 5: Detecting sentiment polarity using the heatmap of sentence," *Proceedings of SemEval*, pp. 296–300, 2016.

A.2.8 Senpy: A Pragmatic Linked Sentiment Analysis Framework

Title	Senpy: A Pragmatic Linked Sentiment Analysis Framework
Authors	Sánchez-Rada, J. Fernando and Iglesias, Carlos A. and Corcuera, Ignacio and Araque, Óscar
Proceedings	2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)
ISBN	978-1-5090-5206-6
Year	2016
Keywords	Algorithm design and analysis, API, application program interfaces, Big data, Big Data, Big Data sentiment analysis, emotion analysis, framework, linked data, NIF service specification, Ontologies, pragmatic linked sentiment analysis framework, reference open source implementation, Semantics, Senpy framework, sentiment analysis, Sentiment analysis, Syntactics
Pages	735–742
DOI	https://doi.org/10.1109/DSAA.2016.79
Online	https://ieeexplore.ieee.org/document/7796961
Abstract	Sentiment and emotion analysis technologies have quickly gained momentum in industry and academia. This popularity has spawned a myriad of service and tools. Due to the lack of common interfaces and models, each of these services imposes specific interfaces and representation models. Heterogeneity makes it costly to integrate different services, evaluate them or switch between them. This work aims to remedy heterogeneity by providing an extensible framework and an API aligned with the NIF service specification. It also includes a reference implementation, a first step towards a successful and cost-effective adoption. The specific contributions in this paper are: (i) the Senpy framework, (ii) an architecture for the framework that follows a plug-in approach, (iii) a reference open source implementation of the architecture, (iv) the use and validation of the framework and architecture in a big data Sentiment Analysis European project. Our aim is to foster the development of a new generation of emotion aware services by isolating the development of new algorithms from the representation of results and the deployment of services.

Senpy: A Pragmatic Linked Sentiment Analysis Framework

J. Fernando Sánchez-Rada, Carlos A. Iglesias, Ignacio Corcuera and Óscar Araque

Intelligent Systems Group

Universidad Politécnica de Madrid

{jfernando,cif}@dit.upm.es, {ignacio.cplatas,oscar.aiborra}@alumnos.upm.es

Abstract—Sentiment and emotion analysis technologies have quickly gained momentum in industry and academia. This popularity has spawned a myriad of service and tools. Due to the lack of common interfaces and models, each of these services imposes specific interfaces and representation models. Heterogeneity makes it costly to integrate different services, evaluate them or switch between them. This work aims to remedy heterogeneity by providing an extensible framework and an API aligned with the NLP Interchange Format service specification. It also includes a reference implementation, a first step towards a successful and cost-effective adoption. The specific contributions in this paper are: (i) the Senpy framework; (ii) an architecture for the framework that follows a plug-in approach; (iii) a reference open source implementation of the architecture; (iv) the use and validation of the framework and architecture in a big data sentiment analysis European project. Our aim is to foster the development of a new generation of emotion aware services by isolating the development of new algorithms from the representation of results and the deployment of services.

Keywords—sentiment analysis; emotion analysis; framework; linked data;

I. INTRODUCTION

Sentiment analysis is a blooming field of research and application, fueled by the popularity of social media and the need to make sense of collective opinions [1]. A vast number of sentiment analysis tools and services have emerged in recent years. Most of these tools and services use ad-hoc representation and schemas. This heterogeneity not only prevents reusing tools, but it also hinders the establishment of common terminology and models. Initiatives like NLP Interchange Format (NIF) [2] paved the way to standardization by publishing a semantic format and an API for NLP services. Thence, applications like the NIF combinator [3] appeared, demonstrating that a semantic format eases the integration of different services. Other works have applied this notion to multimodal sentiment analysis by extending NIF with existing and new ontologies [4]. The new ontologies for emotion representation enable a better and unambiguous annotation, as well as other interesting applications such as automatic mapping of emotions between different models (e.g. from Plutchik's categories to the Valence-Arousal-Dominance space). However, the concepts behind ontologies and linked data publishing are unfamiliar to both the linguistic community and developers. As a consequence, there are still few solutions in the field that use semantic technologies. This is a known problem that motivated the creation of JSON-LD [5].

The contributions of this paper are: (i) Senpy, a generic framework for NLP services based on the vocabularies NIF, Marl and Onyx; (ii) the architecture of a service in this framework; (iii) the reference implementation of the Senpy architecture, which follows a plug-in architecture and demonstrates the practical feasibility of the framework [6], as well as several plugins for custom algorithms and wrappers of popular services; (iv) the extensive use of the reference implementation in a big data sentiment in the context of a big data sentiment analysis platform and other research projects.

The ultimate goal of this work is to ease the adoption of the proposed linked data model for sentiment and emotion analysis services, so that services from different providers become interoperable. With this aim, the design of the reference implementation has focused on its extensibility and reusability. A modular approach allows organizations to replace individual components with custom ones developed in-house. Furthermore, organizations can benefit from reusing prepackaged modules that provide advanced functionalities, such as algorithms for sentiment and emotion analysis, linked data publication or emotion and sentiment mapping between different providers.

The rest of this paper is structured as follows. Section II introduces the main concepts behind Senpy and its linked data approach. Section III describes the architecture of the Senpy framework. Section IV describes the reference implementation of the framework. Section V illustrates how this architecture and existing tools can be used to develop and use an emotion analysis service; Lastly, Section VI presents our conclusions and future work.

II. BACKGROUND

A key aspect of Senpy is its linked data approach. Its model is based on the following specifications:

- Marl [7], a vocabulary designed to annotate and describe subjective opinions expressed on the web or in information systems
- Onyx [8], which is built on the same principles as Marl to annotate and describe emotions, and provides interoperability with Emotion Markup Language (EmotionML) [9]
- NIF 2.0 [2], which defines a semantic format and API for improving interoperability among natural language

processing services NIF follows a linked data principled approach so that different tools or services can annotate a text. To this end, texts are converted to RDF literals and an URI is generated so that annotations can be defined for that text in a linked data way. NIF offers different URI Schemes to identify text fragments inside a string, e.g. a scheme based on RFC5147 [10], and a custom scheme based on context. In addition to the format itself, NIF 2.0 defines a REST API for Natural Language Processing (NLP) services with standardized parameters.

The integration of these ontologies has been covered in previous works [4]. For the sake of clarity, Listing 1 provides an example of annotation by a sentiment analysis service. In particular, it consists of the analysis of a microblog post with the text “The example they used was really good, I really enjoyed it” and whose URL is `http://microblog.com/User1/Post1`. The service response shown in Listing 1 indicates that an opinion (:Opinion1) has been detected. The properties of the entity are shown as well. Finally, it provides details of the analysis, such as the algorithm used, its confidence, polarity range and provenance (using the PROV-O ontology [11]). Note that a query to the service has the following format: `http://{endpoint}?i={text}&prefix={prefix}`.

```
<http://microblog.com/User1/Post1#char=0,49>
rdf:type nif:RDF5147String, nif:Context;
nif:beginIndex "0";
nif:endIndex "75";
nif:isString "The example they used in their last paper
↳ was very clear, I really liked it";
marl:hasOpinion :Opinion1.
:Opinion1
rdf:type marl:Opinion;
marl:describesObject "paper";
marl:describesObjectPart "example";
marl:describesFeature "clarity";
marl:polarityValue "0.8";
marl:hasPolarity: marl:Positive;
prov:wasGeneratedBy :Analysis1.
:Analysis1
rdf:type marl:SentimentAnalysis;
marl:maxPolarityValue "1";
marl:minPolarityValue "-1";
marl:algorithm "dictionary-based";
prov:wasAssociatedWith http://www.gsi.dit.upm.es/.
```

Listing 1: NIF + Marl output of a service call `http://senpy.cluster.gsi.dit.upm.es/api?i=The example they used in their last paper was very clear, I really liked it&prefix=http://microblog.com/User1/Post1#`

III. FRAMEWORK

This section describes a framework for natural language processing services, with a special focus on sentiment and emotion analysis.

The main component of a sentiment analysis service is the algorithm itself. However, for the algorithm to work, it needs to get the appropriate parameters from the user, format the results according to the defined API, interact with the user when errors occur or more information is needed, etc. All this boilerplate of sorts, albeit essential for the service, is a burden on service developers. The situation is even worse when dealing with

different algorithms at the same time, which usually requires developing and deploying them separately. For this reason, Senpy proposes a modular and dynamic architecture that allows: i) implementing different algorithms in an extensible way, yet offering a common interface, ii) offering common services that facilitate development, so developers can focus on implementing new and better algorithms. Furthermore, it fosters the creation of common tools such as service validators, evaluation suites and testing tools.

The framework covers all the aspects of developing, publishing and using a sentiment analysis service. These aspects are grouped into layers. In addition to giving a clearer view of the components of a service, separating the framework in aspects serves another purpose: it later helps with transferring this modularity to its implementations. Finally, modular implementation fosters the creation of new services and functionalities by reducing the cost of adding new features and algorithms.

As of this writing, we have identified five different layers: the Analysis Layer includes the core NLP process and the libraries to connect it to the rest of the layers; the Semantic Layer deals with conceptual models and their integration; the Syntactic Layer handles issues such as formatting, serialization and input/output validation; the User Interface (UI) is the way in which users interact with services; the Evaluation Layer allows users to benchmark different algorithms; and the Service Administration Layer offers tools and information to control running services. Figure 1 depicts these layers and the main components within each of them. The rest of this section describe each layer in detail.

The Analysis Layer includes those components that are directly involved in generating new annotations for a given input. More specifically, it comprises the implemented analysis algorithms and the libraries used in the implementation that are responsible for integrating one particular algorithm with the rest of the components. For instance, a specific service may include one or more sentiment analysis algorithms to choose from, a Named Entity Recognition algorithm, a gender detection algorithm, etc. Each of these algorithms should be developed independently from the rest, and should contain only the logic that concerns the generation of new annotations. The interface between every algorithm and the rest of the layers is well defined. The set of libraries that implement this interface are the Senpy SDK, which is also part of the framework.

The Semantic Layer provides semantic consistency to the service and adapts the results from the Analysis Layer to every request. To exemplify the role of this layer, let us consider the case of an emotion analysis service. The Analysis Layer of this service would consist of at least one implementation of an emotion analysis algorithm. This algorithm generates annotations using Ekman’s six categories. In a traditional service, this would mean that the output of the service could only contain these categories. If an application requires a different representation, such as the VAD (valence, arousal, dominance) dimensional model, the conversion of the results is external to the analysis service. In a Senpy service, the Semantic Layer could include mappings to transparently adapt the annotations to the desired representation. In addition to mappings and conversion, the Semantic Layer could include other steps, such as validation and inference.

The lowest level of abstraction corresponds to the Syntactic Layer. Its role is to validate and adapt input from and output to the user. On the input side, it extracts all the necessary information from every request, and processes it so that it is understood by other layers. If there is an error in the request, such as missing parameters or wrong syntax, this layer communicates it to the user. When the output from other layers is ready to be sent back to the user, the Syntactic Layer formats it using the appropriate is to validate both the input and the output of the service, to process the input so that it can be understood by other layers, and to process the output so that it has the requested format and structure.

The User Interface (UI) Layer handles the interactions between users and the service. The way in which users make requests and receive the results back is different depending on the medium used. For instance, the same underlying analysis could be accessed through a Command Line Interface (CLI) and a web service (Web UI). This difference should be transparent to developers. Hence, the main task of the UI Layer is to gather requests from the user, forward them to the rest of the framework, and then adapt the output to the medium in use. Another element in this layer is the Playground aspect, which will be explained further in Section IV. The main idea behind it is that users want to experiment with new services before integrating them in their workflow or using them programmatically. The Playground is a simple UI that presents users with all available algorithms and options, and guides them through their use.

All previous layers cover functional aspects, i.e. developing a service and allowing users to make requests. The last two layers in this section cover aspects that do not concern users but developers and service administrators.

A key aspect of developing a new analysis algorithm is to evaluate it and compare it to others. The Evaluation Layer contains benchmarking and evaluation tools. Evaluation is facilitated by the fact that the framework imposes a common API. i.e., services of the same type will use the same annotation scheme and will be called in the same way. Using the common API and a set of gold standard corpora, it is possible to evaluate and compare different algorithms. The same concept applies to testing.

Lastly, the Service Administration Layer includes aspects useful to maintain a service and control its lifecycle. Some of its main functions would be: logging, which is used to control execution and find possible errors; resource manager, to control processing, memory and storage consumption; usage statistics, for an overview on how the service is being used; process monitoring, to control what tasks are running and when; and configuration manager, to view and change the parameters used in the service, such as activating or deactivating modules within the service.

IV. REFERENCE IMPLEMENTATION

Providing a reference implementation of the conceptual framework serves three main purposes. Firstly, it allows us to assess the feasibility and completeness of the framework. Secondly, it acts as a showcase of the purpose and the concepts behind the framework. Thirdly, it can be used as a reference or gold standard for future implementations.

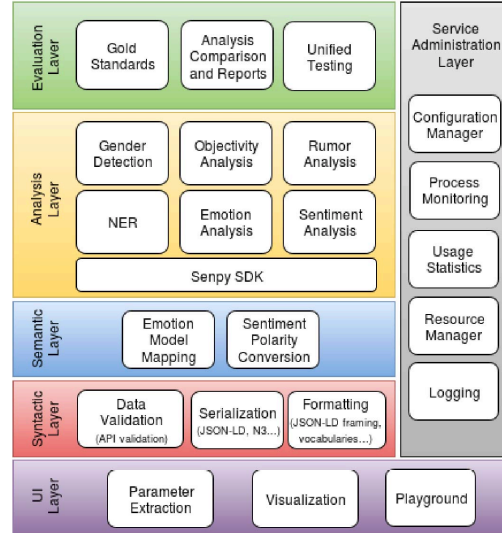


Fig. 1: Senpy framework. Each layer represents a functional block in a service.

The architecture of the reference implementation consists of two main modules: Senpy core, which is the building block of the service, and Senpy plugins, which contain the code for each analysis algorithm. The modularity of the architecture serves the overall goal of Senpy of providing seamless integration of different analysis algorithms while minimizing code duplication and development effort. Several plugins may coexist in the same service, accessing different resources and algorithms while benefiting from the nurturing environment of the common platform. Figure 2 depicts a simplified version of the processes involved in an analysis with Senpy. The following sections describe each component of the architecture in further detail.

The implementation is fully Open Source and published on GitHub¹, and a live demo is publicly available².

A. Core

As its name implies, the core of Senpy provides the main functionalities of the platform: an HTTP server/CLI interface, parameter extraction and validation, serialization of results using different formats and an abstraction and publication of results as Linked Data. It manages the lifecycle of plugins as well, orchestrating their execution and all interaction with the user. To better understand the features of the core, let us follow a typical analysis request from a user.

First of all, there are two ways in which a user may want to run their analysis: as a one-off local process or as a service. For

¹<http://www.github.com/gsi-upm/senpy>

²<http://senpy.cluster.gsi.dit.upm.es>

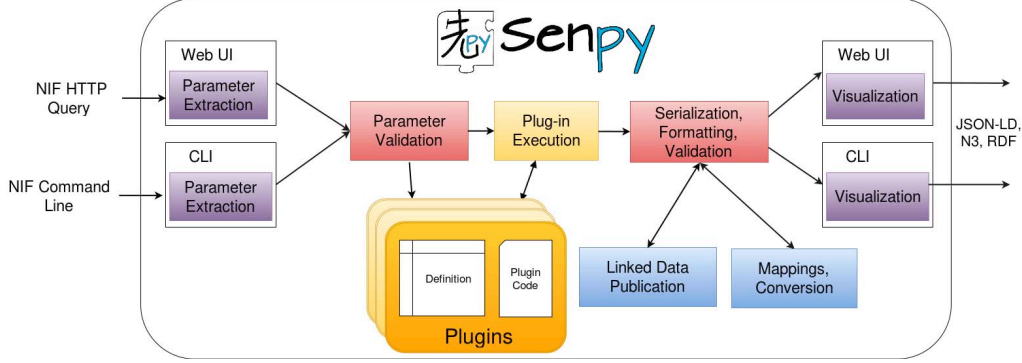


Fig. 2: Modules involved in an analysis with the reference implementation of Senpy

one-off commands, Senpy provides a command line interface (CLI), configurable via arguments. For long running processes or services, Senpy provides an HTTP server. In this case, users send their requests using HTTP queries to the server. Both the CLI and HTTP server use an API aligned with NIF, but using a JSON-LD representation and a JSON-schema by default. This difference makes it friendlier and more appealing to developers, as well as compatible with a wider range of tools. The API defines the parameters that are allowed (Table I), and is complemented by the extra parameters that each plugin declares in its definition (see Listing 2 for an example).

parameter	description
input(i)	serialized data (i.e. the text or other formats, depends on informat)
informat(f)	format in which the input is provided: turtle, text (default) or json-ld
outformat(o)	format in which the output is serialized: turtle (default), text or json-ld
prefix(f)	prefix used to create and parse URIs
emodel(e)	emotion model in which the output is serialized (e.g. WordNet-Affect, PAD, etc.)
minpolarity (min)	minimum polarity value of the sentiment analysis
maxpolarity (max)	maximum polarity value of the sentiment analysis
language (l)	language of the sentiment or emotion analysis
domain (d)	domain of the sentiment or emotion analysis
algorithm (a)	plugin that should be used for this analysis

TABLE I: Parameters of an Emotion or Sentiment analysis service using Senpy

Senpy uses these parameters in every request to extract all parameters from the request, and to warn the user whenever there are missing parameters.

If the basic arguments provided are correct, Senpy uses its selection algorithm to determine the plugin that will receive the request. Typically, users select the plugin manually using the `algorithm` parameter. Senpy will then check if the extra parameters defined in the selected plugin are met as well. If this validation succeeds, the plugin is asked to run an analysis, using the validated parameters.

Senpy leverages different ontologies (e.g. MarI, Onyx) to represent different types of information. For simplicity, the

main types of results as well as their required and optional properties have been defined using JSON schema. This means that results are provided in a documented format that can also be validated before passing them to the user. Plugins use these models to return their results.

Once the analysis is done, its results are further modified before they are returned to the user. First of all, values are transformed to fit the parameters specified by the user. For instance, when a plugin uses a sentiment value in the interval $(-1, 1)$, and the user requested a value in the $(0,)$ interval. This phase is very useful when dealing with emotions. Senpy has several mappings from dimensional models to categories, and vice versa. An example of this can be seen in Section V.

Lastly, Senpy generates the final results in the appropriate format, including metadata and proper URI identifiers, so it can be published as Linked Data.

For convenience, Senpy includes a web interface to test all available plugins: the Senpy Playground (Figure 3). The Playground lists all available services, and dynamically adds fields for every parameter they accept, such as language.

B. Plugins

The components in the Analysis Framework from Figure 1 are plugins in the implementation. Hence, each plugin represents a different analysis process. For instance, we may have a plugin for emotion analysis using WordNet-Affect, and a plugin for sentiment analysis using SVM and the Sentiment140 corpus. In future versions of the implementation we plan to extend plugins to also cover components in other layers of the architecture.

A plugin is defined by two elements: a definition file and the plugin code. The definition file can be written in JSON or YAML (a JSON superset), and has the `.senpy` extension. It contains important information about the plugin such as: name, version, location of the plugin code, parameters needed and attributes of the plugin. Listing 2 shows the description file of an example plugin, which we will use in Section V. In this description we see that the plugin accepts an extra parameter

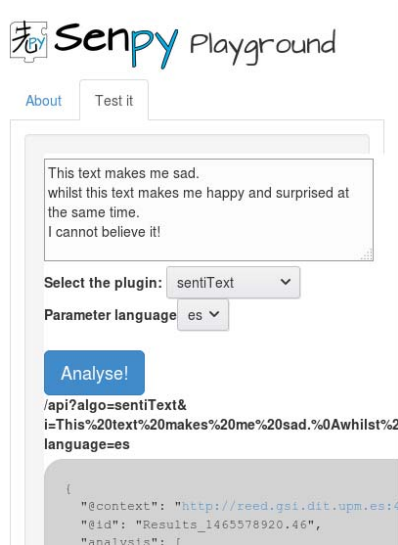


Fig. 3: Senpy Playground web interface.

in requests, language. When not provided, this parameter defaults to en. It also contains an attribute specific to this plugin, `default_value`, which determines the default value for words that are not found in ANEW.

Listing 2: Plugin definition using YAML

```
---
name: EmoTextANEW
module: emotextANEW
version: "0.1"
description: "Emotion classifier using rule-based
  ↳ classification."
extra_params:
  language:
  aliases:
  - language
  - 1
  default: en
  options:
  - es
  - en
required: true
default_value: [0,0,0]
```

The module attribute indicates the module that will be loaded. In this case, that module corresponds to the code in Listing 3. A Senpy (Senpy) plugin has to implement three methods: `activate`, for allocation of resources; `deactivate` for their release; and `analyse`, which takes the user-supplied parameters and performs the analysis. Resource allocation may seem needlessly complicated, but it is an important process when dealing with models that take gigabytes of memory. Section V covers the creation of a this specific plugin in more detail.

There are three main states in the lifecycle of a plugin: unloaded, inactive and active. Only active plugins can be used in requests. For a plugin to be active, two things have to

happen. First, the core has to load it. Once a plugin has been loaded, it gets in the inactive state. In this state, a plugin is listed by the core, but the variables necessary for analysis may not have been initialized. When a plugin is loaded, a special method in the plugin is called that initializes these variables. If the activation process is successful, the plugin enters the active state and can be used by users. If there are errors during the activation, the plugin remains inactive and all errors are logged. Active plugins can also be deactivated, which puts them in the inactive state again and should free up any variables that were initialized during activation.

To exemplify this process, let us consider the case of a sentiment analysis that uses a naive bayes classifier. This plugin requires a trained classifier to analyze text. However, when the plugin is loaded the classifier is not ready yet. The classifier is trained upon activation. Training may take a long time, depending on the size of our corpus and the features used. For this reason, changes of state are asynchronous operations for the core. When the activation finishes, our plugin will be automatically marked as active. Meanwhile, the core may handle other requests. Once our plugin is active, we can use it to analyze text. When our plugin is no longer needed, we may deactivate it. Deactivation will free up the memory used by the trained model.

Releasing resources when a plugin is not needed means that many resource hungry plugins can be loaded at the same time, and only activate them when they are needed. Resource initialization during activation also means that plugin variables will be consistent after it is activated. On the other hand, it also means that costly operations, such as training a model, have to be repeated several times. To avoid repetition and speed up start-up time considerably, Senpy ships with a special type of plugins that provide persistence. These persistent plugins have access to special variables that can be used to store the results of costly operations. When these variables are used, the plugin automatically checks the filesystem for a saved version of the variable. If it does, it loads the variable. If not, the plugin runs the appropriate operation and stores the value of the result in the filesystem.

The reference implementation Senpy has been validated by implementing wrappers to several available sentiment and emotion services, such as Sentiment 140 [12], Meaning-Cloud [13], Cogito [14], Vader [15] and Paradigma [16].

V. USE CASE

In this section we briefly cover the process of using Senpy in a real scenario. Our use case is a Big Data platform that uses a series of NLP services on social media. In fact, the scenario is a simplification of one of the pilots in the MixedEmotions project. This platform is made up of several modules from different parties. Some of them are existing NLP and sentiment analysis services. The rest of the modules depend on one or more of these analysis services. Integrating the different modules and their interfaces would require a big effort from every parties involved. Senpy reduces the cost of integration with its common interface and tools.

Figure 4 depicts the main elements. There are two parts in the platform of this use case. Firstly, there is a live brand monitoring dashboard. The dashboard shows the opinions of

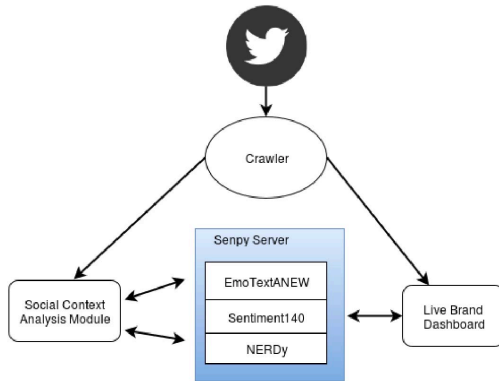


Fig. 4: Using Senpy in a Big Data Sentiment Analysis Platform

social media users about a brand. For this, it uses an external sentiment analysis service (sentiment140³), to annotate social media content with opinions. Secondly, there is a social context analysis module that finds the most influential users and content, as well as the evolution of emotion of all relevant users. The social context analysis module uses a NER (named entity recognition) module to gather only relevant content, and an emotion analysis module to annotate the emotions in the content. Social media content is provided by a separated module, labeled Crawler in Figure 4. A server running senpy will provide the NER, Sentiment and Emotion Analysis services.

Instead of accessing the external sentiment analysis service directly, we choose to use a custom sentiment analysis service in senpy that acts as a wrapper/proxy to the actual service. The main advantage of this approach is that it avoids having to deal with more than one API and schema for NLP service. Additionally, we gain access to all the extra capabilities of Senpy, such as the polarity conversion, benchmarking and service evaluation. Developing the wrapper is very straightforward and requires very little code⁴.

The emotion analysis analysis relies on the ANEW [17] lexicon to analyze the emotion in text, using a simple bag-of-words approach. Turning this code into a Senpy plugin is trivial, and merely a matter of implementing the analyse method. We have already covered the description file in Listing 2. The accompanying code is shown in Listing 3.

Listing 3: Code for the EmoTextANEW plugin

```
from os.path import dirname, abspath
from senpy.plugins import SenpyPlugin, tokenize
from senpy.models import Results, EmotionSet,
    Emotion, VAD
from emotextanew import Analyser

class EmoTextANEW(SenpyPlugin):

    def activate(self, *args, **kwargs):
        self._local_path = dirname(abspath(__file__))
```

³<http://www.sentiment140.com/>

⁴<https://github.com/gsi-upm/senpy/tree/master/senpy/plugins/sentiment140>

```
self._analyser = Analyser(self._local_path)

def deactivate(self, *args, **kwargs):
    del self._analyser

def analyse(self, params):
    r = Results.from_params(params)
    for i in r.entries:
        es = EmotionSet()
        e = Emotion()
        valence = 0
        arousal = 0
        dominance = 0
        for j in i.nif_isString:
            # Find V,A,D for each word
            v, a, d = self._analyser.get_vad(j)
            valence += v
            arousal += a
            dominance += d

        e[VAD.arousal] = a
        e[VAD.valence] = v
        e[VAD.dominance] = d
        es.onyx_hasEmotion.append(e)
        i.emotions = es
    return results
```

Since ANEW uses the VAD emotion model, that is what our plugin will use as well. Normally, this would mean users would need to use the VAD model themselves. However, since we are using Senpy to publish our service, we can make use of its additional features, such as mapping of emotion models. Emotion mapping can be used by setting the `emodel` parameter in the request. Listing 4 shows the response to a request using the WordNet-Affect model. Notice the addition of the emotion category (joy) based on the VAD dimensions. In particular, the conversion from VAD to WordNet-Affect categories is based on centroids [18]. The information about the centroids is displayed in the results, which together with the use of the provenance ontology makes the process transparent and repeatable.

Listing 4: Requesting an emotion analysis with a different emotion than the one provided in the plugin.

```
{
  "@context": "http://senpy.cluster.gsi.dit.upm.es/api/
    contexts/context.jsonld",
  "analysis": {
    {
      "analysis": {
        {
          "@id": "EmoTextANEW_0.1",
          "@type": "onyx:EmotionAnalysis",
          "onyx:usesEmotionModel": "onyx-anew:ANEWModel"
        },
        {
          "@id": "ANEW_Mappings_0.1",
          "@type": "onyx:EmotionAnalysis",
          "onyx:usesEmotionModel": "wnaffect:WNAModel"
        },
        "centroids": {
          "wnaffect:anger": {
            "A": 6.95,
            "D": 5.1,
            "V": 2.7
          },
          "wnaffect:disgust": {
            "A": 5.3,
            "D": 8.05,
            "V": 2.7
          },
          "wnaffect:fear": {
            "A": 6.5,
            "D": 3.6,
            "V": 3.2
          },
          "wnaffect:joy": {
            "A": 7.22,
```

```

        "D": 6.28,
        "V": 8.6
    },
    "wnaffect:sadness": {
        "A": 5.21,
        "D": 2.82,
        "V": 2.21
    }
}
},
"entries": [
{
    "emotions": [
        {
            "onyx:hasEmotion": [
                {
                    "prov:wasGeneratedBy": "EmoTextANEW_0.1",
                    "onyx-anew:arousal": 5.0,
                    "onyx-anew:dominance": 5.62,
                    "onyx-anew:valence": 6.23,
                },
                {
                    "prov:wasGeneratedBy": "ANEW_Mappings_0.1",
                    "onyx:hasEmotionCategory": "wn-affect:joy"
                }
            ]
        },
        {
            "language": "en",
            "nif:isString": "I am feeling excited"
        }
    ]
}
]
}

```

This section illustrates how easy it is to develop a new service from scratch and to integrate it in a bigger scenario. As shown in Listing 3, a plugin code is made up entirely of the `analyse` function, which almost perfectly matches the pseudocode of the algorithm. This succinct code provides a web service and a CLI tool that does parameter extraction and validation automatically. Furthermore, the platform provides additional features such as automatic linked data conversion and publication or mapping of emotions. Finally, the common interfaces and schemas provide loose coupling to the platform. This means that once a module is adapted for `senpy` to make use of a service, it can be made to use any equivalent service just by pointing it to a different endpoint.

VI. CONCLUSIONS AND FUTURE WORK

The sentiment and emotion analysis community would highly benefit from a common framework for service and language resources. Such a framework would ease adoption, development, integration and evaluation of services. A linked data approach further adds to these benefits, but its use needs to be transparent to users and developers.

This paper proposes a generic framework that combines both worlds and a reference implementation of that framework that is currently being used in MixedEmotions⁵, a European R&D project. The linked data model for sentiment and emotion services is based on the combination of NIF, Marl and Onyx vocabularies. Moreover, a number of parameters (e.g. min, max and e) have been defined following NIF Service specification so that sentiment and emotion service calls are interoperable.

We want to increase the adoption of the framework and to foster a community approach, where most plugins and features are provided by third parties. For this reason, we are currently working on easing the development of new plugins, and on making it possible to create plugins for any part of the

framework. Other lines of research would be the connection between different plugins (e.g. pipelining), the integration of the framework with other distributed and big data systems, and the addition of authentication and rate limiting.

In conclusion, the framework proposed in this paper has already proven useful in a multilingual sentiment analysis scenario. It has enabled the integration of multiple services from different parties and eased the creation of novel algorithms. This new approach paves the way for new testing and validation tools, as well as advanced capabilities such as deployment in high availability and cluster environments.

VII. ACKNOWLEDGEMENTS

This work has been partially funded by the European Union - Horizon 2020 (Industrial Leadership) through the MixedEmotions project (number H2020 644632). Part of this work has been carried out in the context of the MOSI-AGIL-CM research programme (grant S2013/ICE-3019, supported by the Autonomous Region of Madrid and co-funded by EU Structural Funds FSE and FEDER, and the SEMOLA project, funded by the Ministry of Economy and Competitiveness of Spain (TEC2015-68284-R).

REFERENCES

- [1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.
- [2] S. Hellmann, J. Lehmann, S. Auer, and M. Brümmer, "Integrating nlp using linked data," in *The Semantic Web-ISWC 2013*. Springer, 2013, pp. 98–113.
- [3] S. Hellmann, J. Lehmann, S. Auer, and M. Nitzschke, "Nif combinator: Combining nlp tool output," in *Knowledge Engineering and Knowledge Management*. Springer, 2012, pp. 446–449.
- [4] J. F. Sánchez-Rada, C. A. Iglesias, and R. Gil, "A linked data model for multimodal sentiment and emotion analysis," *ACL/IJCNLP 2015*, p. 11, 2015.
- [5] M. Lanthaler and C. Gütl, "On using json-ld to create evolvable restful services," in *Proceedings of the Third International Workshop on RESTful Design*. ACM, 2012, pp. 25–32.
- [6] E. Dalci, E. Fong, and A. Goldfine, "Requirements for gsc-is reference implementations. national institute of standards and technology," *Information Technology Laboratory*, 2003.
- [7] A. Westerski, C. A. Iglesias, and F. Tapia, "Linked opinions: Describing sentiments on the structured web of data," in *Proceedings of the Fourth International Workshop on Social Data on the Web (SDoW2011)*. CEUR, Oct. 2011, pp. 21–32.
- [8] J. F. Sánchez-Rada and C. A. Iglesias, "Onyx: A linked data approach to emotion representation," *Information Processing & Management*, vol. 52, no. 1, pp. 99–114, 2016.
- [9] M. Schröder, P. Baggia, F. Burkhardt, C. Pelachaud, C. Peter, and E. Zovato, "Emotionml – an upcoming standard for representing emotions and related states," in *Affective Computing and Intelligent Interaction*, ser. Lecture Notes in Computer Science, S. D'Mello, A. Graesser, B. Schuller, and J.-C. Martin, Eds. Springer Berlin Heidelberg, 2011, vol. 6974, pp. 316–325.
- [10] E. Wilde and M. Duerst, "URI Fragment Identifiers for the text/plain Media Type," Internet Engineering Task Force, Apr. 2008.
- [11] P. Missier, K. Belhajjame, and J. Cheney, "The w3c prov family of specifications for modelling provenance metadata," in *Proceedings of the 16th International Conference on Extending Database Technology*. ACM, 2013, pp. 773–776.
- [12] P. Chikersal, S. Poria, and E. Cambria, "Sentu: sentiment analysis of tweets by combining a rule-based classifier with supervised learning," in *Proceedings of the International Workshop on Semantic Evaluation, SemEval*, 2015, pp. 647–651.

⁵<http://mixedemotions-project.eu/>

- [13] I. Segura-Bedmar, P. Martínez, R. Revert, and J. Moreno-Schneider, "Exploring spanish health social media for detecting drug effects," *BMC medical informatics and decision making*, vol. 15, no. Suppl 2, p. S6, 2015.
- [14] Expert System, *COGITO Intelligence Platform*, 2015.
- [15] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.
- [16] J. F. Sánchez-Rada, M. Torres, C. A. Iglesias, R. Maestre, and R. Peinado, "A Linked Data Approach to Sentiment and Emotion Analysis of Twitter in the Financial Domain," in *Second International Workshop on Finance and Economics on the Semantic Web (FEOSW 2014)*, vol. 1240, May 2014, pp. 51–62.
- [17] M. M. Bradley and P. J. Lang, "Affective norms for english words (ANEW): Instruction manual and affective ratings," Technical Report C-1, The Center for Research in Psychophysiology, University of Florida, Tech. Rep., 1999.
- [18] S. M. Kim, A. Valitutti, and R. A. Calvo, "Evaluation of unsupervised emotion models to textual affect recognition," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. Association for Computational Linguistics, 2010, pp. 62–70.

A.2.9 Aspect Based Sentiment Analysis of Spanish Tweets

Title	Aspect Based Sentiment Analysis of Spanish Tweets
Authors	Araque, Oscar and Corcuera, Ignacio and Román, Constantino and Iglesias, Carlos Angel and Sánchez-Rada, J. Fernando
Proceedings	Proceedings of TASS 2015: Workshop on Sentiment Analysis at SEPLN co-located with 31st SEPLN Conference (SEPLN 2015), Alicante, Spain, September 15, 2015
ISSN	1613-0073
Volume	1397
Year	2015
Keywords	Machine learning, Natural Language Processing, Sentiment analysis,Aspect detection
Pages	29–34
Online	http://ceur-ws.org/Vol-1397/gsi.pdf
Abstract	This article presents the participation of the Intelligent Systems Group (GSI) at Universidad Politécnica de Madrid (UPM) in the Sentiment Analysis workshop focused in Spanish tweets, TASS2015. This year two challenges have been proposed, which we have addressed with the design and development of a modular system that is adaptable to different contexts. This system employs Natural Language Processing (NLP) and machine-learning technologies, relying also in previously developed technologies in our research group. In particular, we have used a wide number of features and polarity lexicons for sentiment detection. With regards to aspect detection, we have relied on a graph-based algorithm. Once the challenge has come to an end, the experimental results are promising.

Aspect based Sentiment Analysis of Spanish Tweets

Análisis de Sentimientos de Tweets en Español basado en Aspectos

Oscar Araque, Ignacio Corcuera, Constantino Román,
Carlos A. Iglesias y J. Fernando Sánchez-Rada

Grupo de Sistemas Inteligentes, Departamento de Ingeniería de Sistemas Telemáticos,
Universidad Politécnica de Madrid (UPM), España
Avenida Complutense, nº 30, 28040 Madrid, España
{oscar.aiborra, ignacio.cplatas, c.romang}@alumnos.upm.es
{cif, jfernando}@dit.upm.es

Resumen: En este artículo se presenta la participación del Grupo de Sistemas Inteligentes (GSI) de la Universidad Politécnica de Madrid (UPM) en el taller de Análisis de Sentimientos centrado en tweets en Español: el TASS2015. Este año se han propuesto dos tareas que hemos abordado con el diseño y desarrollo de un sistema modular adaptable a distintos contextos. Este sistema emplea tecnologías de Procesado de Lenguaje Natural (NLP) así como de aprendizaje automático, dependiendo además de tecnologías desarrolladas previamente en nuestro grupo de investigación. En particular, hemos combinado un amplio número de rasgos y léxicos de polaridad para la detección de sentimiento, junto con un algoritmo basado en grafos para la detección de contextos. Los resultados experimentales obtenidos tras la consecución del concurso resultan prometedores.

Palabras clave: Aprendizaje automático, Procesado de lenguaje natural, Análisis de sentimientos, Detección de aspectos

Abstract: This article presents the participation of the Intelligent Systems Group (GSI) at Universidad Politécnica de Madrid (UPM) in the Sentiment Analysis workshop focused in Spanish tweets, TASS2015. This year two challenges have been proposed, which we have addressed with the design and development of a modular system that is adaptable to different contexts. This system employs Natural Language Processing (NLP) and machine-learning technologies, relying also in previously developed technologies in our research group. In particular, we have used a wide number of features and polarity lexicons for sentiment detection. With regards to aspect detection, we have relied on a graph-based algorithm. Once the challenge has come to an end, the experimental results are promising.

Keywords: Machine learning, Natural Language Processing, Sentiment analysis, Aspect detection

1 Introduction

In this article we present our participation for the TASS2015 challenge (Villena-Román et al., 2015a). This work deals with two different tasks, that are described next.

The first task of this challenge, Task 1 (Villena-Román et al., 2015b), consists of determining the global polarity at a message level. Inside this task, there are two evaluations: one in which 6 polarity labels are considered (P+, P, NEU, N, N+, None), and another one with 4 polarity labels considered (P, N, NEU, NONE). P stands for *positive*, while N means *negative* and NEU is *neutral*. The “+” symbol is used for intensification of the polarity. It is considered that

NONE means absence of sentiment polarity. This task provides a corpus (Villena-Román et al., 2015b), which contains a total of 68.000 tweets written in Spanish, describing a diversity of subjects.

The second and last task, Task 2 (Villena-Román et al., 2015b), is aimed to detect the sentiment polarity at an aspect level using three labels (P, N and NEU). Within this task, two corpora (Villena-Román et al., 2015b) are provided: SocialTV and STOMPOL corpus. We have restricted ourselves to the SocialTV corpus in this edition. This corpus contains 2.773 tweets captured during the celebration of the 2014 *Final of Copa del*

rey championship¹. Along with the corpus a set of aspects which appear in the tweets is given. This list is essentially composed by football players, coaches, teams, referees, and other football-related concepts such as crowd, authorities, match and broadcast.

The complexity presented by the challenge has taken us to develop a modular system, in which each component can work separately. We have developed and experimented with each module independently, and later combine them depending on the Task (1 or 2) we want to solve.

The rest of the paper is organized as follows. First, Section 2 is a review of the research involving sentiment analysis in the Twitter domain. After this, Section 3 briefly describes the general architecture of the developed system. Following that, Section 4 describes the module developed in order to confront the Task 1 of this challenge. After this, Section 5 explains the other modules necessities to address the Task 2. Finally, Section 6 concludes the paper and presents some conclusions regarding our participation in this challenge, as well as future works.

2 Related Work

Centering the attention in the scope of TASS, many researches have experimented, through the TASS corpora, with different approaches to evaluate the performance of these systems. Vilares et al. (2014) present a system relying in machine learning classification for the tasks of sentiment analysis, and a heuristics based approach for aspect-based sentiment analysis. Another example of classification through machine learning is the work of Hurtado and Pla (2014), in which they utilize Support Vector Machine (SVM) with remarkable results. It is common to incorporate linguistic knowledge to this systems, as proposed by Urizar and Roncal (2013), who also employ lexicons in its work. Balahur and Perea-Ortega (2013) deal with this problem using dictionaries and translated data from English to Spanish, as well as machine-learning techniques. An interesting procedure is performed by Vilares, Alonso, and Gómez-Rodríguez (2013): using semantic information added to psychological knowledge extracted from dictionaries, they combine these features to train a

machine learning algorithm. Fernández et al. (2013) employ a ranking algorithm using bi-grams and added to this a skipgrams scorer, which allow them to create sentiment lexicons that are able to retain the context of the terms. A different approach is by means of the Word2Vec model, used by Montejo-Ráez, García-Cumbreras, and Díaz-Galiano (2104), in which each word is considered in a 200-dimensional space, without using any lexical or syntactical analysis: this allows them to develop a fairly simple system with reasonable results.

3 System architecture

One of our main goals is to design and develop an adaptable system which can function in a variety of situations. As we have already mentioned, this has taken us to a system composed of several modules that can work separately. Since the challenge proposes two different tasks (Villena-Román et al., 2015b), we will utilize each module when necessary.

Our system is divided into three modules:

- **Named Entity Recognizer (NER)** module. The NER module detects the entities within a text, and classifies them as one of the possible entities. In the Section 5 a more detailed description of this module and the set of entities given is presented, as it is used in the Task 2.
- **Aspect and Context detection** module. This module is in charge of detecting the remaining aspects -aspects that are not entities and therefore can not be detected as such- and the contexts of all aspects. In the Section 5 this module is described in greater detail since it is only used for tackling the Task 2.
- **Sentiment Analysis** module. As the name suggests, the goal of this module is to classify the given texts using sentiment polarity labels. This module is based on combining NLP and machine learning techniques and is used in both Task 1 and 2. It is explained in more detail next.

3.1 Sentiment Analysis module

The sentiment analysis module relies in a SVM machine-learning model that is trained with data composed of features extracted from the TASS dataset: General corpus for

¹www.en.wikipedia.org/wiki/2014_Copa_del_Rey_Final

the Task 1 and SocialTV corpus for Task 2 (Villena-Román et al., 2015b).

3.1.1 Feature Extraction

We have used different approaches to design the feature extraction. The reference document taken in the development of the features extraction was made by Mohammad, Kiritchenko, and Zhu (2013). With this in mind, the features extracted from each tweet to form a feature vector are:

- *N-grams*, combination of contiguous sequences of one, two and three tokens consisting on words, lemmas and stem words. As this information can be difficult to handle due to the huge volume of N-grams that can be formed, we set a minimum frequency of three occurrences to consider the N-gram.
- *All-caps*, the number of words with all characters in upper cases that appears in the tweets.
- *POS information*, the frequency of each part-of-speech tag.
- *Hashtags*, the number of hashtags terms.
- *Punctuation marks*, these marks are frequently used to increase the sentiment of a sentence, specially on the Twitter domain. The presence or absence of these marks (!?) are extracted as a new feature, as well as its relative position within the document.
- *Elongated words*, the number of words that has one character repeated more than two times.
- *Emoticons*, the system uses a Emoticons Sentiment Lexicon, which has been developed by Hogenboom et al. (2013).
- *Lexicon Resources*, for each token w , we used the sentiment score $score(w)$ to determine:
 1. Number of words that have a $score(w) \neq 0$.
 2. Polarity of each word that has a $score(w) \neq 0$.
 3. Total score of all the polarities of the words that have a $score(w) \neq 0$.

The best way to increase the coverage range with respect to the detection of

words with polarity is to combine several resources lexicon. The lexicons used are: Elhuyar Polar Lexicon (Urizar and Roncal, 2013), ISOL (Martínez-Cámara et al., 2013), Sentiment Spanish Lexicon (SSL) (Veronica Perez Rosas, 2012), SOCAL (Taboada et al., 2011) and ML-SentiCON (Cruz et al., 2014).

- *Intensifiers*, a intensifier dictionary (Cruz et al., 2014) has been used for calculating the polarity of a word, increasing or decreasing its value.
- *Negation*, explained in 3.1.2.
- *Global Polarity*, this score is the sum of the punctuations from the emoticon analysis and the lexicon resources.

3.1.2 Negation

An important feature that has been used to develop the classifier is the treatment of the negations. This approach takes into account the role of the negation words or phrases, as they can alter the polarity value of the words or phrases they precede.

The polarity of a word changes if it is included in a negated context. For detecting a negated context we have utilized a set of negated words, which has been manually composed by us. Besides, detecting the context requires deciding how many tokens are affected by the negation. For this, we have followed the proposal by Pang, Lee, and Vaithyanathan (2002).

Once the negated context is defined there are two features affected by this: N-grams and lexicon. The negation feature is added to these features, implying that its negated (e.g. positive becomes negative, +1 becomes -1). This approximation is based on the work by Saurí and Pustejovsky (2012).

4 Task 1: Sentiment analysis at global level

4.1 Experiment and results

In this competition it is allowed for submission up to three experiments for each corpus. With this in mind, three experiments have been developed in this task attending to the lexicons that adjust better to the corpus:

- *RUN-1*, there is one lexicon that is adapted well to the corpus, the ElhPolar lexicon. It has been decided to use only this dictionary in the first run.

- *RUN-2*, in this run the two lexicons that have the best results in the experiments have been combined, the ElhPolar and the ISOL.
- *RUN-3*, the last run is a mix of all the lexicon used on the experiments.

Experiment	Accuracy	F1-Score
6labels	61.8	50.0
6labels-1k	48.7	44.6
4labels	69.0	55.0
4labels-1k	65.8	53.1

Table 1: Results of RUN-1 in the Task 1

Experiment	Accuracy	F1-Score
6labels	61.0	49.5
6labels-1k	48.0	44.0
4labels	67.9	54.6
4labels-1k	64.6	53.1

Table 2: Results of RUN-2 in the Task 1

Experiment	Accuracy	F1-Score
6labels	60.8	49.3
6labels-1k	47.9	43.7
4labels	67.8	54.5
4labels-1k	64.6	48.7

Table 3: Results of RUN-3 in the Task 1

5 Task 2: Aspect-based sentiment analysis

This task is an extension of the Task 1 in which sentiment analysis is made at the aspect level. The goal in this task is to detect the different aspects that can be in a tweet and afterwards analyze the sentiment associated with each aspect.

For this, we used a pipeline that takes the provided corpus as input and produces the sentiment annotated corpus as output. This pipeline can be divided into three major modules that work in a sequential manner: first the NER, second the Aspect and Context detection, and third the Sentiment Analysis as described below.

5.1 NER

The goal of this module is to detect the words that represent a certain entity from the set of entities that can be identified as a *person* (players and coaches) or an *organization* (teams).

For this module we used the Stanford CRF NER (Finkel, Grenager, and Manning, 2005). It includes a Spanish model trained on news data. To adapt the model, we trained it instead with the training dataset (Villena-Román et al., 2015b) and a gazette. The model is trained with two labels: *Person* (PER) and *Organization* (ORG). The gazette entries were collected from the training dataset, resulting in a list of all the ways the entities (players, teams or coaches) were named. We verified the performance of the Stanford NER by means of cross-validation on the training data. With this, we obtained an average F1-Score of 91.05%.

As the goal of the NER module is to detect the words that represent a specific entity, we used a list of all the ways these entities were named. In this way, once the Stanford NER detect the general entity our improved NER module search in this list and decides the particular entity by matching the pattern of the entity words.

5.2 Aspect and Context detection

This module aims to detect the aspects that are not entities, and thus have not been detected by the NER module. To achieve this, we have composed a dictionary using the training dataset (Villena-Román et al., 2015b) which contains all the manners that all the aspects -including the entities formerly detected- are named. Using this dictionary, this module can detect words that are related to a specific aspect. Although the NER module already detects entities as players, coaches or teams, this module can detect them too: it treats these detected entities as more relevant than its own recognitions, combining in this way the capacity of aspect/entity detection of the NER module and this module.

As for the context detection, we have implemented a graph based algorithm (Mukherjee and Bhattacharyya, 2012) that allows us to extract sets of words related to an aspect from a sentence, even if this sentence has different aspects and mixed emotions. The context of an aspect is the set of words related

to that aspect. Besides, we have extended this algorithm in such a way that allow us to configure the scope of this context detection.

Combining this two approaches -aspect and context detection- this module is able to detect the word or words which identify an aspect, and extract the context of this aspect. This context allows us to isolate the sentiment meaning of the aspect, fact that will be very interesting for the sentiment analysis at an aspect level.

We have obtained an accuracy of 93.21% in this second step of the pipeline with the training dataset (Villena-Román et al., 2015b). As for the test dataset (Villena-Román et al., 2015b) we obtained an accuracy of 89.27%².

5.3 Sentiment analysis

The sentiment analysis module is the end of the processing pipeline. This module is in charge of classifying the detected aspects in polarity values through the contexts of each aspect. We have used the same model used in Task 1 to analyse every detected aspect in Task 2, given that the detected aspect contexts in Task 2 are similar to the texts analysed in Task 1.

Nevertheless, though using the same model, it is needed to train this model with the proper data. For this, we extracted the aspects and contexts from the train dataset, process the corresponding features (explained in Section 3), and then train the model with these. In this way, the trained machine is fed contexts of aspects that will classify in one of the three labels (as mentioned: positive, negative and neutral).

5.4 Results

By means of connecting these three modules together, we obtain a system that is able to recognize entities and aspects, detect the context in which they are enclosed, and classify them at an aspect level. The performance of this system is showed in the Table 4. The different RUNs represent separate adjustments of the same experiment, in which several parameters are controlled in order to obtain the better performance.

As can be seen in Table 4, the global performance obtained is fairly positive, as our

²We calculated this metric using the output granted by the TASS uploading page www.daedalus.es/TASS2015/private/evaluate.php.

Experiment	Accuracy	F1-Score
RUN-1	63.5	60.6
RUN-2	62.1	58.4
RUN-3	55.7	55.8

Table 4: Results of each run in the Task 2

system ranked first in F1-Score and second in Accuracy.

6 Conclusions and future work

In this paper we have described the participation of the GSI in the TASS 2015 challenge (Villena-Román et al., 2015a). Our proposal relies in both NLP and machine-learning techniques, applying them jointly to obtain a satisfactory result in the rankings of the challenge. We have designed and developed a modular system that relies in previous technologies developed in our group (Sánchez-Rada, Iglesias, and Gil, 2015). These characteristics make this system adaptable to different conditions and contexts, feature that results very useful in this competition given the diversity of tasks (Villena-Román et al., 2015b).

As future work, our aim is to improve aspect detection by including semantic similarity based on the available lexical resources in the Linguistic Linked Open Data Cloud. To this aim, we will integrate also vocabularies such as Marl (Westerski, Iglesias, and Tapia, 2011). In addition, we are working on improving the sentiment detection based on the social context of users within the MixedEmotions project.

Acknowledgement

This research has been partially funded and by the EC through the H2020 project MixedEmotions (Grant Agreement no: 141111) and by the Spanish Ministry of Industry, Tourism and Trade through the project Calista (TEC2012-32457). We would like to thank Maite Taboada as well as the rest of researchers for providing us their valuable lexical resources.

References

- Balahur, A. and José M. Perea-Ortega. 2013. Experiments using varying sizes and machine translated data for sentiment analysis in twitter.

- Cruz, Fermín L, José A Troyano, Beatriz Pontes, and F Javier Ortega. 2014. Building layered, multilingual sentiment lexicons at synset and lemma levels. *Expert Systems with Applications*, 41(13):5984–5994.
- Fernández, J., Y. Gutiérrez, J. M. Gómez, P. Martínez-Barco, A. Montoyo, and R. Muñoz. 2013. Sentiment analysis of Spanish tweets using a ranking algorithm and skipgrams.
- Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. pages 363–370.
- Hogenboom, A., D. Bal, F. Franciscar, M. Bal, F. De Jong, and U. Kaymak. 2013. Exploiting emoticons in polarity classification of text.
- Hurtado, Ll. and F. Pla. 2014. ELiRF-UPV en TASS 2014: Análisis de sentimientos, detección de tópicos y análisis de sentimientos de aspectos en twitter.
- Martínez-Cámara, E., M. Martín-Valdivia, MD Molina-González, and L. Ureña López. 2013. Bilingual experiments on an opinion comparable corpus. *WASSA 2013*, 87.
- Mohammad, Saif M, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 321–327.
- Montejo-Ráez, A, M.A. García-Cumbreras, and M.C. Díaz-Galiano. 2104. Participación de SINAI Word2Vec en TASS 2014.
- Mukherjee, Subhabrata and Pushpak Bhat-tacharyya. 2012. Feature specific sentiment analysis for product reviews. volume 7181 of *Lecture Notes in Computer Science*, pages 475–487. Springer.
- Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proc. of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- Sánchez-Rada, J. Fernando, Carlos A. Iglesias, and Ronald Gil. 2015. A Linked Data Model for Multimodal Sentiment and Emotion Analysis. 4th Workshop on Linked Data in Linguistics: Resources and Applications.
- Saurí, Roser and James Pustejovsky. 2012. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299.
- Taboada, Maite, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.
- Urizar, Xabier Saralegi and Iñaki San Vicente Roncal. 2013. Elhuyar at TASS 2013.
- Veronica Perez Rosas, Carmen Banea, Rada Mihalcea. 2012. Learning sentiment lexicons in spanish. In *Proc. of the international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey.
- Vilares, D., M. A. Alonso, and C. Gómez-Rodríguez. 2013. LyS at TASS 2013: Analysing Spanish tweets by means of dependency parsing, semantic-oriented lexicons and psychometric word-properties.
- Vilares, David, Yerai Doval, Miguel A. Alonso, and Carlos Gómez-Rodríguez. 2014. LyS at TASS 2014: a prototype for extracting and analysing aspects from Spanish tweets.
- Villena-Román, Julio, Janine García-Morera, Miguel A. García-Cumbreras, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña-López, editors. 2015a. *Proc. of TASS 2015: Workshop on Sentiment Analysis at SEPLN*, number 1397 in CEUR Workshop Proc., Aachen.
- Villena-Román, Julio, Janine García-Morera, Miguel A. García-Cumbreras, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña-López. 2015b. Overview of TASS 2015.
- Westerski, Adam, Carlos A. Iglesias, and Fernando Tapia. 2011. Linked Opinions: Describing Sentiments on the Structured Web of Data. In *Proc. of the 4th International Workshop Social Data on the Web*.