TRABAJO DE FIN DE GRADO

Título:	Desarrollo de un analizador de emociones en un entorno universitario
Título (inglés):	Development of an Emotion Analysis System in a University community
Autor:	Ignacio Corcuera Platas
Tutor:	Carlos A. Iglesias Fernández
Departamento:	Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:	Mercedes Garijo Ayestarán
Vocal:	Tomás Robles Valladares
Secretario:	Carlos Ángel Iglesias Fernández
Suplente:	Amalio Francisco Nieto Serrano

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE
INGENIEROS DE TELECOMUNICACIÓN

Departamento de Ingeniería de Sistemas Telemáticos
Grupo de Sistemas Inteligentes

TRABAJO DE FIN DE GRADO

DEVELOPMENT OF AN EMOTION ANALYSIS SYSTEM IN A UNIVERSITY COMMUNITY

IGNACIO CORCUERA PLATAS

Julio de 2014

Resumen

Este trabajo fin de grado contiene el resultado de un proyecto cuyos objetivos han sido desarrollar un sistema de análisis de sentimiento y evaluarlo con dos conjuntos diferentes. El primero está basado en la recopilación de un corpus de Twitter extraído por nosotros. En el otro, un corpus ha sido proporcionado por el taller de análisis de sentimiento TASS 2015.

La recopilación del corpus de Twitter se ha extraído a través de diferentes estrategias. Se ha diseñado un módulo basado en estas estrategias y ha estado capturando tweets durante varios meses. Esta recopilación se ha dividido en dos conjuntos: entrenamiento y testeo. El conjunto de entrenamiento se ha anotado manualmente.

El sistema de análisis de sentimiento está basado en una técnica de aprendizaje automático supervisado. Este sistema consta de un preprocesador y un extractor de características especializado en mensajes de Twitter.

El taller TASS 2015 consiste en dos tareas diferentes y para ambas tareas, se han presentado tres experimientos, eligiendo la mejor combinación de características extraídas por el sistema de análisis de sentimiento en el conjunto de entrenamiento. La tarea 1 consiste en la evaluación del sistema con dos conjuntos etiquetados diferentes, uno con seis etiquetas de sentimiento y el otro con cuatro etiquetas de sentimiento. La tarea 2 consiste en reconocer los diferentes aspectos que aparecen en un mensaje y clasificar su sentimiento.

El sistema de análisis de sentimiento se ha evaluado también con el conjunto de entrenamiento de Twitter. Una vez que ha sido entrenado, el conjunto de testeo se clasifica. Este conjunto se ha utilizado para el desarrollo de un dashboard interactivo basado en la plataforma Sefarad 2. 0.

Finalmente, juntamos las conclusiones extraídas de este proyecto, las tecnologías que hemos aprendido durante el desarrollo y las posibles líneas de futuro trabajo.

Keywords: Analisis de sentimiento, Twitter, tweets, características, TASS, Python, aprendizaje automático, polaridad, diccionarios, dashboard, MongoDB

Abstract

This graduate thesis collects the result of a project whose objectives are developing a sentiment analysis system and testing it with two different corpora. The first one is based on a Twitter corpus extracted by ourselves. In the other one, a corpus set is committed by the TASS 2015 evaluation workshop.

The Twitter corpus has been extracted following different strategies. A collector has been designed focused on these strategies, and has been capturing tweets for several months. This corpus has been divided into two sets: training and testing. The training set has been annotated manually.

The sentiment analysis system relies on a supervised machine learning technique. This system has a preprocessor and a feature extractor specializing on Twitter messages.

The TASS 2015 workshop consists on two different tasks and for both tasks, three experiments have been submitted, choosing the best combination of features extracted by the sentiment analysis system in the training set. Task-1 consist on evaluating the system with two different tagged corpus, one with six sentiment labels and the other one with four sentiment labels. Task-2 consist on detecting different aspects in a message and classifying their sentiment polarity.

The sentiment analysis system has been evaluated too with the Twitter training set. Once it has been trained, the test set has been classified. This set has been used for developing an interactive dashboard based on the platform Sefarad 2.0.

Finally, we gather the extracted conclusions from this project, the technologies we have learned during the development and the possible lines of future work.

Keywords: Sentiment analysis, Twitter, tweets, features, TASS, Python, machine learning, polarity, lexicons, dashboard, MongoDB

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Context

The term micro-blogging defines a service that allows users to send and publish short messages such as text, individual images or video links. The main and most popular feature is its simplicity and synthesizing, that is because in most micro-blogging systems, user's messages are delimited by the number of characters. In those characters you can do many things such as talking about what you are doing, interact with other users through private messages and replies, announce things, promote, make or maintain friendships and networking, finding jobs, etc. These messages publish by the user are displayed on your profile page and are also immediately sent to other users who have chosen the option of receiving them. The home user can restrict the sending of these messages only to members of his circle of friends, or allow all users to access(the public).

Several studies such as [6] have tried to analyse user behaviour on micro-blogging services and one in particularly, Twitter[1]. This service has become an important source of information, where people share what they observe in their surroundings, information about events, and their opinions about topics. This information is a reliable source of appraisals.

[1] https://twitter.com/

However, it is also one of the most challenging ones. Users are restricted to express their views in messages of up to 140 characters. This requires to build precise systems, since arguments are limited to one or two sentences. Moreover, Twitter presents a singular language with its own elements such as hashtags, user-name and special tags.

The principal problem it is how to manage this information, since a manual monitoring of the content generated by users is not viable. Thus, researchers have focused on this task. The field in charge of simplifying these shorts messages and translate them into the writer's emotion, sentiment or attitude, it is called Sentiment Analysis.

Sentiment Analysis[8] is a term used to talk about the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source information. In other words, Sentiment Analysis is used to look for the opinions in content and choosing the sentiment within those opinions. An opinion is an expression which consists of two key components:

- A topic, principal idea in which is based the opinion.
- A sentiment on the topic.

Being able to identify and extract those opinions about topics, events, or products is becoming an essential part of market analysis and reputation management systems.The vast majority of related works on this area are conducted in English e.g [12], due to it is the predominant language on the Internet. However, languages such as Spanish are also playing an important role.

All sentiment analysis tools rely, at varying degrees, on lists of words and phrases with positive and negative connotations or are empirically related to positive or negative comments. There are two types of Sentiment Analysis based on which kind of tool it has been used for developing it, machine learning and dictionary analysis. In this project we have used machine learning on the development of the system.

1.2 Project Goals

In the long term, this project aims at analysing the feelings expressed in the social media Twitter. With this aim, the project will develop a system for collecting related tweets and will develop a sentiment analysis system based on machine learning techniques and dictionary analysis. In addition, the project will provide a dashboard to understand the results of the analysis.

Among the main goals inside this project, we can find:

- Design a strategy for collecting tweets focused on the university environment.
- Build a collector which is able to communicate with Twitter(through his API) and a database for storing tweets, based on the strategy before.
- Perform a sentiment analysis for the TAS2015[2] competition.
- Perform a sentiment analysis of the tweets collected.
- Design a dashboard which will show the information and results related with the experiment.

1.3 Project phases

This section is going to describe the different phases faced on this project.

1. The first phase consists on a touchdown with Twitter, studying the composition and language used to write a tweet. Then, starting knowing the different approaches to extract tweets related to a topic. Finally, build a system capable of collecting and storing all the tweets.
2. The second phase consists on analysing the structure of the classifier and their functionality. Propose different strategies for extracting features of the tweet message and test them.
3. The third phase consists on performing the sentiment analysis system for the TASS-2015 competition and decide all the experiments that are going to be proposed for each of the tasks.
4. The fourth phase consists on annotating part of the corpus extracted in the second phase forming a training set. Once it is annotated, the classifier is going to be tested with this training set.
5. The fifth phase consists on classifying all the tweets and used them to generate an interactive dashboard showing information relevant to Twitter and the sentiment analysis.
6. The sixth phase consists on elaborating a conclusion of all the work realized and establishing a strategy for future work.

[2]http://www.daedalus.es/TASS2015/tass2015.php

1.4 Structure of this document

In this section we provide a brief overview of the chapters included in this document. The structure is as follows:

- ***Chapter 1*** Provides an introduction to the context in which this project is developed. Besides, it describes the main objectives to achieve once concluded.
- ***Chapter 2*** Describes the strategy used on capturing tweets and the collector module designed for this task.
- ***Chapter 3*** Provides an introduction to sentiment analysis and describes the sentiment classifier and its different parts.
- ***Chapter 4*** Brief view of the different method of evaluation and measure in sentiment analysis, and the development of the two use cases: TASS-2015 competition and Twitter.
- ***Chapter 5*** Conclusion and future work.

CHAPTER 2

Mining Twitter

2.1 Introduction

Twitter provides news about the person who is posting them: commentary on links, directed discussion, location information, status or any other content. Each user is able to monitor the tweets of other users, who will be listed in the profile, under the name "Following". Thus that user becomes Follower for them. Twitter has more than 100 million users who post nearly 340 million tweets per day.

Moreover, Twitter has been used to search for information for its large amount of social information and timeless response. This search functionality is used by users to look for specific information (e.g. events,weather,news), but the reason behind the search query could be totally different. According to Broder [1], the three main categories of search queries are navigational, informational and transactional. However, this functionality is limited by terms, expressions or keywords used by the user for seeking the information in Twitter.

The principal reasons behind a developer's decision to use Twitter or another micro-blogging in his web application are:

- Twitter is used by people to express their opinion or their feelings, providing a diversity of views about different topics.
- Twitter produces an enormous posting texts which are growing every day.
- Twitter does not distinguish between celebrities, politicians, sportsmen, etc; even if users belong to different ideology groups. Twitter has representation in many countries so it provides a global view about the topic.

For those reasons, Twitter is going to be used as a source of information about the topic project, providing a dataset with tweet related to them. The topic project is related to University, particularly to the "Escuela Técnica Superior de Ingenieros de Telecomunicación" (ETSIT). In this chapter we are going to present the technologies used on this part, the different strategies adopted to collect information, the process involved in it and a deep view of the tweets.

2.2 Enabling Technologies

2.2.1 Twitter Search API

The *Twitter Search API* relies on a REST API that allows us to query it against the tweets that have been published in a brief time and its behaviour is similar to the functionality of the Search feature available in Twitter mobile or web clients.

The best way to build a query is using the Twitter Search web browser [1], that provides a simple way to make a request for filtering data,to test if it is valid and to return matched Tweets. The query can have operators that modify its behaviour; the most common operators used are:

1. The use of quotation marks. If there are two or more words between a double quotation mark, the search will look for the tweets that contain the exact words. You can use an OR between two words for searching one word or another.
2. The symbol operator is used to seek for hashtags(#) or referencing people(@) on tweets. It is added in front of the word.
3. For tracking people tweets, the operator is "from:username", with the words that are looked for before it.

[1] https://twitter.com/search-home

4. The default operator will look for every word specified on the query.

Apart from making a query, there is a set of additional parameters that allows developers a better control of the search results. The most important parameters are mentioned below:

1. Result Type: this parameter allows us to select the kind of tweet that is sought. It can take two values: recent or popular, or even a mix of both.
2. Geolocalization: this parameter is not available in the API, but it can be used setting the geocode in the query. The first step is to choose a center, giving the corresponding latitude and longitude, and a radius, expressed in kilometres or miles. It will look for tweets that have a specific location and they can be converted from a geocode to a lat/log. This feature is only available when users have their location active (GPS) when they make a tweet.
3. Language: this parameter allows you to restrict the language in which tweets are written.
4. TimeLines: there a few parameters that are used to limit the amount of tweets returned to the user, such as:
   - until: it returns tweets that have been generated before the given date. The date format is YYYY-MM-DD.
   - Since-id: it returns results with an ID greater than the specified ID.
   - Max-id: it returns results with an ID less than or equal to the specified ID.

The result format that is returned from this API it is JSON.

2.2.2 MongoDB

MongoDB[2] is an open-source documented-oriented database written in C++. MongoDB relies on flexibility data model, high scalability and high performance. Data in MongoDB has a flexible schema. This schema is based on this hierarchy MongoDB-databases-collections-documents. A database is formed from collections and a collection is formed from documents.

A document is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects but they use a derivative format called BJSON[3]. The advantages of using documents are:

[2] http://www.mongodb.org/
[3] http://bsonspec.org/

1. Documents support native data types from other programming languages. This feature helps to exploit the data.

2. Documents can be embedded and exported into arrays, that reduces the need for expensive joins.

3. Dynamic schema supports fluent polymorphism.

The functions find() and findOne() provide simple ways to manipulate documents in a collection. In case it is wanted to filter specific documents in a collection, the query operator to fix a criteria is available.

Although MongoDB supports single-instance operation, production MongoDB deployments are distributed by default. Replica sets provide high performance replication with automated fail-over, while shared clusters make it possible to partition large data sets over many machines transparently to the users. MongoDB users combine replica sets and shared clusters to provide high levels redundancy for large data sets transparently for applications.

2.3 Strategies for collecting tweets

The first step to collect information about Twitter is to define a strategy to follow. The project's topic is focused on tweets related to the university, so the strategy has to be based on a topic searcher.

This section is going to provide a deeper overview through the different strategies proven for searching tweets from a specific topic, then it will present a solution to the case treaty in this project.

2.3.1 Hashtags

Hashtag is one of the tools employed by Twitter to bring some coherence to the large amount of information. The hashtag symbol, #, converts a word or a unspaced group of words in a search link used to refer a certain topic. Thus the content is organized and it gives the possibility to track topics based on their keywords hashtags. When choosing a hashtag it is important to be fully aware that the used keywords are specific and relevant to the subject being treated. Twitter provides a functionality, that helps the user to seek information related to a specific topic, showing all the tweets where the hashtag is mentioned.

Looking for a hashtag is a common way to find information about a topic but in the

project case it is useless. There are not common hashtags related to the topic and only a few users use hashtag with topic's keywords. One solution could be to unify users by a common hashtag, but it will take too long to do it.

2.3.2 Geolocation

Geolocation is not one of the principal features of Twitter, even though it is taking a more relevant function.The relation between where you are and the topic you are arguing about it is very close, as it is discussed here [2], and it is a good start-point. Moreover, geolocation is used for drawing social maps. It has several functionalities that can be employed for collecting information:

- Looking for potential followers analysing people that use Twitter around a place and a given topic.
- Doing geolocation researches about certain topics identifying the tweets that are relevant.

The use of the geolocation depends of three facts:

- Users have to activate the geolocation of the tweets, in order to show their location when a user publishes a tweet. Twitter stores the location in case users will use it again.
- It is possible to configure users's profile to show the location of the users.
- Another way is to add geographical data to users's profile.

Using geolocated tweets requires that a big majority of the users related to the topic have it activated, otherwise it will be a unsuccessful search.

2.3.3 Users and keywords

One of the most common ways to collect information of a certain topic is to identify users and keywords related to it. The users are the ones who discuss and comment about it, they use keywords to refer to the topic. Once they have been identified it will be easy to make a list with the keywords related to the topic and to expand the number of users that you have collected.

The problem is that Twitter has a large amount of information. Finding users or keywords will become difficult if you only use a few words to refer to a specific topic.

2.3.4 Conclusion

Following only one of these strategies is incomplete and some information will be missed so the best option is to try to mix them thus the information will be more complete, and it will be easy to make an accurate analysis.

The strategy of this project relies on *Query Expasion*. Query expansion is the process of reformulating a seed query to improve retrieval performance in information retrieval operations. This strategy has three principals points:

1. Collecting tweets related to the topic using the geolocation and Keywords strategy. This first step is to form a simple list of keywords related to the topic.

2. The second part consists on searching users whose followers are related to the topic. Then, we make a list of users related to the topic and complete it with some users from step one.

3. Finally there are two lists with users and keywords, you can start making and posting queries; collecting new information and back to step one with this information, making a more complete list of keywords and users.

2.4 Processing

Once we have established the strategy followed in this project, the next step is to create a process base on it. The process is written in Python and uses some Python libraries. This process could be resumed by two complementary tasks and the result of joining them. These two tasks consist on making the lists specified on the Strategy's section, one for users and the other one for keywords.

Using a Python tool called "twitter-tap" [4], the process started to collect some tweets using the geolocated property provided by the tweets. Once a few tweets were stored, they were separated in words, deleting the stop-words and other types of unnecessary words, and it got a word-count. An overview in that word-count list is a good start for making the keywords list, but only the root word is added to the final list. The reason is that the

[4] http://janezkranjc.github.io/twitter-tap/

Twitter search engine will look for all the coincidences with the word searched, expressed differently, if the root word is used in the search process, it will seek all the derivative words from it. For completing the list, it has been added some useful keywords.

Despite the keyword's list, the process builds another list of users. For making this list, some target users have been selected; this selection is based on the potential number of followers, related with the University, that those targets have. In this case it has been used a Python tool, "tweepy" [5], that provides the followers from every target. Then the target's followers were crossed, which means that if a follower is following two or more targets, it belongs to the University circle, and it will be added to the user list.

The final part, it is to mount the query. This query is going to look for every keyword of the list and if a user has said it, then the tweet will be stored. Twitter-tap provides a easy way to launch queries and to store the results in MongoDB. When the information is returned, it is possible to form a new list of keyword based on this new information and launch more queries, this is a re-feeding process. The image below shows a diagram of the process.

[5] http://www.tweepy.org/

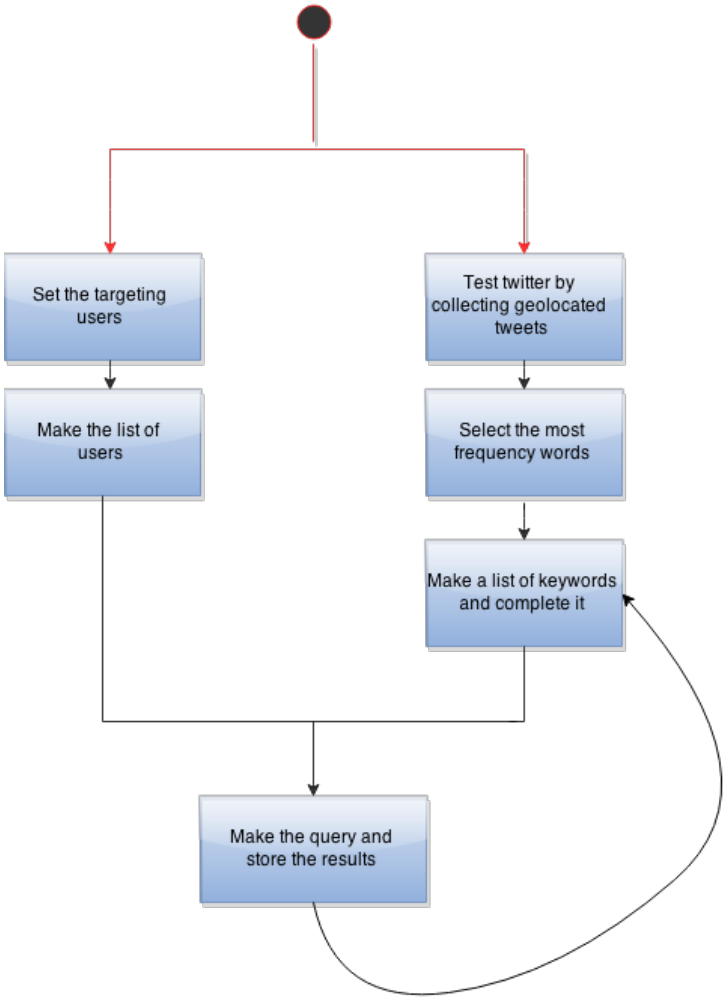

Figure 2.1: Diagram of the process

CHAPTER 3

Sentiment Analysis

3.1 Introduction

Sentiment analysis system based on machine learning techniques explores the construction and study of algorithms that can learn from and make predicts on data. Such algorithms operate by building a model from example inputs in order to make data-driven predictions or decisions, rather than following strictly static program instructions. Machine learning tasks are typically classified into two main categories:

- Supervised learning is based on the availability of a labeled corpus. In this corpus, each sentence is classified into one ore serveral categories or classes (e.g. positive, negative, etc.). Classes are called informational opposed to generating spectral classes unsupervised classification.

- Unsupervised learning are multivariate automatic classification algorithms in which individuals will closest classes are used to form gathering.

The selection of a supervised or unsupervised learning algorithms depends on the availability of a labeled corpus as well as in the desired learning task (e.g. discovery, clustering

or classification). In our case, we have decided to use a supervised model because we have the labeled corpus in both cases. The methodology used to design the supervised model is described in the image below.

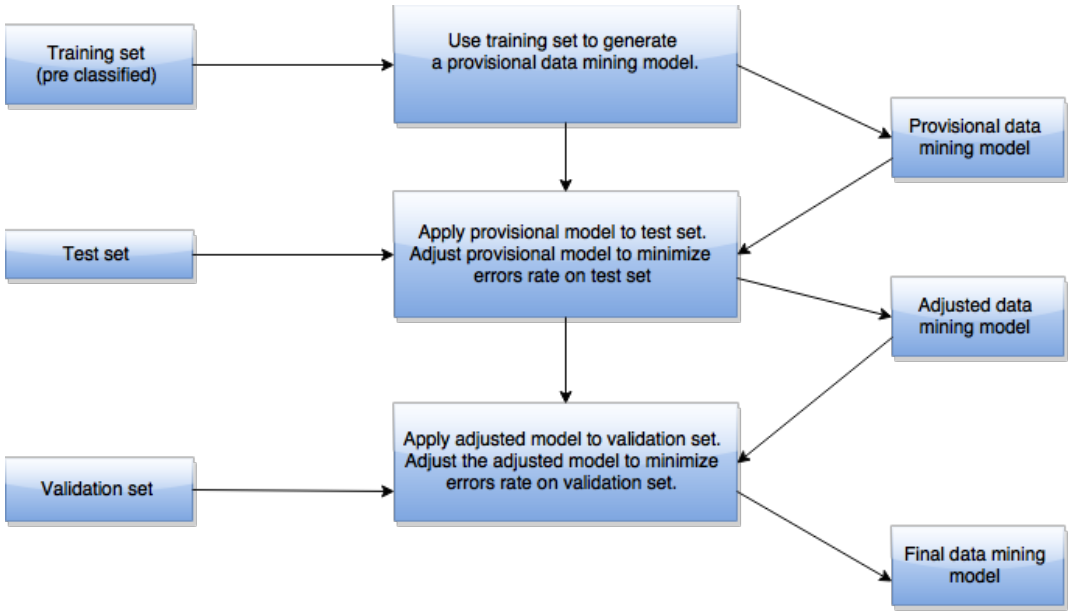

Figure 3.1: Diagram of the supervised modeling

Another important part of the sentiment analysis is feature extraction. Feature extraction involves reducing the amount of resources required to describe a large set of data. Although, when performing analysis of complex data one of the major problems stems from the number of variables involved. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy.

A preprocessing step it is always necessary, converting the input into data from which it is easy to extract the features that we have decided. This step is very useful in text classifier that usually needs to clean the text before processing it. In our case, Twitter has his own language to write text(emoticons ,urls ,hashtags ,user mentioned ,etc) so it is very useful to pre-process the text before extract features.

The most commonly algorithms used for sentiment analysis are:

- Naive Bayes are probabilistic classifiers based on applying Bayes theorem with strong independence assumptions between the features.
- Supporting Vector Machine (SVM) are supervised learning models that analyse data and recognize patterns. Given a set of supervised examples, an SVM algorithm builds

a model that assigns new examples into one category or the other, making it a non probabilistic binary linear classifier.

3.2 System architecture

The system relies on a supervised lexicon based system. In this section we are going to describe the different strategies used on developing the system. The process is divided in two principal stages: *training* and *testing*.

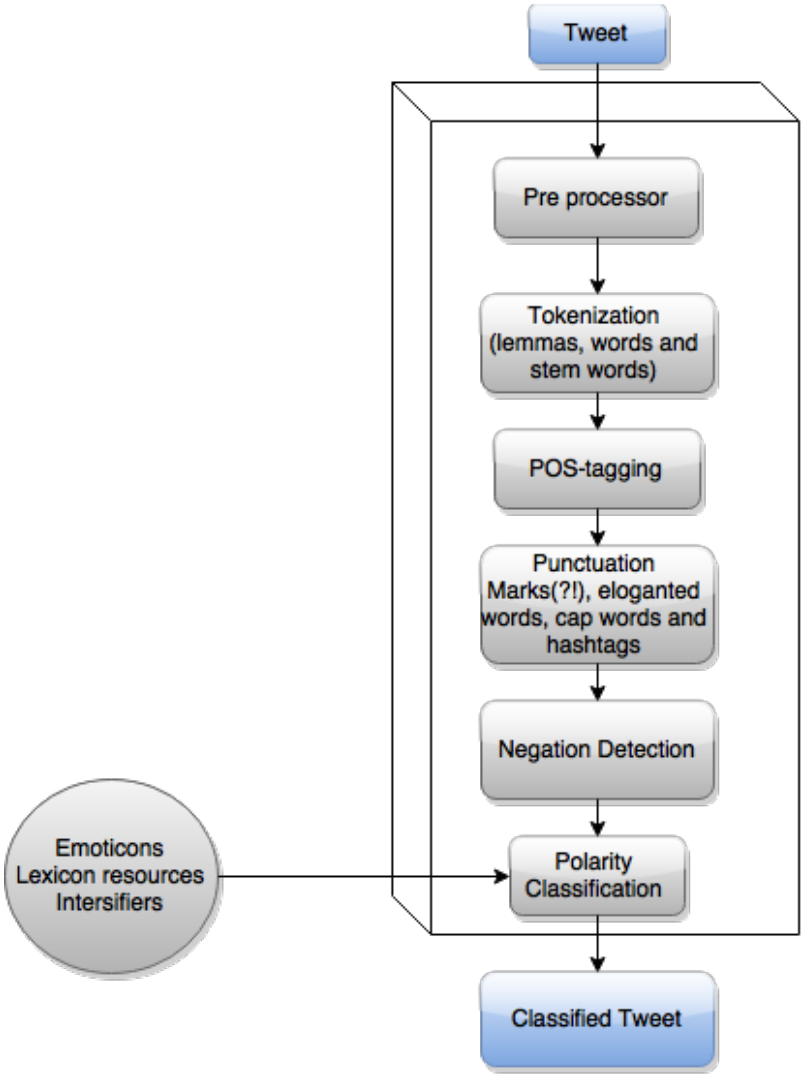

Figure 3.2: Diagram of the classifiers

3.2.1 Preprocessing

In this section, the text is preprocessed by two steps. First step eliminates all the URLs and numbers or digits that appear in the tweet. In addition, all the users and the syntax "RT @user :" are eliminated. This base-text is going to be used in the feature extractor

module. The second step is variable and depends on the type of feature that is going to be extracted. In case it is an emoticon, we are going to replace the duplicated punctuation signs that can be part of the emoticon, such as ":)))))". In case the text is going to be tokenized:

- All the hashtag terms are converted to normal words, meaning that the # sign is going to be eliminated.
- The elongated words are converted too, e.g. "largooo" is going to be converted to "largo". The problem with elongated words in Spanish, it is that they are part of the vocabulary such as "rr", "ll" and "cc". These exception were contemplated on the system design.
- Only punctuation signs that doesn't mean the end of a sentence, such as parenthesis, quotes, etc. are replaced.

3.2.2 Feature Extraction

In this part, we have used different approaches to design the feature extraction. We tokenized and extracted the part-of-speech information (POS) of the tweets, using a python module called *Pattern*[1]. The reference document taken in the preparation of this implementation was [10], which offers a broad explanation in the design of sentiment analysis classifiers. 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. This information is extracted from the training corpus, storing all the possible combinations that can appear on it. However, this information can be difficult to handle according to the huge volume of N-grams that can be formed. Therefore, it is set a minimum frequency of three occurrences to use the N-gram. The N-grams can consist of words, lemmas and stem words.
- *All-caps*, the number of words with all characters in upper cases that appears in the tweets, it is stored as a new feature.
- *POS information*, the frequency of each part-of-speech tag.
- *Hashtags*, the number of hashtags terms.

[1] http://www.clips.ua.ac.be/pages/pattern-es

- *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. Furthermore, the position where these marks appear could be a relevant information for detecting the polarity of the sentence. In that case, another feature is created in order to indicate whether the sentence ends with an exclamation or interrogation mark.
- *Elongated words*, the number of words that has one character repeated more than two times.
- *Negation*, this part is described in 3.3.1.
- *Intensifiers*, this part is described in 3.3.2.
- *Emoticons*, this part is described in 3.4.2.
- *Lexicon Resources*, the lexicon resources are described in 3.4.1. 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 $score(w) \neq 0$.
- *Global polarity*, this part is described in 3.4.3

3.3 Lexical Structures

3.3.1 Negation

To determine the polarity of an opinionated document, the first step is to divide the document into individual sentences and calculate their polarity, aggregating them, the document polarity is calculated. The polarity of a sentence is usually identified by certain words, phrases or expressions. However, their contextual polarities are dependent on the scope of each negation word or phrase preceding them, because their polarities might be flipped by negation words or phrases.

The polarity of a word changes if it is included in a negated context. First of all, it is necessary to detect this negated context. Satisfy this condition, a list of negation words has been used. Once the negative word has been detected, we have to decide the impact of it, in others words, how many of the tokens belonging to the sentence were affected by

this negation (negation context). Following [13], negated contexts have been defined as a segments of a text that start with a negation word (e.g. no, nunca) and end with one of the next punctuation marks: ',', ';', '.', ':', '!', '?' and the regular expressions '\n' and '\r' that imply the end of a sentence.

Once the negated context is defined, there are two features affected by this, N-grams and lexicon. Every N-gram that is going to be affected by the negation word, it is going to add the prefix "NEG-", e.g. "gustar" becomes "NEG_gustar". The "NEG_" prefix is also added to polarity lexicon features, e.g. "score_gustar" becomes "score_NEG_gustar". In case of polarity, the score of the lexical item involved in the negated context is going to change to the opposite (e.g. positive becomes negatives or +1 becomes -1). This approximation is based on [17].

3.3.2 Intensification

Although the negation affects the polarity of the sentence, there is another type of expressions may also change it, increasing or decreasing the polarity of a word next to them, called intensifiers. There are two types of intensifiers, one increase the polarity intensity of the nearest term, called amplifier and the other ones decrease it, called diminisher. For example, in the sentence "Ese cuadro es muy feo", the word "muy" is acting as an amplifier of the polarity of "feo", making its polarity more negative. On the other side, in the sentence "Ese cuadro es ligeramente feo", the word "ligeramente" is acting as a diminisher of "feo", making its polarity less negative.

One approach to this terms is to simple adding or subtracting a fixed amount to the polarity of the word, as it is describer here [7]. However, this approach doesn't contemplate that not all these terms modify the words with the same intensity,e.g. "poco" vs "menos".

SO-CAL[18], propose a different approach for this case. They have used a list of intensifiers, with each intensifying word having a percentage associated with it; positive for increase and negative for decrease. Once a intensifier is located they add or subtract this percentage to the polarity of the word. For example, "muy" have a percentage of 25% so the term "muy bueno" will amplify the polarity of the word "bueno" a 25%. This approach is going to be used on a future development of the system.

3.4 Identifying polarity in a sentence

This section is going to provide a broad view about the different resources used to extract features and the use of them.

3.4.1 Lexicon Resources

The best way to increase the coverage range with respect to the detection of words with polarity is to combine several resources lexicon. The lexicon resources used are:

- Elhuyar Polar Lexicon [19]. The ElhPolar polarity lexicon for Spanish was created from different sources, and includes both negative and positive words.
- ISOL [9]. ISOL is a list of sentiment words in Spanish independent of the domain. The list consists of 2,509 positive words and 5,626 negative words.
- Sentiment Spanish Lexicon (SSL) [20]. This resource contains two polarity lexicons. We have used the fullStrengthLexicon. It contains a Spanish sentiment lexicon which is more robust, as it leverages manual sentiment annotations from the OpinionFinder lexicon.
- SOCAL [18]. SOCAL lexicon is classified with a range value [-5,5]. To use this dictionary, it has been decided to use only words that have a polarity greater than 3 or fewer than -3.
- ML-SentiCON [3]. Multilingual, layered sentiment lexicons at lemma level. This resource contains lemma-level sentiment lexicons at lemma level for English, Spanish, Catalan, Basque and Galician. For each lemma, it provides an estimation of polarity (from very negative -1.0 to very positive +1.0), and a standard deviation. In this lexicon, it has been decided to use only words that have a polarity greater than 0.7 or fewer than -0.7

3.4.2 Emoticons

A smiley or emoticon is a sequence of ASCII characters representing a human face and express an emotion. These non-verbal cues are usually used to express different feelings or to amplify or modify the meaning of the message [22]. Therefore they become an important feature for use in analysis of feelings.

The system uses a Emoticons Sentiment Lexicon, which has been developed here[4]. This Lexicon contains a list with 447 emoticons aggregated in three classes: -1(negative),1(positive) and 0(neutral).

The features extracted from a tweet based on its emoticons, are:

- Number of emoticons that appear in the tweet.
- Polarity of each emoticon that appears in the tweet.
- Global score of all the emoticons that appear in the tweet. This score is going to be used later on the 3.4.3
- If a emoticon is the last token of the tweet, it will be pass as a feature.

3.4.3 Calculating global polarity of a sentence

In order to analyse the global sentiment of a sentence, on the one hand it has been analysed based on its emoticons and it has given a total score and on the other hand it has been analysed based on its sentiment-carrying words and it has given a total score. However, the sentiment score of a text GSV can be calculated as a combination of all the sentiment scores ζ_{e_i} of each emoticon e_i in the text and all the sentiment scores ζ_{w_i} of each sentiment-carrying word w_i:

$$GSV = \sum_{i=0}^{n} \zeta_{e_i} + \sum_{i=0}^{n} \zeta_{w_i} \tag{3.1}$$

This formulate is extracted from this article [4].

CHAPTER 4

Use cases

4.1 Introduction

A classifier has always two stages: one of training and the other one of testing. Once the classifier has been trained, the result of the testing stages could be divided in four different categories:

- True positives, the number of items correctly labelled to them belonging category (TP).
- False positives, the number of items incorrectly labelled to a category (FP).
- False negatives, the number of items not labelled as belonging to the positive class but should have been (FN).
- True negatives, the number of items correctly identified to the other categories (TN).

The different measures used on the evaluation of a binary classification are:

- Accuracy, it is the proportion of the true results (true positives plus true negatives)

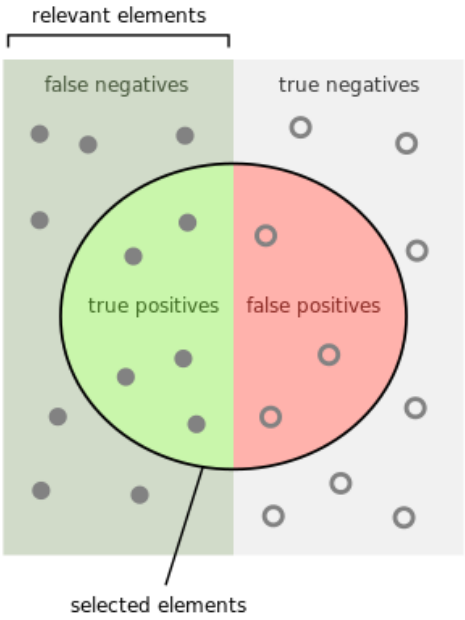

Figure 4.1: Diagram of the result in the testing stage of a classifier

among the total number of the cases classified.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- Precision, it is the proportion of the true relevant results (true positives) among the total number of relevant results (true positives plus true negatives).

$$Precision = \frac{TP}{TP + TN} \tag{4.2}$$

- Recall, it is the proportion of the true relevant results among the total number of items classified in that category (true positives plus false negatives).

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

- F1-score, it is a measure of a test's accuracy. It uses the precision and recall to calculate the score.

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4.4}$$

With this in mind, it has been decided to use the cross-validation for evaluating the classifier. Cross-validation consists on divided the train set into N-packets of the same length. Then, the classifier is trained with N-1 packets and tested with the other one. This process is repeated N times with the other packets as the test set. This method is useful

for testing your classifier with all the possible scenarios and improve your classifier. In this project, it has been used 10-cross-validation support by scikit-learn[1]

This section is going to provide an introduction to the different terms and components used to evaluate Sentiment analysis, and the two use cases: TASS 2015 and the Twitter case. Each of one cases have in common an evaluation and an analysis of errors section.

4.2 TASS 2015

TASS[2] is an experimental evaluation workshop for sentiment analysis and online reputation analysis focused on Spanish language, organized as a satellite event of the annual conference of the Spanish Society for Natural Language Processing (SEPLN). The aim of TASS is to provide a forum for discussion and communication where the latest research work and developments in the field of sentiment analysis in social media, specifically focused on Spanish language, can be shown and discussed by scientific and business communities. The main objective is to promote the application of state-of-the-art algorithms and techniques for sentiment analysis applied to short text opinions extracted from social media messages (specifically Twitter).

In the TASS edition of 2014, Lys team[21] presents a system which relies on a LiBLINEAR classifier trained with bags of words, lemmas, PoS-tags and several lexicons for Spanish-languages, obtaining good results. They sent two experiments, one of them has an information gain (IG) filter, which measure the relevance of each feature with respect to the class. Another approach is the one offered by JRC team[14] who proposed a system based on feature replacements: RPSR (Repeated Punctuation Signs Replacement), ER (Emoticon Replacemente) and AWR (Affect Word Replacement), replacing all of these features by fixed labels and combining it with the use of skip-grams. SINAIword2vec team present a solution based on vector representations of words and skip-gram, they have obtained a huge volume of text collected from the Wikipedia's articles and using a SVM classifier for the experiment. ELiRF-UPV team[5] winners of the competition, propose a strategy "one-vs-all" based on various SVM classifiers. They proposed 3 experiments, one with lemmas, another with words and a combination of n-grams of words and lemmas. SINAI-ESMA team[23] made two experiments, one with negation and the other with without it. This system recognized the negation terms that appears on the Twitter and it changes the polarity, but the problem with those changes, it is that the negation in Spanish language doesn't have

[1] http://scikit-learn.org/stable/

[2] http://www.daedalus.es/TASS2015/tass2015.php

to change the sentiment of the sentence. Elhuyar team[15] proposed 3 experiments using the combination of various lexicons, although they have based the syntax of the words on N-grams and they have treated the negation on the sentences.

4.2.1 General Corpus

The general corpus contains over 68 000 Twitter messages, written in Spanish by about 150 well-known personalities and celebrities of the world of politics, economy, communication, mass media and culture, between November 2011 and March 2012. Although the context of extraction has a Spain-focused bias, the diverse nationality of the authors, including people from Spain, Mexico, Colombia, Puerto Rico, USA and many other countries, makes the corpus reach a global coverage in the Spanish-speaking world.

The general corpus has been divided into two sets: training (about 10%) and test (90%). The training set will be released so that participants may train and validate their models. The test corpus will be provided without any tagging and will be used to evaluate the results provided by the different systems. Obviously, it is not allowed to use the test data from previous years to train the systems.

Each message in both the training and test set is tagged with its global polarity, indicating whether the text expresses a positive, negative or neutral sentiment, or no sentiment at all. A set of 6 labels has been defined: strong positive (P+), positive (P), neutral (NEU), negative (N), strong negative (N+) and one additional no sentiment tag (NONE).

In addition, there is also an indication of the level of agreement or disagreement of the expressed sentiment within the content, with two possible values: AGREEMENT and DISAGREEMENT. This is especially useful to make out whether a neutral sentiment comes from neutral keywords or else the text contains positive and negative sentiments at the same time.

4.2.2 Social-TV Corpus

This corpus was collected during the 2014 Final of Copa del Rey championship in Spain between Real Madrid and F.C. Barcelona, played on 16 April 2014 at Mestalla Stadium in Valencia. Over 1 million tweets were collected from 15 minutes before to 15 minutes after the match. After filtering useless information, tweets in other languages than Spanish, a subset of 2 773 was selected.

All tweets were manually tagged with the aspects of the expressed messages and its

sentiment polarity. Tweets may cover more than one aspect. Sentiment polarity has been tagged from the point of view of the person who writes the tweet, using 3 levels: P, NEU and N. No distinction is made in cases when the author does not express any sentiment or when he/she expresses a no-positive no-negative sentiment.

The Social-TV corpus was randomly divided into two sets: training (1 773 tweets) and test (1 000 tweets), with a similar distribution of both aspects and sentiments. The training set will be released so that participants may train and validate their models. The test corpus will be provided without any tagging and will be used to evaluate the results provided by the different systems.

4.2.3 Task 1: Sentiment Analysis at global level

This task consists on performing an automatic sentiment analysis to determine the global polarity of each message in the provided test sets (complete set and 1k set) of the general corpus. There will be two different analysis: one based on 6 different polarity labels (P+,P,NEU,N,N+,NONE) and another based on just 4 labels(P,N,NEU,NONE). Accuracy will be used for ranking the system.

Polarity	Number of Tweets	Percentage(%)
P+	1652	22.88
P	1232	17.07
N+	847	11.73
N	1335	18.49
NEU	670	9.28
NONE	1483	20.54

Table 4.1: This table shows the distribution of the training set over the different polarities.

They system used on this task, it is the one described in the previous chapter. This system have been trained by the training corpus of 6 labels and tested with a 10-cross-validation. The next table and graph shows the different distribution of accuracy using all the features described in the previous section, we have made different experiments pulling out features and classifying the results. The table is using terms like "only-" to talk about what kind of lexicon dictionaries is used for the analysis and "all-" to talk about what kind of feature is pulled out from the experiment. The graph shows the different values on the x axis are: all-f (all-features), all-l (all-lexicons, only-E (only-ElhPolar), only-I(only-ISOL), only-S(only-SSL), only-SO(only-SOCAL), only-ML(only-ML-Senticon),

all-n(all-ngrams), all-w(all-words), all-lm(all-lemmas), all-lm&w(all-lemmas&words), all-neg(all-negation), all-POS, all-enc(all-encoding), all-emo(all-emoticons)

Experiment	Accuracy	F1-Score	Precision	Recall
all-features	49.57	40.21	44.27	42.38
all-lexicons	44.99	37.07	41.36	38.91
only-ElhPolar	48.99	40.49	44.45	42.74
only-ISOL	47.25	39.11	43.19	41.42
only-SSL	45.83	37.77	41.7	39.7
only-SOCAL	46.21	37.8	41.67	39.85
only-ML-Senticon	44.99	37.06	41.35	38.9
all-ngrams	45.76	34.86	42.85	38.89
all-chars	47.78	38.77	43.88	41.385
all-lemmas	48.23	39.36	44.78	41.85
all-words	48.53	40.13	44.71	42.32
all-words&lemmas	47.77	38.47	44.35	41.27
all-POS	48.2	39.79	43.82	41.87
all-encoding	48.06	39.86	43.82	41.97
all-encoding	48.57	40.21	44.27	42.38
all-encoding	47.66	39.22	42.06	41.46

Table 4.2: All the system experiments with the different features.

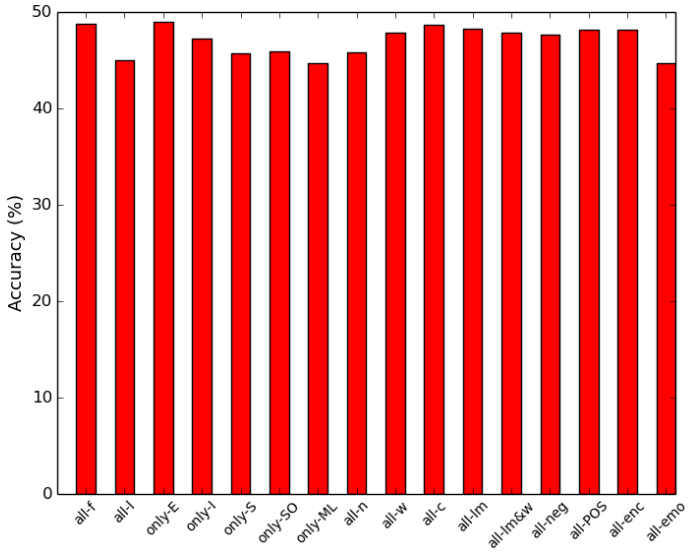

Figure 4.2: Graph of the different experiments with the features.

It is available to submit three experiment for each task in the TASS-2015 competition. With this in mind, three experiments have been developed attending to the result showed in the table:

- *RUN-1*, seeing the results it is easy to differentiate that there is one lexicon that is adapted well to the corpus, the ElhPolar lexicon. It is has been decided to use only this dictionary in the first run.

- *RUN-2*, in this run, it has been combined the two lexicons that have the best results in the experiments, the ElhPolar and the ISOL.

- *RUN-3*, the last run is a mix of all the lexicon used on the experiments.

These experiments have been used on the different test corpus (1k and the whole test) and the same experiments have been submitted for 6 labels and 4 labels. The best experiment has been the one who used the ElhPolar lexicon, that is because ElhPolar is specific lexicon for Twitter analysis. The RUN-2 and RUN-3 have the same distribution of punctuations (RUN-2 have a little improvement with respect to RUN-3). In conclusion, it is better to use a specialised lexicon that to combine several lexicons.

Category	Accuracy	Precision	Recall	F-Score
6labels	61.8	47.2	53.2	50
6labels-1k	48.7	44.1	45.1	44.6
4labels	69	55.8	54.1	55
4labels-1k	65.8	52.6	53.7	53.1

Table 4.3: Results of the RUN-1.

Category	Accuracy	Precision	Recall	F-Score
6labels	61	46.6	52.6	49.5
6labels-1k	48	43.5	44.4	44
4labels	67.9	55.9	53.3	54.6
4labels-1k	64.6	45.5	52.4	53.1

Table 4.4: Results of the RUN-2.

Category	Accuracy	Precision	Recall	F-Score
6labels	60.8	46.6	52.5	49.3
6labels-1k	47.9	43.2	44.3	43.7
4labels	67.8	55.9	53.3	54.5
4labels-1k	64.6	45.5	52.4	48.7

Table 4.5: Results of the RUN-3.

4.2.4 Task 2: Aspect-based sentiment analysis

This task consists on performing an automatic sentiment analysis to determine the polarity of each aspect detected on the tweet. This task is based on three principal subtasks. First, it is needed to identify the different aspects that appear in the tweet. Secondly, it is needed to identify the impact or range (number of words affected) of the aspect in the tweet. At last, it is needed to classify the aspect and give it a polarity.

The first two subtasks have been developed by the GSI("Grupo de Sistemas Inteligentes"). The method used for identifying aspects is based on Named-entity recognition, that is a

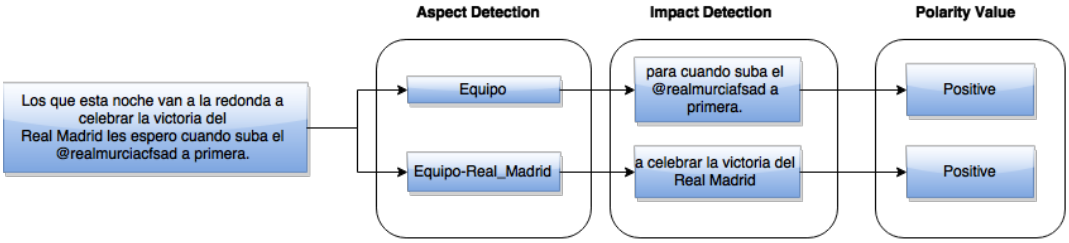

Figure 4.3: Graph of the task2 design.

subtask of information extraction that seeks to locate and classify elements in text into predefined categories such as the names of persons, organizations, locations, etc. Two methods have been used for identifying the range of the aspect: setting a window of one or more words near to the word detected as an aspect, or if the aspect is the first or the last select all the words at the beginning or at the end.

In the last subtask, the system used in the classification is the same that described in the previous chapter. In the Task2 is permitted to submit three experiments too, it has been decided to use the same runs of the sentiment classifier as in the Task1 4.2.3.

Experiments	Accuracy	F1-Score	Precision	Recall
RUN-1	63.5	63.7	57.8	60.6
RUN-2	62.1	62.2	55.1	58.4
RUN-3	55.7	59.3	52.7	55.8

Table 4.6: Results of the three experiments in the Task2

4.2.5 Errors

This section is going to provide a simple view of the errors that the classifiers have when they are training a corpus set. These errors can be classified into three sets: ambiguous message, human errors and system's errors.

An ambiguous message is the ones produced when the words expressed in the message haven't had a strong polarity to determinate the polarity that has been annotated for the message. For example:

-Qué frío (umbrella symbol)(lighting symbol)

This message has a negative polarity but our system has decided that its corresponding polarity is none, this is a case where the message's polarity is limited by their features.

Maybe the symbols identified it as a negative context but for the classifier is really complicated to determinate it.

A human error are the ones produced when the annotation that has been given to the message it is not clear enough. For example:

-Y digo que lo mejor del gobierno extremeño está en la Consejera de Educación y Cultura, a la que le atribuyen la culpa de la UEX sin razón

This message has a positive polarity but our system has decided that its corresponding polarity is strong negative, this is case where the message has been wrongly annotated and the polarity is not well defined.

System's errors, these errors can be divided into different sets:

- Negation errors, these errors are the ones produced by the use of negations as a feature. In the Spanish language, not all the negations change the polarity of the words nearest to them. For example:

  -Ultimo dia de 'cole' esta semana, nada mejor que despedirlo con mucha marcha en @mananascuatro con @mrtfernandez http //t.co/NCu0pecV

  This message has a positive polarity but our system has decided that its corresponding polarity is none, that is because the word "nada" is acting as a negation feature and it is changing the sentiment polarity.

- Unseen words and expressions, these errors are the ones produced by the used of words or expressions that don't belong to the language that is being used to training the classifier.For example:

  -EL LUNES UN AÑO EN TWITTER@TwBirthday: @julia_otero Happy 1st TwBirthday You've been around since 09 April 2011 http //t.co/dRylZ9c3

  This message has a strong positive value but our system has decided that its corresponding polarity is none, that is because it doesn't recognize the words "Happy" and "Birthday" that means positive.

- Other errors, these errors are the ones produced by the features detected on the message, in other words, the system decided to classify into a different polarity a message because the features are indicated that polarity. For example:

  -A las 3 en Antena3, ZP deja a Rajoy la aprobación de la ley antidescargas en Internet, después de muchas discusiones entre ministros

This message has a neutral polarity but our system has decided that its corresponding polarity is negative, that is because the words “dejar” and “discursiones” are negatives words.

4.2.6 Evaluation of the Results

This section is going to describe the result obtaining in the different tasks of the TASS 2015 workshop. The total number of teams that have participated in the workshop is twelve. In the Task1, we have been jumping between the third and the fourth position and in the task2, we have achieved the second position. But if we look at the global position, we have obtained good results. In both tasks we are above the majority of the participant teams with results above the average. The future work lines are included in the last chapter. Another future line work that could be contemplated, is to check all the papers and sentiment analysis systems from the other teams and learn from their work. The next table shows the best experiment submitted by each team.

Table 4.7: Results in the TASS 2015 workshop

Group-Experimt	Task1-6Full	Task1-61k	Task1-4Full	Task1-41k	Task2-ST
LIF	65.4	51.6	72.5	69.2	
ELiRF	64.8	47.6	71.2	64.5	65.5
GTI-GRAD	59.2	50.9	69.5	67.4	
GSI	**61.8**	**48.7**	**69**	**65.8**	**63.5**
LyS	56.8	43.4	66.4	63.4	61
DT	56	40.7	62.5	60.1	
CU	49.5	41.9	59.7	60	
INGEOTEC	48.8	43.1	61.3	59.5	
SINAI_wd2v	47.4	38.9	61.9	64.1	
TID-spark	46.2	40	59.4	64.9	63.1
UCSP	27.3		61.3	62.6	
GAS-UCR	34.2	33.8	44.6	55.6	

4.3 Twitter Analysis

4.3.1 Corpus annotation

With the strategy and the system proposed in previous chapters, it has been collected over four thousand tweets related to the University environment. This set has been divided into a training set and a testing set. The training set has 498 tweets. However, this corpus is not annotated and first of all we have to annotate it, for this task it has been used a tool called "WebAnno[3]".

WebAnno is a general purpose web-based annotation tool for a wide range of linguistic annotations including various layers of morphological, syntactical, and semantic annotations. Additionaly, custom annotation layers can be defined, allowing WebAnno to be used also for non-linguistic annotation tasks. It has been created a custom layer for the sentiment analysis with three tags: positive, negative and neutral.

The user interface for annotating is very easy to manipulate. You have the left side with the text you want to annotate and the right side with the different layers you can use to annotate it. You select the part of the sentence that you want to annotate, select the sentiment, category or topic and click on annotate. This user interface simply the annotation task.

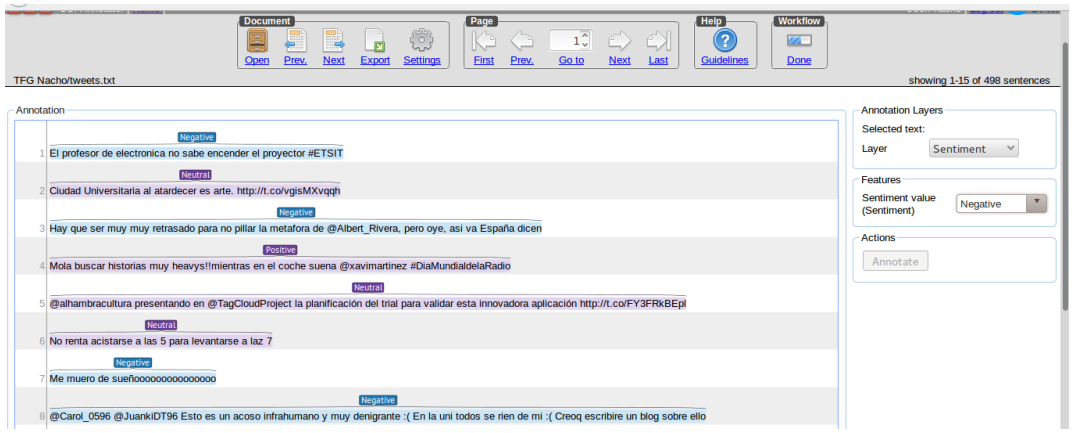

Figure 4.4: WebAnno user interface

But WebAnno does not have any comfortable format to export the result. For that reason a converter between TSV WebAnno[4] and NLP Interchage Format(NIF)+[Marl[5]]

[3]https://code.google.com/p/webanno/
[4]https://en.wikipedia.org/wiki/Tab-separated_values
[5]http://www.gsi.dit.upm.es/ontologies/marl/

has been used. The converter used for this task is called docon[6]. This tool will take several input formats and translates them to semantic formats. It focuses on translating corpora to the NIF+[Marl], using JSON-LD. In order to satisfy the parameters needed by this tool, a template has been designed to translate the format TSV WebAnno to the NIF+[Marl]. Once the training corpus is in a format easy to manipulate, we can start to train our sentiment classifier.

4.3.2 Sentiment Analysis

4.3.2.1 Evaluation of the training corpus

In this case, it has been decided to use only the ElhPolar lexicon dictionary because it is the most adaptable lexicon to a Twitter corpus. The training set is provided by WebAnno and annotated manually, categorizing the set with three labels: Negative,Positive and Neutral. The distribution of the polarities in the training corpus is described in the next table.

Polarity	Number of Tweets	Percentage(%)
P	157	31.53
N	195	39.15
NEU	146	28.94

Table 4.8: This table shows the distribution of the training set over the different polarities.

The classifier have been evaluated following the same criteria as the task1 from TASS competition. Looking the result from the table, it is possible to see that with a small corpus, the most important feature is the one extracted from the dictionary. Moreover, if the classifier does not use any lemmas,chars or words the global score is not affected at all and in some case the global score is increased. The negation is another feature that does not affected too much to the global score due to it affects the ngrams and the polarity words.

Sentiment140 is a tool that allows you to discover the sentiment of a brand, product, or topic on Twitter. This tools relies on:

- They use classifiers built from machine learning algorithms.
- They are transparent in how the tweets were classified. Other sites do not show you the classification of individual tweets.

[6]https://pypi.python.org/pypi/docon/0.1.3

Experiments	Accuracy	F1-Score	Precision	Recall
all-features	72.43	71.25	73.03	71.43
all-ElhPolar	68.7	66.88	69.32	67.43
all-ngrams	71.86	70.83	72.56	70.93
all-chars	72.71	71.68	73.33	71.81
all-words	72.43	71.25	73.03	71.24
all-lemmas	73.02	71.96	73.58	72.03
all-lemmas&words	72.99	72.03	73.33	72.12
all-negation	73.24	71.93	74.08	72.08
all-POS	72.17	71.48	72.62	71.47
all-encoding	71.26	70.19	71.53	70.5
all-emoticon	72.43	71.25	73.03	71.42

Table 4.9: All the system experiments with the different features.

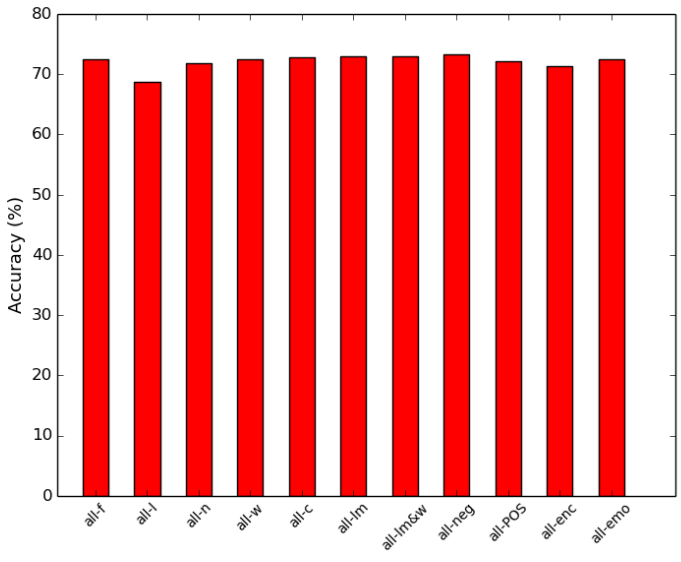

Figure 4.5: Graph of the different experiments with the features.

This tool allows users to use it as a simple REST api. You send the text and them it is classified: 0 for negative, 2 for neutral and 4 for positive. The languages supported by

this tool are English and Spanish. We are going to use this tool in order to compare it with our classifier. The results show that in the set of 498 tweets, only 167 were predicted correctly, having only a 33.53% accuracy. We must bear in mind that our classifier has been specially trained for that corpus , so the results from Sentiment140 are not too relevant. Also the training set is very small, only 498, making it very difficult to reach any classifier can achieve a significant percentage.

4.3.2.2 Errors

In this section we are going to take a look around all the possible errors that could appear in this classifier. The most important error is that we are trying our classifier with only a set of *498 tweets*, this is a very small corpus. most of features are not relevant for the classifier. Using a more bigger corpus increase the capability of predicting result and the classifier will be better trained for the sentiment analysis. A brief explanation of the errors is on the section above 4.2.5, here we are only to locate some of those errors on this training corpus.

-De vuelta a mi vida de mierda... Viva!!!! #UMA #ETSIT

In this case, the system has an error in recognising a pattern that means a respectively polarity (negative), although the positive word Viva and the exclamation marks could have made the classifier to mix up the polarities. This error is categorized as a system error.

-Origen, ese peliculón que no me cansaré nunca de ver

Now the system have made a mistake, classifying this tweet as negative, when it is neutral. That is due to the negations that appear in the tweet that have made the classifier to mix up the polarities. This error is categorized as a negation system error.

-Cuando sales de un examen final...

This is a typical ambiguous message error due to the lack of information that appear on it. The system has predicted this message as negative but the polarity is neutral. This could be understood as a negative message because the context of “sales de un examen” as a negative context or neutral because it does not show any more information.

Through this training corpus have been annotated by us, it is really complicated to localize any human errors corresponding to a bad annotation.

4.3.3 Dashboard

A dashboard had been development using Web Components[7]. This technology is composed of four complementary elements:

- Custom Elements, this specification describes the method for enabling the author to define and use new types of DOM elements in a document.
- HTML IMPORTS, it is a way to include and reuse HTML documents in others.
- TEMPLATES, this specification describes a method for declaring inert DOM subtrees in HTML and manipulating them to instantiate document fragments with identical contest.
- SHADOW DOM, this specification describes a method of establishing and maintaining functional boundaries between DOM trees and how these trees interact with each other within a document, thus enabling better functional encapsulation within the DOM.

This dashboard has been created based on the platform Sefarad 2.0[16] and has implemented a lot of widgets that are automatically refreshed with the values selected on the other widget, it is dynamically dashboard.

First of all, we have four different widgets that show us the information contained in the tweet. On the one hand we have a doughnut chart showing the percentage of neutral,positive and negatives tweets that are in the set. The next two graphs are bar that shows the information related to the retweeted and favourites of the tweet, showing how many time a tweet have been retweeted or made favourite. The last bar graph shows a representation of the days of the weeks and the amount of tweets that have been sent those days.

[7] http://webcomponents.org/

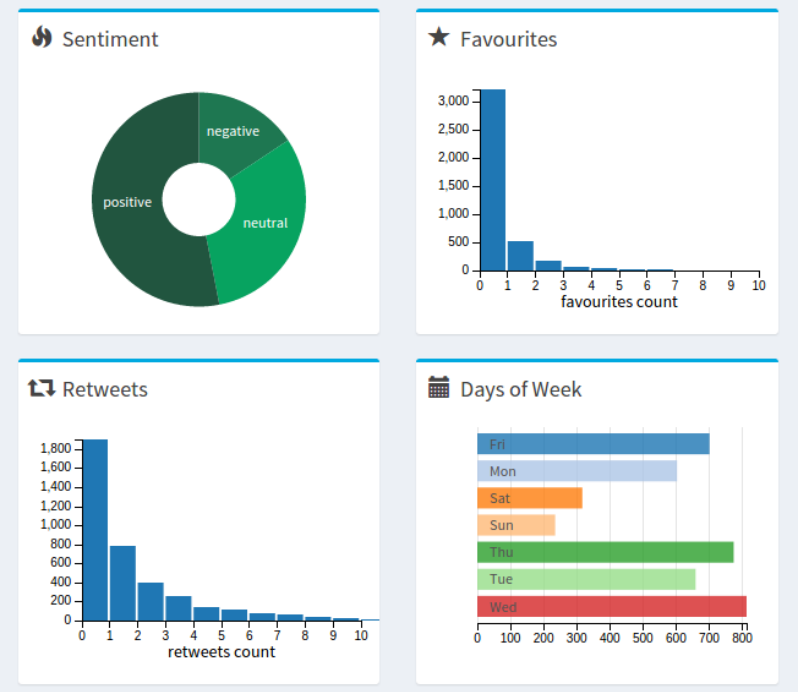

Figure 4.6: Dashboard graphics.

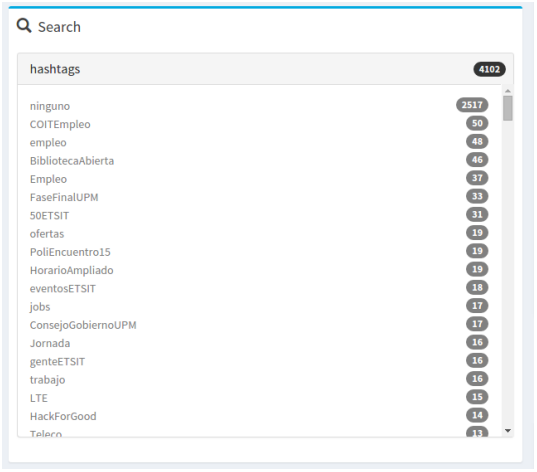

Figure 4.7: Dashboard hashtag list

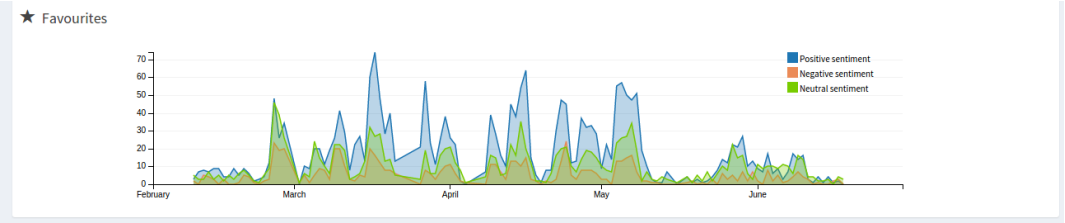

Figure 4.8: Line polarity evolution in time

Figure 4.9: Tag cloud of the three hundred first words that appear more in the whole set of tweets

CHAPTER 5

Conclusions and future work

In this chapter we will describe the conclusions extracted from this project, and the thoughts about future work.

5.1 Conclusions

In this project we have studied two different approaches to the social network *Twitter*: Mining Twitter and Sentiment Analysis in Twitter.

We have seen the different strategies used for extracting information from Twitter, and using the Twitter API and a database in MongoDB, we have developed a collector capable of obtaining information from a certain topic given a list of keywords and users.

We have performed a sentiment analysis system. During its development, several experiments have been done with a large list of features. This has allowed us to use the most relevant features and improve our system. The system has needed a preprocessor for text messages due to improve the features extracted from them.

We have participated in both tasks of the workshop TASS 2015. Moreover, we have submitted three experiments based on this system and the features that adjust better with

the TASS corpus, obtaining good results.

Nonetheless, we have experimented the difficulties of having a small training set due to the lack of features that this training set can provide to the classifier. Although, we have seen that the best feature adapted to this kind of corpus is the dictionary lexicons.

The development of the dashboard has showed us all the information that could be extracted from a tweet message, and how to analyse it.

5.2 Future work

There are several lines that can be followed to extend some of the features used on the sentiment analysis but were not included into this project due to the time limitation. In the next points some lines of work or improvement to continue the development are presented.

- During the preprocessing of a message, a corrector algorithm could be very useful avoiding the misspelling that could appear on it. This will help the classifier to distinguish the words or lemmas that are equal and the ones that not.
- Due to the problem with special characters (accents, 'ñ', etc.) and pronunciation in Spanish language, a metaphone, which is a phonetic algorithm for indexing words by their pronunciation, could be really useful[11].
- Test the possibilities of using other lexicons to extract features for the classifier.
- Perform a different approach with the negation to improve the results and try to avoid the negation errors occurred in the results.
- Detect augmentatives and diminutives when the text is analysed use them to attempt to classify the text or give any value to their polarity.
- Build a manually lexicon using the approach described by Saif[10].
- Develop a classifier based on a more complex model of emotions.
- Release the corpus extracted from Twitter following an approach LLD (NIF + Marl).
- Applying techniques of social network analysis to identify opinion makers in Telecom.
- Combine other external sources outside twitter (slashdat, facebook, etc.) and build a larger corpus.

Bibliography

[1] Andrei Broder. A taxonomy of web search. In *ACM Sigir forum*, volume 36, pages 3–10. ACM, 2002.

[2] Zhiyuan Cheng, James Caverlee, and Kyumin Lee. You are where you tweet: a content-based approach to geo-locating twitter users. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 759–768. ACM, 2010.

[3] Fermín L Cruz, José A Troyano, Beatriz Pontes, and F Javier Ortega. Building layered, multilingual sentiment lexicons at synset and lemma levels. *Expert Systems with Applications*, 41(13):5984–5994, 2014.

[4] Alexander Hogenboom, Daniella Bal, Flavius Frasincar, Malissa Bal, Franciska De Jong, and Uzay Kaymak. Exploiting emoticons in polarity classification of text.

[5] Lluıs-F Hurtado and Ferran Pla. Elirf-upv en tass 2014: Análisis de sentimientos, detección de tópicos y análisis de sentimientos de aspectos en twitter.

[6] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM, 2007.

[7] Alistair Kennedy and Diana Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125, 2006.

[8] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[9] E Martınez-Cámara, MT Martın-Valdivia, MD Molina-González, and LA Urena-López. Bilingual experiments on an opinion comparable corpus. *WASSA 2013*, 87, 2013.

[10] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan 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, pages 321–327, 2013.

[11] Alejandro Mosquera, Elena Lloret, and Paloma Moreda. Towards facilitating the accessibility of web 2.0 texts through text normalisation. In *Proceedings of the LREC workshop: Natural Language Processing for Improving Textual Accessibility (NLP4ITA) ; Istanbul, Turkey.*, pages 9–14, 2012.

[12] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[13] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[14] José M Perea-Ortega and Alexandra Balahur. Experiments on feature replacements for polarity classification of spanish tweets.

[15] Inaki San Vicente Roncal, Elhuyar Fundazioa, and Xabier Saralegi Urizar. Looking for features for supervised tweet polarity classification.

[16] Alejandro Saura-Villanueva. Development of a framework for GeoLinked Data query and visualization based on web components. Proyecto fin de carr, Universidad Politécnica de Madrid, Avda. Complutense, 30, June 2015.

[17] Roser Saurí and James Pustejovsky. Are you sure that this happened? assessing the factuality degree of events in text. *Computational Linguistics*, 38(2):261–299, 2012.

[18] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[19] Xabier Saralegi Urizar, Elhuyar Fundazioa, and Inaki San Vicente Roncal. Elhuyar at tass 2013.

[20] Rada Mihalcea Veronica Perez Rosas, Carmen Banea. Learning sentiment lexicons in spanish. In *Proceedings of the international conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey, 2012.

[21] David Vilares, Yerai Doval, Miguel A Alonso, and Carlos Gómez-Rodrıguez. Lys at tass 2014: A prototype for extracting and analysing aspects from spanish tweets.

[22] Joseph B Walther and Kyle P D'Addario. The impacts of emoticons on message interpretation in computer-mediated communication. *Social science computer review*, 19(3):324–347, 2001.

[23] Salud Marıa Jiménez Zafra, Eugenio Martınez Cámara, M Teresa Martın Valdivia, and L Alfonso Urena López. Sinai-esma: An unsupervised approach for sentiment analysis in twitter.