UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

GRADO EN INGENIERÍA DE TECNOLOGÍAS Y SERVICIOS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DEVELOPMENT OF A REAL TIME CLASSIFICATION SYSTEM OF TWITTER TRENDS BASED ON MACHINE LEARNING TECHNIQUES

DANIEL DE LA MATA NIEVES

2018

TRABAJO FIN DE GRADO

Título:	Desarrollo de un Clasificador a Tiempo Real de Tendencias de Twitter basado en Técnicas de Machine Learning
Título (inglés):	Development of a Real Time Classification System of Twitter Trends based on Machine Learning Techniques.
Autor:	Daniel de la Mata Nieves
Tutor:	Carlos A. Iglesias Fernández
Departamento:	Ingeniería de Sistemas Telemáticos

MIEMBROS DEL TRIBUNAL CALIFICADOR

Presidente:

Vocal:

Secretario:

Suplente:

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 FIN DE GRADO

DEVELOPMENT OF A REAL TIME CLASSIFICATION SYSTEM OF TWITTER TRENDS BASED ON MACHINE LEARNING TECHNIQUES

Daniel de la Mata Nieves

Enero de 2018

Resumen

En los últimos años las redes sociales han experimentado un crecimiento exponencial, actualmente nos encontramos en la era de la información, la cual es vital analizar minuciosamente para poder extraer y explotar todo su potencial. Dentro de las redes sociales más utilizadas, nos encontramos con Twitter, se trata de un servicio de microblogging en el que se envían cada día más de 500 millones de mensajes diarios, denominados tweets, con el fin de compartir nuestros intereses comunes acerca de un tema. Aquellos tweets que son más relevantes en un momento concreto y se convierten en tendencia, se denominan trending topics.

El objetivo ha sido investigar el trasfondo de estas tendencias sociales con el fin de analizar y entender la causa por la cual se producen. Para ello, se ha desarrollado un clasificador automático permitiendo identificar la categoría de la tendencia e informar al usuario a tiempo real para poder obtener el máximo beneficio posible de ello.

En primer lugar se ha monitorizado la API de Twitter durante una semana, del 1 al 7 de Diciembre de 2017, obteniendo el top 10 de trending topics de España cada 30 segundos a través de Tweepy. Una vez obtenidos los trending topics se han descargado los tweets más recientes asociados a cada uno de ellos. A continuación, se han categorizado los tweets según dos tipologías. La primera de ellas se ha dividido en cuatro categorías: noticias, eventos en vivo, días conmemorativos y memes. Mientras que la segunda la división ha sido realizado en seis categorías: deportes, negocios, entretenimiento, salud, política y tecnología. Para ello se ha realizado una anotación manual de los trending topics obteniendo un Cohen Kappa ratio de 0.78 y 0.89 respectivamente que mide el grado de acuerdo entre anotadores.

Después, se han organizado los tweets extraídos para su posterior pre procesado y extracción de características para alimentar al clasificador. Más tarde, se han desarrollado clasificadores que buscan ciertas estructuras o patrones en los datos e implementan modelos predictivos que permitan una optimización del mismo de manera automática. Por último, se ha desarrollado distintos clasificadores y realizado una comparación de prestaciones entre los mismos. Para ello se hecho uso de diversos algoritmos obteniendo los mejores resultados con Support Vector Machines (SVC), ExtraTreesClassifier (ETC) y Multinomial Naive Bayes (MNB).

Respecto a la primera categorización se ha obtenido como mejor resultado a través de cross validation del 0,91 con ETC. Comparando dicho resultado con los obtenidos por otros autores los cuales han conseguido un mejor rendimiento de 0,81 con SVC [1], resulta una notable mejora del 12,35%.

En la segunda categorización se han obtenido como mejor resultado 0,92 con SVC, MNB y ETC mientras que el mejor resultado obtenido por otros autores ha sido del 0,78 [12] suponiendo una mejora del 17,95%.

Palabras clave: Aprendizaje automátio, Twitter, Trending topics, Phyton, Scikit-learn, Clasificador.

Abstract

In recent years social networks have experienced an exponential growth, we are currently in the information age, which is vital to carefully analyze to extract and exploit its full potential. Within the most used social networks, we find Twitter that is a micro-blogging service where more than 500 million messages are sent every day, called tweets, in order to share our common interests about a topic. Those tweets that are more relevant at a specific moment and become a trend, are called trending topics.

The objective will be to investigate the background of these social trends to analyze and understand the cause why they occur. To achieve this, an automatic classifier will be developed to identify the trend category and inform the user in real time, in order to obtain the maximum possible benefit from it.

Firstly, the Twitter API has been monitored for a week through Tweepy, from December 1 to 7, 2017, obtaining the top 10 trending topics in Spain every 30 seconds. Once the trending topics have been obtained, the most recent tweets associated with each of them have been downloaded. Later the tweets have been categorized according to two typologies. The first one has been divided into four categories: news, live events, commemorative days and memes while the second division has been made in six categories: sports, business, entertainment, health, politics and technology. Then a manual annotation of the trending topics was done, obtaining a Cohen Kappa ratio of 0.78 and 0.89 respectively, which measures the degree of agreement.

Afterwards, the extracted tweets have been organized for their subsequent preprocessing and extraction of characteristics to feed the classifier. Next, classifiers have been developed that look for certain structures or patterns in the data and implement predictive models that allow its automatically optimization. Finally, different classifiers have been developed and a performance comparison has been made between them. Several algorithms were used, obtaining the best results with Support Vector Machines (SVC), ExtraTreesClassifier (ETC) and Multinomial Naive Bayes (MNB).

Regarding the first categorization, the best result of 0.91 has been obtained through cross validation with ETC. Comparing this result with those obtained by other authors

who have achieved a better performance of 0.81 with SVC [1], a remarkable improvement of 12.35% has been achieved.

In the second categorization, the best result has been 0.92 with SVC, MNB and ETC, while the best result achieved by other authors was 0.78 [12], assuming an improvement of 17.95%.

Keywords: Machine Learning, Twitter, Trending topics, Phyton, Scikit-learn, Classifier

Agradecimientos

Me gustaría dar las gracias a todas aquellas personas que me han apoyado incondicionalmente.

A mi familia y amigos los cuales son los pilares de mi vida.

A mi tutor, Carlos Ángles Iglesias y a los compañeros del GSI por guiarme y ayudarme en el desarrollo de este proyecto.

Gracias a todos.

Contents

List of Figures

CHAPTER 1

Introduction

1.1 Context

In the last few years, social networks have suddenly increased in popularity and the way we produce and consume information has changed dramatically. There is a massive amount of data flowing through these social networks. One of the most popular, is Twitter where people share information in form of short text messages of 140 characters, recently updated to 280.

The key to the growth of Twitter is due to its simplicity since its appearance on March 21, 2006 when Jack Dorsey its creator wrote the first tweet. More than 500 millions of messages are shared everyday generating a lot of information that must be analyzed. The objective is to investigate the background of social trends shared in order to analyze and understand the cause why they occur[2].

The fist step to achieve it, must be to organize all the tweets, specifically in this project, they will be categorized in four different taxonomy: news, ongoing-event, commemorative and memes. Each one of this taxonomy will have their own specific characteristics that can not be directly discovered by just reading a single tweet. Detecting this kind of hidden information and put it into context will be very useful for a lot of companies.

1.2 Project goals

Nowadays, the huge amount of data that we have can present several problems to manage it correctly. To solve this problem, we try to show how important it is to treat and organize data properly to obtain useful information for us.

The main objective of the project is to study the use of classification techniques to see the relationship between tweet taxonomy with the rest of their characteristics. If we have a random tweet, we will try to predict which taxonomy it belongs to between some predefined taxonomies. The objective is to obtain a classifier with the best possible score, discovering which characteristics are the most relevant to obtain the best results.

1.3 Methodology

- ***Build our own data set:*** Monitoring the top 10 trending topics every 30 seconds during a week and obtain the most recent associated tweets.

- ***Organize and preprocessing the data:*** We begin with a data set consisting of labeled tweets related to a trending topic. The dataset is sent through a meticulous preprocessing step where the punctuation, hyperlinks and stop words are removed to obtain clean data.

- ***Implementation of the classifier:*** Election and extraction of the different features that will be analyzed by our system. Divide the clean data into train and test sets. Testing with different algorithms and first experiments with the training set. Once the classifier is trained, test data is passed through it and a predicted label for a trend will be generated.

- ***Development of the classifier:*** Using different feature extraction techniques and comparing them to draw conclusions trying to achieve the best performance as possible.

1.4 Structure of this document

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

- ***Chapter 1*** describes the context and the main goals to achieve in this project.

- ***Chapter 2*** provides a description of the main technologies which allows the development of this project.

- ***Chapter 3*** corresponds to the project architecture where the different modules that define our project are explained.

- ***Chapter 4*** presents the different tests evaluated, their results and how to interpret their performance.

- ***Chapter 5*** discuss briefly the conclusions drawn from this project, problems faced, achieved goals and suggestions for a future work.

CHAPTER 2

Enabling Technologies

2.1 Introduction

In this section the main technologies that have contributed to the realization of the project are going to be presented below. In each of them, the relevance they have had in the development of the project will be briefly described.

2.2 Machine Learning

Machine learning is the main technology used in the development of this project. It is a sub field of computer science, a type of artificial intelligence that comes from the evolution of the study of pattern recognition techniques and computational learning theory. It can be classified according to learning style into supervised learning, unsupervised learning and semi-supervised learning.

Supervised learning techniques applied on classification tasks where the machine learning algorithm takes a training data set as an input, made of feature vectors and labels, and produces a predictive model which is used to make prediction on new data will be the focus of the project [3].

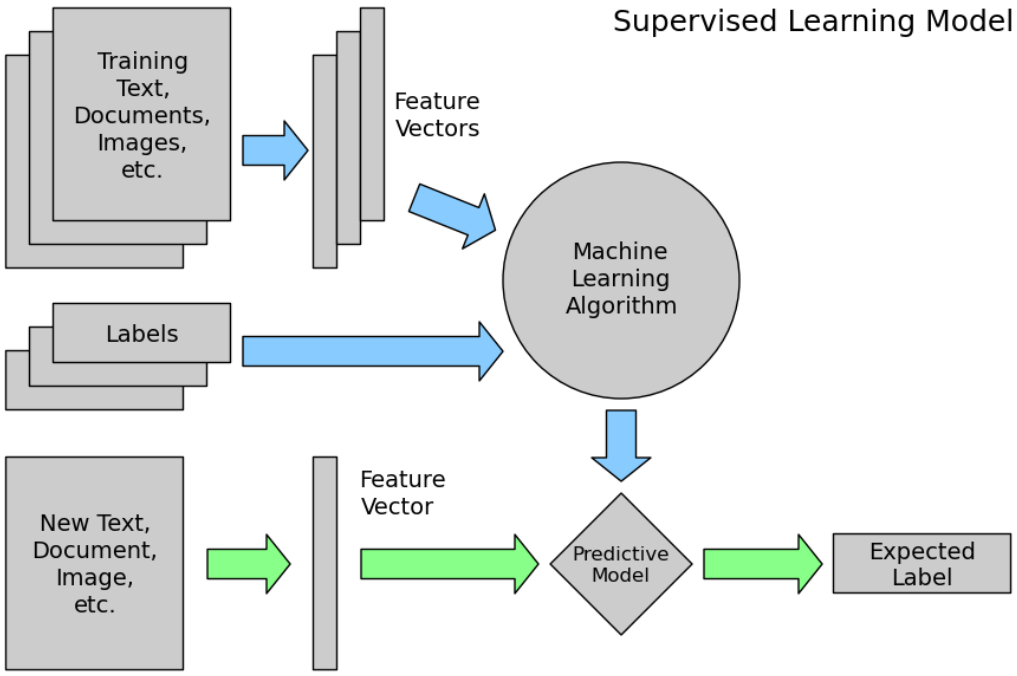

Figure 2.1: ML Architecture [3]

2.3 Scikit-learn

Scikit-learn is an open source machine learning library for Python which provides a range of supervised and unsupervised machine learning tools for data mining and data analysis built on NumPy, SciPy, matplotlib and pandas. Scikit- learn provides algorithms for solving problems like classification, clustering, regression and dimensional reduction. Classification ones (e.g., kNN, SVM, Random Forest) based on identifying to which category an object belongs to, will be the focus of the project [4].

- ***NumPy:*** provides basic routines for manipulating large arrays and matrices of numeric data [**?**].
- ***SciPy:*** extends the functionality of NumPy with a substantial collection of useful algorithms, like minimization, Fourier transformation, regression, and other applied mathematical techniques [**?**].
- ***Matplotlib:*** comprehensive 2D Plotting library that produces publication quality figures in a variety of formats and interactive environments across platforms [7].
- ***Pandas:*** provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive[6].

2.4 NLTK

The Natural Language Toolkit (NLTK) is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning [8].

2.5 Twitter API

The Twitter API is a tool that provides programmatic access to read and write Twitter data. It is based on a REST architecture which allows the system to access and manipulate textual representations of Twitter web resources with a predefined set of stateless operations.

In this project Twitter API is used to monitor the trending topics of Twitter during a given time through the WOEID which is the identifier of a specific place, in our case, Spain. In addition, once the trending topics are downloaded, a request is made to extract the most recent tweets associated.

It is necessary to emphasize that you need to register as a developer in Twitter to get the credentials to have API access and it is very important to consider the several limitations of this tool as it is only possible to make a limited number of petitions every 15 minutes. This can be a crucial factor if the information required is big [9].

2.6 Tweepy

The developer community creates a lot of libraries which extend the language and make it easier the use of several services. One of those libraries that has been key in the development of this project is called Tweepy. It is an open-sourced library that allows Phyton to communicate with the Twitter platform and make use of its API. On their own website, they describe themselves as: "An easy-to-use Python library for accessing the Twitter API" [10].

CHAPTER 3

Architecture

3.1 Introduction

In this section, the general architecture of the project will be explained including the design phase and implementation details. We will start providing a general view of the system in order to understand how it works without going into detail. We need to identify the project subsystems and the processes followed in its development. Afterwards, every module will be explained in detail so we will have a better and deeper overview of the system.

3.2 Overview

In this section, the general architecture of the project will be explained giving a brief description of the modules that are involved in the system functionality. It is important to acquired a global vision showing how the different modules are interrelated.

As we have explained previously, the final goal of the project is to be able to predict the trending topic typology of a tweet between four predefined groups: news, ongoing events, memes and commemorative.

Previous work done in classification of Twitter trends have been the basis for the selection of the categories. Observing the trending topics for a long period we realized that the typology chosen with meme, ongoing-event, commemorative and news fits quite well to reality, it is a cross categorization that covers a large percentage of current trends in Twitter. Despite this typology covers most of the triggers on Twitter, it is true that some trends could go in several categories. Although a multi-label classification could be done, in this project we have chosen to associate the best-fitting label to each trend.

Investigating Api we discovered that there is a method where you can obtain the trending topics of a certain place through the WOEID (Where On Earth IDentifier) which is a 32-bit reference identifier that identifies any place on Earth. We chose Spain as a place and we monitored the top 10th trending topics every 30 seconds during a week, introducing them in a list in order to after labeling them. Each time a new trending topic appears, we introduce it into the list only if it is not already there. Once the list is completed, we download the most recent tweets associated with those trending topics obtaining our data set. The next step is categorizing each trending topic of the list obtained in one of the following groups:

- ***News:*** one of the greatest opportunities that twitter provides is their information spread. Several times a normal user on Twitter notices and posts that something is happening before the news organizations. This gives rise to the possibility of news breaking on Twitter before the traditional media can get to it. This situation automatically makes this category of trending topic one of the most common and attractive among others.

- ***Ongoing events:*** another type of trending topic we categorized was produced by a community of users tweeting about an ongoing event as it unfolds. The practice of live-tweeting an event as it is taking place has become fundamental as Twitter has gained importance as a real-time information sharing media. This kind of trending topics are discussions going around live events like concerts, sports, elections, etc. Normally, these topics appear when a big event is going on or something out of the ordinary happens during a popular event.

- ***Memes:*** there is a portion of trending topics which are kind of catchy taglines or viral ideas in order to resonate with the users to support a movement, product, presidential candidate, celebrity, etc.

- ***Commemorative:*** these topics are understandably the least frequent since they are commemorating a certain event or person that is being remembered in a given day. We consider a trending topic as triggered by a commemorative when users are

congratulating a celebrity for their birthday, celebrate the anniversary of a certain event or person or it is just a memorial day.

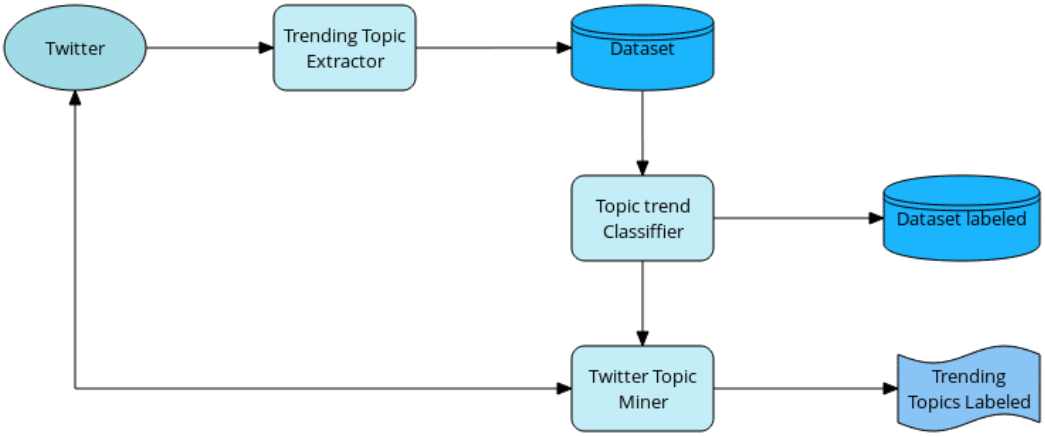

Figure 3.1: Architecture

3.3 Data set

For the realization of this project it has been decided to build our own data set which allow us to assign the characteristics that we consider most appropriate to be able to optimize the classifier. In related work, the proportion between the trending topic categories was considerably disproportionate. The commemorative trending topics did not exceed 3% of the total trending topics and ongoing events are close to 60%. To solve this problem, it has been decided to establish specific dates in order to increase this type of tweets by matching the dates of extraction of the data set with significant days.The data sets also presented the tweets in different languages, in our case, it has been decided to focus on tweets written only in Spanish.

3.3.1 Trending topics monitoring

The construction of the data set can be divided into several processes that complement each other. The first of them consists of, as previously explained, making a query through the WOEID, for this we have chosen the identifier associated with Spain and we have monitored

the top 10 of trending topics of Twitter during a week from December 1st to 7th, 2017. The monitoring consisted of making a query through the Twitter API every 30 seconds and get the list of the top 10th trends in our country. A list has been drawn up where the new trending topics were introduced at the time they were detected. Following the above gathering process, we collected a total of 855 unique trending topic with the following distribution (150, 121, 114, 119, 123, 120, 108).The first day analyzed presents the highest number of trending topics because if a trend lasts more than a day, as it is not a new trend, it does not appear in the rest of the days.

3.3.2 Trending topics annotation

In order to perform the classification experiments, and to analyze the characteristics of the trending topics, all them were manually annotated according to its type within the typology we previously defined: memes, news, ongoing events, or commemorative. The manual annotation process consists in assigning to each trending topic a unique category that fits better among the four available categories. To achieve this, the tweets associated with each trending topic have been meticulously read for a more adequate categorization and the following results have been obtained with the corresponding distribution: 457 ongoing events (54%), 172 news (20%), 165 memes (19%) and 60 commemorative (7%).

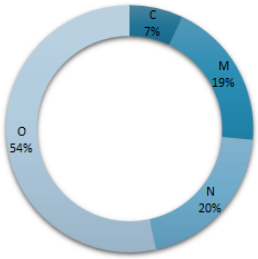

Figure 3.2: Trending topics distribution (I)

3.3.3 Cohen Kappa

In order to measure the inter-annotator agreement we computed the Cohen Kappa coefficient. It is a score that expresses the level of agreement between two annotators on a classification problem.

It is defined as $k = \frac{p_0 - p_e}{1 - p_e}$ where p_o is the empirical probability of agreement on the label assigned to any sample, and $p_e$is the expected agreement when both annotators assign

labels randomly. p_e is estimated using a per-annotator empirical prior over the class labels.

We have obtained a 0.778 ratio which is considered as substantial result.

3.3.4 Tweets download

The second process is based on obtaining the most recent tweets related to each trending topic making a query through the Twitter API and getting only those tweets that were written in Spanish.

There were some trending topics like "Navidad" (Christmas) where the amount of recent associated tweets was huge. The size of the Data set was enormous, to avoid that, it has been decided to limit the number of tweets per trending topic to 1000. With this solution, our data is more homogeneous and there are no substantial differences.

Below is the daily distribution of the monitored trending topics where it is observed that the ongoing events clearly predominate over the rest, especially in those days of big sport events.

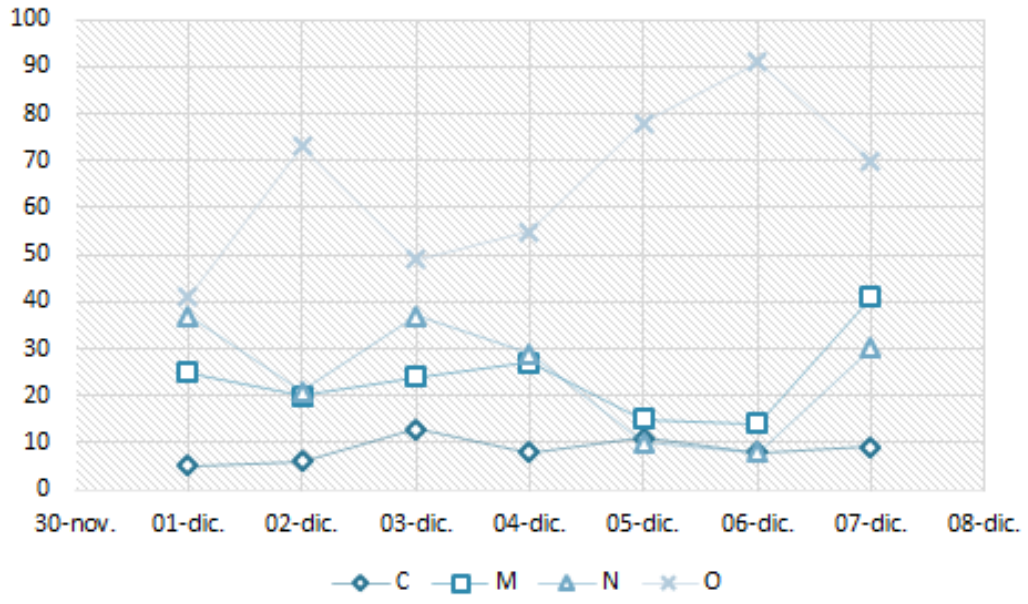

Figure 3.3: Daily distribution (I)

3.4 Dataset extraction

One of the essential features of Twitter consist in dealing with natural language so it is very important to find out the keywords in the tweets among the noise: stop words, punctuation, hyperlinks, emoticons, numbers, urban slang or twitter specific vocabulary like RT or '#'.

Therefore, once we have the raw text we will start the preprocessing. The first step consists in cleaning the text, eliminating undesirable elements. We are going to eliminate the emojis, smileys, mentions and reserved specific words of twitter like RT or FAV. It has been decided not to eliminate hashtags as usual preprocessors do, since in our project, they can provide fundamental information.

Also there are a lot of tweets that include hyperlinks that the user wishes to share. Implementing a hyperlink filter is critical due to we will avoid a lot of noise. In addition, punctuation has also been removed to keep only words.

We select all the tweets, we divide them into words and sent through a rigorous multi-step filtering process. There is a need of preprocessing texts to avoid those words that can introduce noise in the classification, they are called stop words. Our data set is in Spanish, so the Stop Words Filter provided by NLTK python library must take it into account.

After that, lemmatization will be applied that consists on the extraction of the root of the words. This is very useful due to the concept of the word gives more information than all of their variations and we avoid more noise. The next step is measuring the frequency that each word appears into that vector space which is called "Term Frequency".

Term Frequency is an extremely powerful method to rank features which consists in evaluating how important a word is in a text. We convert the text into a Vector Space Model ignoring the order of words but focusing on how many times a words occurs.

3.5 Features extraction

Once we have done the preprocessing of our data and we have separated the text by words, it is time to obtain useful information to be able to analyze the text. To achieve this, we will convert the text of each tweet into vectors from which we can extract features to analyze each tweet in detail. We are going to focus on the characteristics of the text only since the objective is to be able to predict the taxonomy of the tweet based on the text itself.

3.5.1 Text features

In this module, we are going to explain which features of the tweets are important in the election of the taxonomy of the text. The different text features chosen that can be directly extracted from the raw text of the tweets are list below.

- ***Capital letters:*** the amount of capital letters used in the tweets, the number of capital letters is compared with the full length of the tweet.
- ***Elongated words:*** when a word has a character repeated for 3 times or more it has been considered as an elongated word.
- ***Exclamations:*** number of tweets with exclamation signs. Tweets with at least an exclamation are given a value of 1, whereas tweets without exclamations are given a value of 0. This feature computes the average of those values.
- ***Interrogations:*** number of question signs in tweets. It is the same as with exclamations, but with question signs. It is necessary to emphasize that our data set is in Spanish so the signs '¡' and '¿' have to be taken into account since they do not exist in other languages and they usually appear in tweets as news or commemorative days.
- ***Punctuation marks:*** the use of punctuation marks like commas or full stop. The grammar can be more neglected in some taxonomy groups like memes or more take it into account for example in news. So the number of punctuation marks is a feature to have in count.
- ***Expressions:*** despite Tweets are in Spanish, there are some expressions very used in social networks, like "LOL" or "OMG" for example, which are associated normally with memes. The use of these expressions is quantified in this feature.
- ***Emoticons:*** the use of emoticons and smileys is a very important feature to take into account in taxonomy classification as it may be representative of memes groups.
- ***Mentions, URLs, hashtags and retweets:*** with this features we measure how the users interact with other users, which may be different between taxonomy groups. We keep track of this by observing the appearance of '#', hyperlinks, '@', or 'RT' symbols.
- ***Average text and word lengths:*** the study of the text length and the contained words can give us valuable information for classification. This feature measures the average length of this parameters.

- ***Topic repetition:*** average number of uses of the trending topic in tweets. The term corresponding to the trending topic may appear more than once in a tweet, especially when users repeat it to make it stronger and trend.

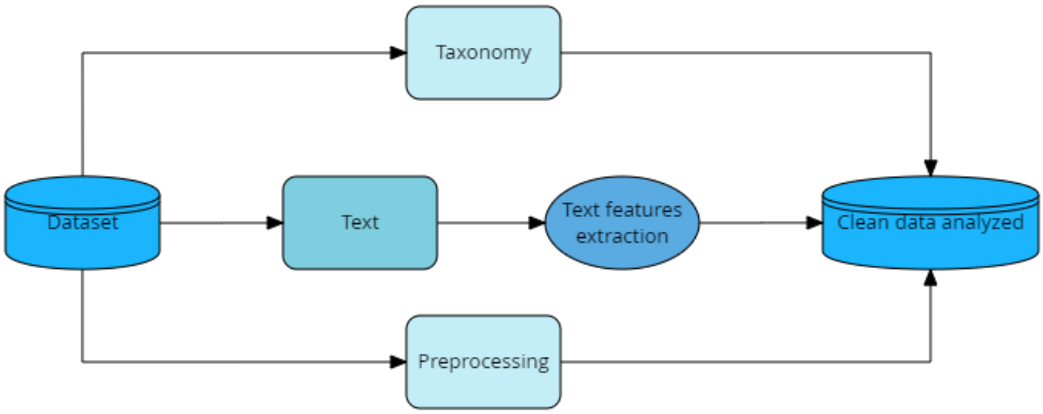

Figure 3.4: Text features

3.6 Pipeline

The aim is that the classifier is able to read the data we have extracted. To achieve this, a pipeline has been created with a feature union that allows parallel transformations to the input data and concatenates the results at the end. In addition, an algorithm that is used to discover the topics that are present in a corpus has been added to the pipeline. This algorithm is called Latent Dirichlet Allocation (LDA), which consists that each document can be seen as a set of topics and associates the appearance of certain words with a specific topic.

3.7 Classification Model

Now we have our dataset analyzed with the selected features which can be read by the classifier. A classification model will be implemented to interpret these features and predict the taxonomy of the tweet. To achieve this, we have been testing our dataset with different types of classifiers already designed by the Scikit-learn library, in order to evaluate which one best fits our corpus and obtain better results. All of them are going to be briefly described in the list below[4].

- ***AdaBoost classifier:*** is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.
- ***DecisionTrees:*** are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features[4].
- ***ExtraTreeClassiffier:*** extra-trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each of the maxfeatures randomly selected features and the best split among those is chosen. When maxfeatures is set 1, this amounts to building a totally random decision tree.
- ***Random forest classifier:*** is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True.
- ***Support Vector Classification:*** the implementation is based on libsvm. The fit time complexity is more than quadratic with the number of samples which makes it hard to scale to dataset with more than a couple of 10000 samples. The multiclass support is handled according to a one-vs-one scheme.
- ***Linear Support Vector Classification:*** similar to SVC with parameter kernel='linear', but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples. This class supports both dense and sparse input and the multiclass support is handled according to a one-vs-the-rest scheme.

- ***MultinomialNB:*** the multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.

- ***K-Nearest Neighbors classifier (K-NN):*** neighbors-based classification is a type of instance-based learning or non-generalizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point.

3.7.1 Steps

Once we have presented all the classifiers that are going to implemented, we must test the accuracy of the algorithm. The different steps to achieve it will be explained in detail in the list below.

- ***Splitting the features into training set and testing set:*** To evaluate the algorithms it is necessary to divide our data into two groups. One of them is going to be used for training the algorithm, as the learning source. The test group it is used to test the performance of the system. Stratified K-Folds cross validation using ten folds is implemented for doing this splitting in an efficient way. This way we create stratified folds preserving the percentage of samples of each class.

- ***Training the classifier:*** Once we have all the features splitted, the next step consists in feeding the system with the training set. With this step we allow the algorithm to learn building relationships between the features. After the training process is finished the system will be prepared to start with their first predictions from new input data.

- ***Testing the classifier:*** The remaining part of the data set is used as input of the classifier. The data is going to be classified into a taxonomy group according to the previous learning process step where the test features are analyzed.

- ***Results:*** Once the test is done, the predictions of the classifier will be compared with the labeled data obtaining a percentage of success that will show the classifier performance.

3.7.2 Classification Metrics

Once we have finished our classifier which is capable of making predictions, it is time to measure if it is good enough to solve our problem. To be able to check the performance of our classifier, we will start by obtaining the confusion matrix that is a graphical, clean and unambiguous way to present the prediction results of a classifier. For a more simple explanation of its operation we will explain it with a binary classification. In this case, the main diagonal represents the number of elements where the predicted label matches the true label while the off-diagonal elements correspond to those values mislabeled by the classifier. Therefore, the greater number of elements in the main diagonal will show a better classifier performance. According to the collected values we can obtain [11]:

- ***True Positives (TP):*** these are the correctly predicted positive values which means that the value of labeled and predicted class are both positive.
- ***True Negatives (TN):*** these are the correctly predicted positive values which means that the value of labeled and predicted class are both negative.
- ***False Positives (FP):*** when labeled class is negative and predicted class is positive.
- ***False Negatives (FN):*** when labeled class is positive and predicted class is negative.

  Once we have explained these four parameters then we can calculate Accuracy, Precision, Recall and F1 score.

- ***Accuracy:*** is the most intuitive and is the simply ratio of correct predictions among the total number of predictions. $\frac{TP+TN}{TP+FP+FN+TN}$
- ***Precision (P):*** is the ratio of correctly predicted positive observations among the total predicted positive observations. $\frac{TP}{TP+FP}$
- ***Recall (R):*** is the ratio of correctly predicted positive observations among the total observations in labeled class. $\frac{TP}{TP+FN}$
- ***F1 score:*** is the weighted average of Recall and Precision. Therefore, this score takes both false positives and false negatives into account. $2\frac{RP}{R+P}$

CHAPTER 4

Evaluation

4.1 Introduction

In order to evaluate the performance of the classifier, a series of tests have been carried out which must be interpreted thoroughly for a correct analysis. In this section, the objective is to describe those tests and give a complete analysis of the results obtained to reach a conclusion on the performance of the classifier. The differences between the tests performed depend on the following characteristics:

- ***Data sets:*** the size of the data set has been modified in order to check if a higher system performance was directly proportional to a larger data set.
- ***Algorithms:*** a series of classification algorithms have been selected, obtaining different results due to each algorithm has their own specific characteristics and we must look for the one that best fits our system.
- ***Taxonomy:*** a new way to categorize the tweets has been implemented, a change from four different categories to six to check how the system performs under different conditions. The trending topics have been labeled again with the new categories which are: business, health, entertainment, sports, technology and politics.

4.2 Data sets

The different tests have been carried out with two data sets of different sizes to check the behavior of the system. In the first one, as explained above, the number of tweets associated for each trending topic was reduced to 1000. In the case of the second data set, it has been further reduced to 100 tweets per trending topic.

4.3 Taxonomy

Another type of categorization has also been established where the new trending topics have been categorized into business, health, entertainment, sports, technology and politics. For this, it has been necessary to redo the manual annotation of the trending topics obtaining the following distribution: 26 business (3.04%), 58 health (6.79%), 263 entertainment (30,80%), 270 sports (31.62%), 28 technology (3.28%) and 209 politics (24.47%).

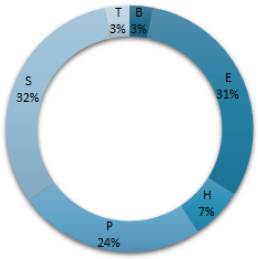

Figure 4.1: Trending topics distribution (II)

4.4 Cohen Kappa

For this new way of categorizing, the Cohen Kappa ratio has also been calculated, obtaining a value of 0.894 which is considered almost perfect agreement. Below is a table summarizing the Cohen Kappa ratios obtained. It is clearly observed that a considerably higher ratio has been obtained which indicates a greater reliability when training the classifier.

The daily distribution of the new categories of trending topics is shown where one can observe that trending topics of entertainment usually predominate except those days when sports increase due to weekend or events such as the Champions League.

Table 4.1: Kohen Kappa Ratio

Dataset	Reference	Results
1	0.597	0,778
2	-	0,894

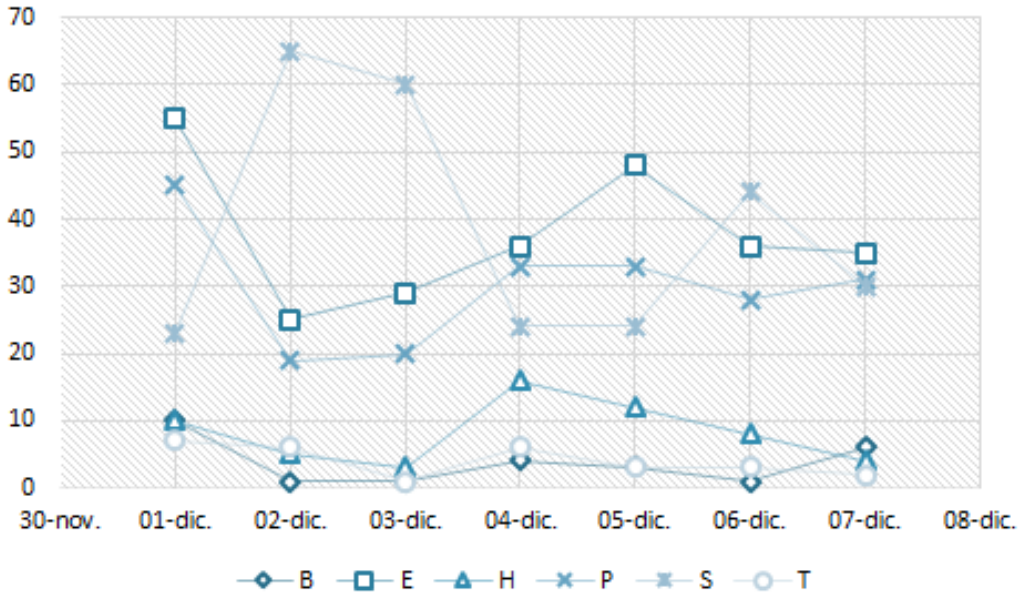

Figure 4.2: Daily distribution (II)

4.5 Algorithms

To implement the classifier a series of different algorithms supplied by sklearn have been used obtaining the following results listed below.

It has been differentiated when performing the tests between the way to categorize being *1* the usual way with the four categories already mentioned and with a *2* the new implemented with six different categories. The size of the data set is also differentiated.

Table 4.2: Algorithms classification (I)

	1-1000	**1-100**	**2-1000**	**2-100**
ADB	0,58	0,57	0,57	0,56
DT	0,49	0,48	0,52	0,43
KNN	0,50	0,48	0,50	0,48
RF	0,47	0,46	0,41	0,39
SVC L	0,50	0,48	0,50	0,48
SVC	0,90	0,87	**0,92**	0,85
ETC	**0,91**	**0,89**	**0,92**	**0,90**
MNB	0,90	0,87	**0,92**	0,89

There is a large performance difference between algorithms corresponding to *ExtraTreeClassiffier (ETC), Support Vector Classification (SVC) and MultinominalNB (MNB)* the best results.

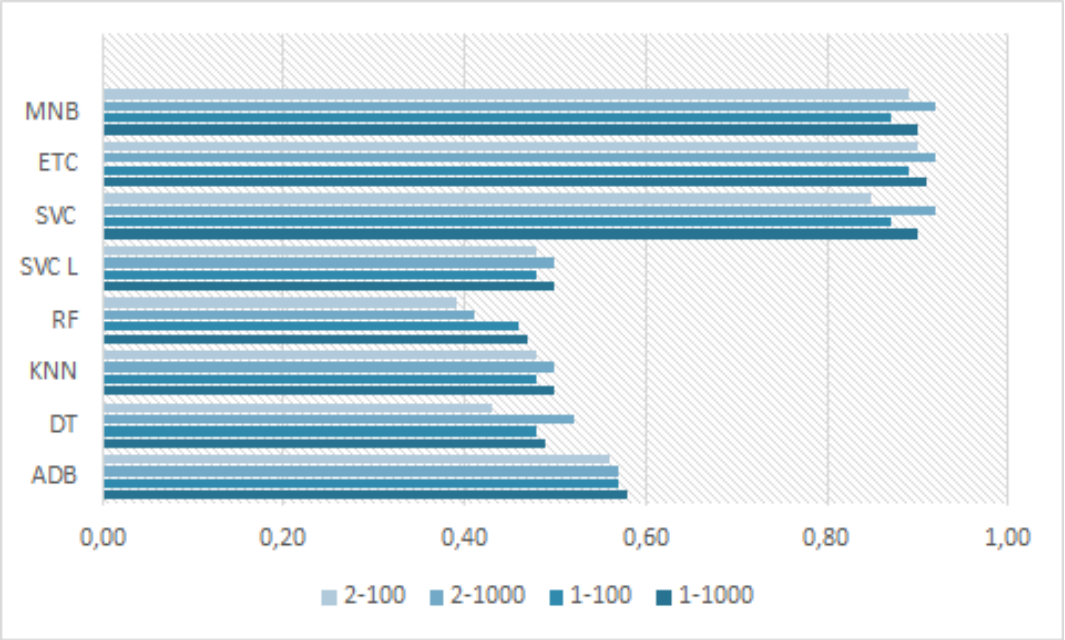

Figure 4.3: Algorithms classification (II)

4.6 Confusion matrix

It has been decided to analyze more in detail the three cases in which greater performance has been obtained by calculating the confusion matrices. To achieve this, each category has been associated with a number. Commemorative with 0, memes 1, news 2, ongoing-event 3, business 4, entertainment 5, health 6, politics 7, sports 8 and technology 9.

4.6.1 Dataset 1

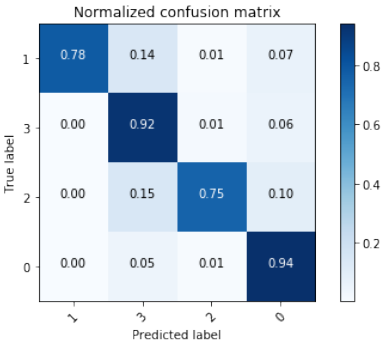

Figure 4.4: SVC1-100

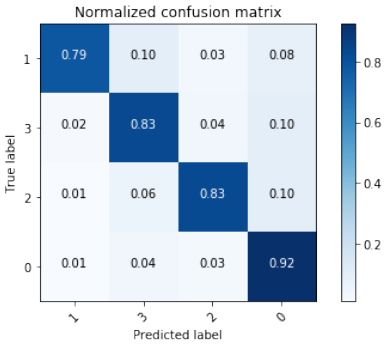

Figure 4.5: MNB1-100

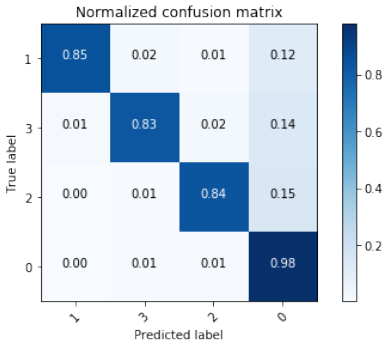

Figure 4.6: ETC1-100

It can be observed that the one that presents a better performance is ETC but it is necessary to emphasize that for ongoing events SVC it obtains a ratio of 0.92 compared to 0.83 of ETC. SVC has a tendency to get confused associating with ongoing events while MNB and ETC with commemorative.

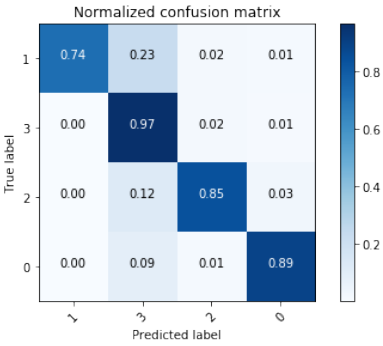

Figure 4.7: SVC1-1000

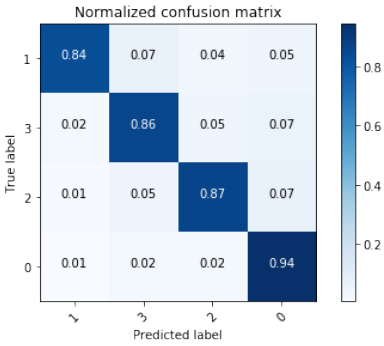

Figure 4.8: MNB1-1000

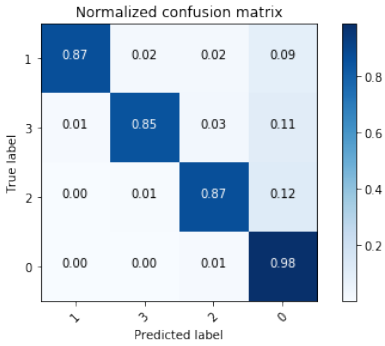

Figure 4.9: ETC1-1000

It can be seen that for this case both ETC and MNB present a similar performance with a tendency to be confused with category commemorative. On the other hand, SVC has acquired a strong inclination for ongoing events.

4.6.2 Dataset 2

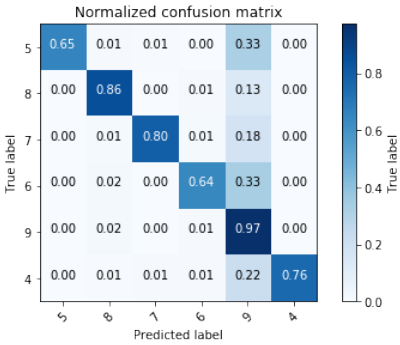

Figure 4.10: SVC2-100

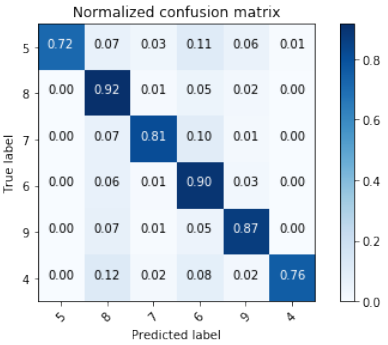

Figure 4.11: MNB2-100

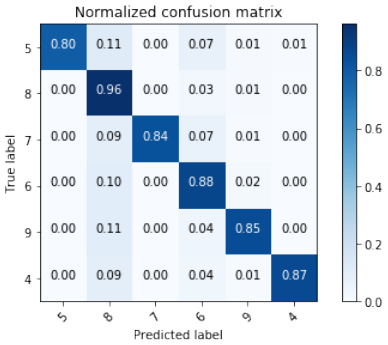

Figure 4.12: ETC2-100

It can be observed that for this case the one that also presents a better performance is ETC. There is a clear tendency in SVC to associate everything to technology, especially when they are entertainment and health while MNB and ETC decline more towards sports.

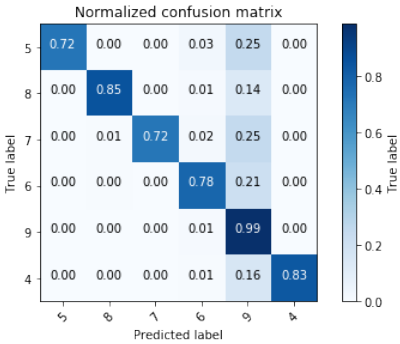

Figure 4.13: SVC2-1000

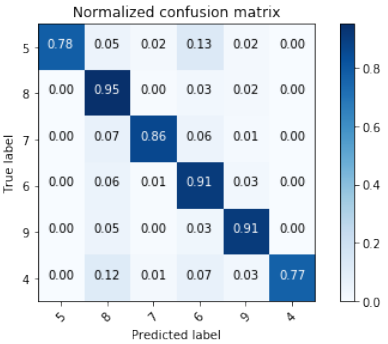

Figure 4.14: MNB2-1000

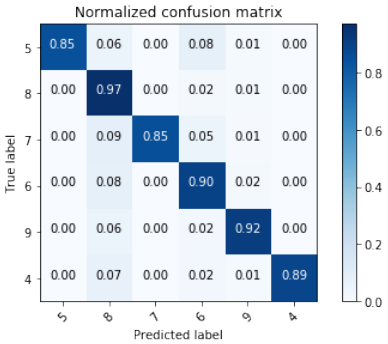

Figure 4.15: ETC2-1000

It can be seen that for these conditions ETC obtains the best performance. ETC and MNB have a tendency to be confused with entertainment and sports. On the other hand, SVC has acquired again a strong inclination for technology.

Table 4.3: Performance SVC

SVC	**1-1000**	**1-100**	**2-1000**	**2-100**
Accuracy	0,88	0,88	0,86	0,81
Recall	0,89	0,89	0,94	0,91
Precision	0,86	0,85	0,82	0,78
F1 score	0,86	0,86	0,86	0,82

Table 4.4: Performance MNB

MNB	**1-1000**	**1-100**	**2-1000**	**2-100**
Accuracy	0,90	0,87	0,92	0,88
Recall	0,88	0,86	0,90	0,90
Precision	0,88	0,84	0,96	0,82
F1 score	0,88	0,85	0,90	0,86

Table 4.5: Performance ETC

ETC	**1-1000**	**1-100**	**2-1000**	**2-100**
Accuracy	0,92	0,90	0,90	0,90
Recall	0,94	0,90	0,96	0,94
Precision	0,89	0,88	0,90	0,87
F1 score	0,91	0,90	0,93	0,90

4.7 Demo

It has been decided to make a demo where the category of the tweets has been predicted given a timeline. It has been made use of the *API.home_timeline* method provided by Tweepy[10] that accesses Twitter and returns the most recent statuses, including retweets, posted by the authenticating user and that user's friends. We have selected the text of each of the tweets that appear in the timeline and passed the data to our classifier trained with ETC obtaining the following results.

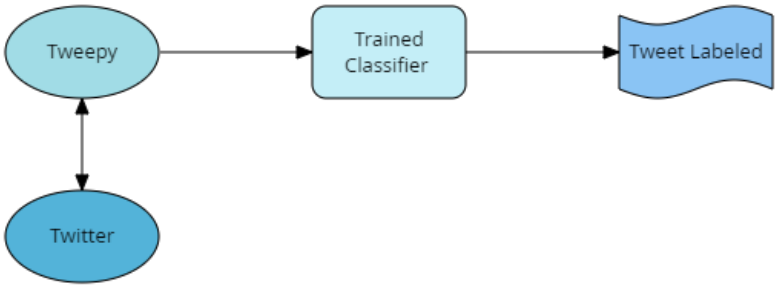

Figure 4.16: Demo Architecture

To check the performance of the classifier, the results that the classifier should predict have been assigned before in order to compare them later. In general, the results obtained are quite acceptable, it can be seen that there is more success in the second way of categorizing, making the correct prediction in 15 of the 18 tweets shown. Regarding the first way to categorize a success occurs in 12 of the 18 tweets shown.

If we analyze the wrong cases it shows that there is a clear tendency to associate with meme all issues related to Catalonia. Sports issues are usually considered as ongoing event and may even associate words such as tobacco with commemorative days since in the training data there is a world day against tobacco.

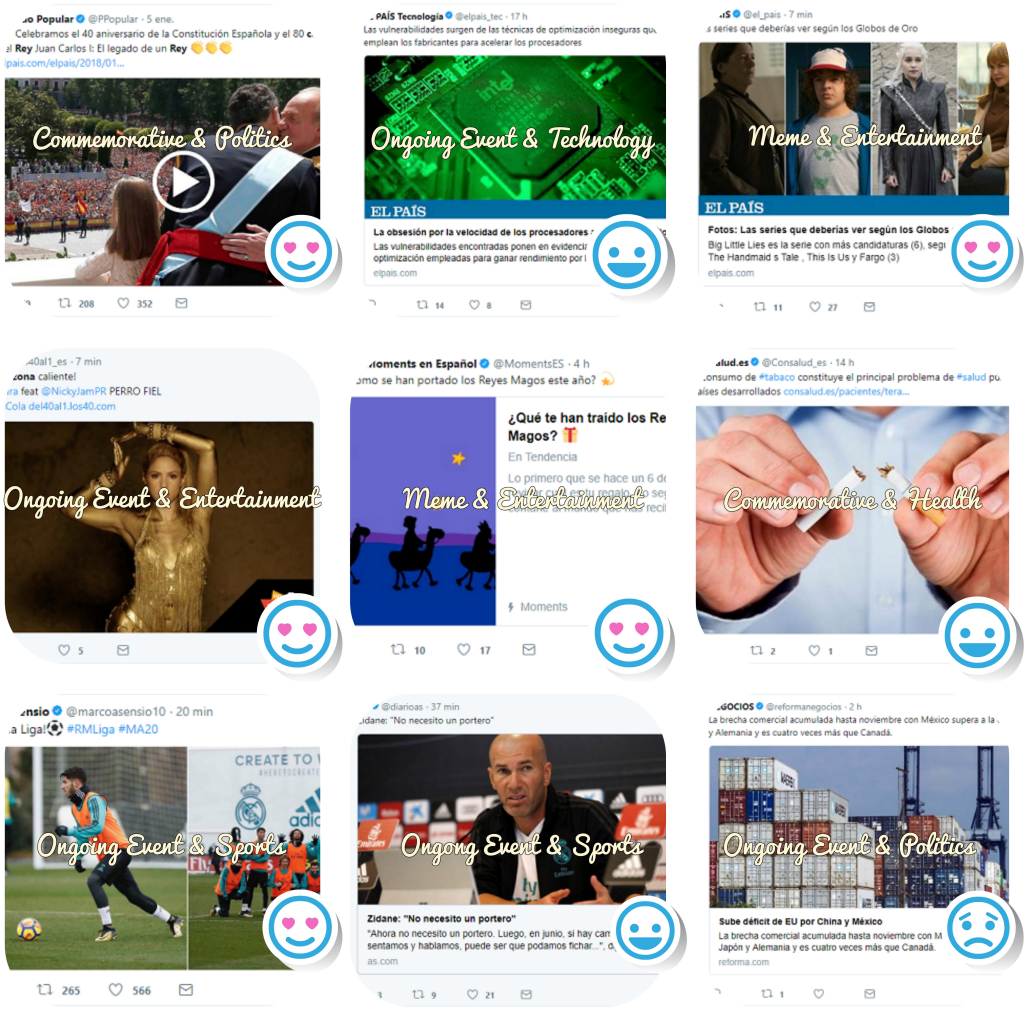

Figure 4.17: Timeline (I)

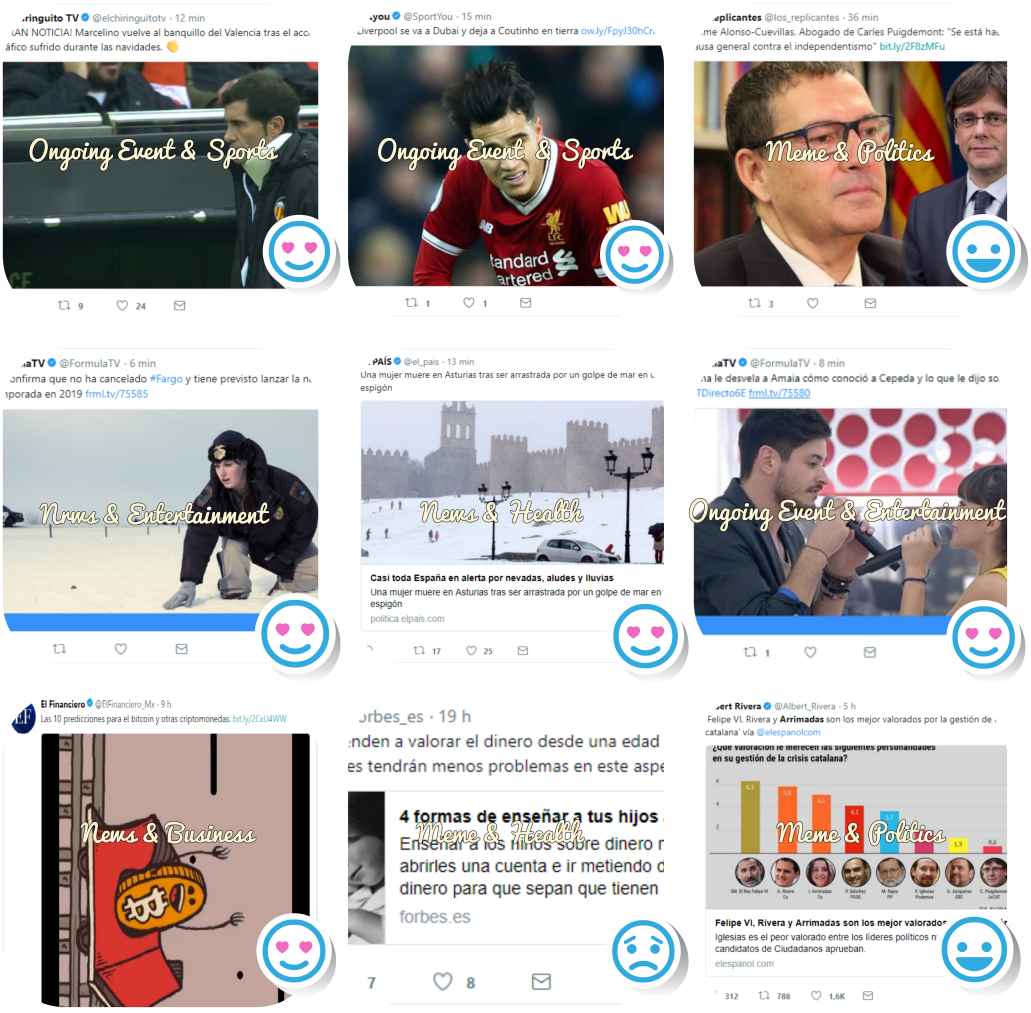

Figure 4.18: Timeline (II)

CHAPTER 5

Conclusions and future work

5.1 Introduction

In this chapter we will describe the conclusions extracted from this projects, problems faced, achievements and suggestions about future work.

5.2 Conclusions

In this section we will analyze in detail the set of tests explained in the previous section to be able to perform a meticulous analysis. In addition, the results obtained will be compared with the work of other authors on the same subject.

There are hardly any differences regarding the size of the data sets, which indicates that the classifier is capable of learning at high speed without the need for large amounts of data. On the other hand respecting to the two ways of categorizing it must be indicated that although the results are practically the same around 90%, the second way of categorizing is more appropriate due to differentiating between six categories is more complicated than with four because there is a greater range of error probabilities.

Table 5.1: Classifier performance (I)

Taxonomy	**N**	**O**	**M**	**C**
Reference	0,67	0,94	0,73	0,55
SVC	0,85	0,97	0,74	0,89
ETC	0,84	0,83	0,85	0,98
MNB	0,87	0,86	0,84	0,94
SVC Improvement	**26,87%**	**3,19%**	**1,37%**	**61,82%**

Regarding the first way to categorize the tweets, it can be seen that it has been improved in all the categories with respect to the reference [1]. Specifically, in the commemorative trending topics, it has improved notably, above 60% due to in the previous results there was a considerable margin for improvement. In addition, ongoing events presented a high value of success that has been improved which indicates that our classifier is quite versatile and effective.

Table 5.2: Classifier performance (II)

	SVC	**ETC**
Reference	0,81	-
Results	0,90	0,91
Improvement	**11,11%**	**12,35%**

It indicates that our classifier has presented a performance with SVC of 91% compared to 81% obtained by other authors [1], which reflects an improvement of more than 11% and compared to our best result obtained of 92 % with ETC it reaches around 12%.

Regarding the second way to categorize our classifier, it presents a 92% performance achieved with the MNB algorithm compared to 78% obtained by other authors [12], which indicates an improvement of approximately 18%. In addition, our algorithm with SVC has a very similar behavior, reaching 92% compared to 74% [12], which is a notable difference of almost 25% which is considerable.

Table 5.3: Classifier performance (III)

	MNB	**SVC**
Reference	0,78	0,74
Results	0,92	0,92
Improvement	**17,95%**	**24,32%**

5.3 Problems faced

During the development of this project we had to face some problems. These problems are listed below:

- ***Twitter API limitations:*** one of the points that has represented the greatest number of problems has been the limitations imposed by the Twitter API. It presents a limited number of requests during a specific time and in our project we made use of the API for monitoring the trending topics and also for the download of the associated tweets. This limitation has taken a great amount of time to download the required information.

- ***Processing time:*** the other point that has also meant great problems when carrying out the project has been the processing time due to the large amount of data to be analyzed. This problem has involved a considerable delay during the whole process when performing the different tests to evaluate the performance of our system.

5.4 Achieved goals

In this section we will summarize the conclusions for an overview of the project. Our main objective was the development of a twitter trend classifier based on machine learning techniques. This objective has been achieved by obtaining a maximum score F1-score of 0.93 which is really high.

One of the objectives of the project consisted of building our own data set through the monitoring of Twitter which has been done successfully obtaining the top 10 of trending topics during a week and download the associated tweets. Moreover, it was necessary to organize the large amount of data obtained, categorize the trending topics and preprocessing the tweets.

The main idea was to implement a twitter trend classifier based on four different categories. In addition, this classifier has been improved by the correct choice of features and algorithms that best fit the system as well as adding another way to categorize the tweets. Gratifying results have been obtained that improve those made by other authors in this matter.

5.5 Future work

In the following section the possible new features or improvements that could be done to the project will be explained.

Combination of both ways of categorizing The objective will be that the classifier allows to predict simultaneously with a high performance the two categorizations presented.

Summary of your own Twitter timeline The idea is to make a brief summary of your timeline since the last time you read it so you can access more quickly to those tweets that are of most interest to you at that time.

Improve the performance of the classifier Despite having achieved a high result, we will look for possible algorithms that allow improving the performance and also include tweets in different languages.

Bibliography

[1] Zubiaga, Arkaitz and Spina, Damiano and Fernandez, Victor and Martinez-Unanue, Raquel. *Real-Time Classification of Twitter Trends.* http://www.damianospina.com/wp-content/uploads/2014/10/trending-topics-jasist2014_preprint.pdf Accessed January 2, 2018.

[2] *Twitter Usage Statistics.* http://www.internetlivestats.com/twitter-statistics/ Accessed January 2, 2018.

[3] Carlos Angel Iglesias. *Exercises for Intelligent Systems Course at Universidad Politécnica de Madrid, Telecommunication Engineering School.* https://github.com/gsi-upm/sitc Accessed January 2, 2018.

[4] *Scikit-learn: machine learning in Python.* http://scikit-learn.org/stable/ Accessed January 2, 2018.

[5] *An introduction to Numpy and Scipy.* https://engineering.ucsb.edu/~shell/che210d/numpy.pdf Accessed January 2, 2018.

[6] *Pandas: powerful Python data analysis toolkit.* http://pandas.pydata.org/pandas-docs/stable/index.html Accessed January 2, 2018.

[7] *Matplotlib.* https://matplotlib.org/ Accessed January 2, 2018.

[8] *Natural Language Toolkit (NLTK).* https://www.techopedia.com/definition/30343/natural-language-toolkit-nltk Accessed January 2, 2018.

[9] *Twitter API.* https://developer.twitter.com/en/docs Accessed January 2, 2018.

[10] *Tweepy.* http://www.tweepy.org/ Accessed January 2, 2018.

[11] *Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures* http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures Accessed January 2, 2018.

[12] Paras Sharmal. *Tweet-Classifier.* https://github.com/Parassharmaa/Tweet-Classifier/tree/master/twc Accessed January 2, 2018.