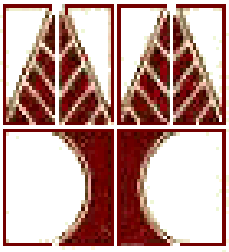
Bachelor's Thesis

DATA MINING ON TWITTER

TRENDING TOPIC CLASSIFICATION

Yiangos Georgiou

UNIVERSITY OF CYPRUS



COMPUTER SCIENCE DEPARTMENT

March 2017

UNIVERSITY OF CYPRUS

COMPUTER SCIENCE DEPARTMENT

Data mining on Twitter: Trending Topic classification

Submitted by

Yiangos Georgiou

Under the supervision of:

Dr. George Pallis

Assistant Professor

Thesis submitted in partial fulfilment of the requirements for the award of degree of

Bachelor in Computer Science

March 2017

Acknowledgement

I express my sincere and deep gratitude to my supervisor Dr. George Pallis, Assistant Professor in Computer Science Department of University of Cyprus, for the invaluable guidance, support and encouragement. He provided me all resource and guidance through this thesis work.

Moreover, I would like to express my gratitude to my family –my parents: George and Euaggelia and my brother: Themis- for their support.

Abstract

We are running through a period where the Internet has a great influence on the society due to the fact that many services became digital through the use of internet in order to make their clients' life more convenient.

This resulted in the rapid growth of Internet and the steep rise in online information. Therefore, the demand to process and learn from these data led to the development of machine learning approaches to serve this need. This kind of approaches are methods of data analysis that automates analytical model building using algorithms that iteratively learn from data in order to discover hidden insights without being explicitly programmed where to look.

Along with the growth of the internet, microblogging sites such as Twitter have increased their popularity and this has, as a result to, generate over 300 million tweets on a daily basis.

In this thesis, we tried to classify tweets that Twitter provides through its API into general categories. The importance to classify these data is to achieve better information retrieval in order to increase convenience for users, companies and governments to collect or monitor information that is close to their interests.

Thus, to do so we had to follow a procedure in order to collect, organize, analyse and classify data. Firstly, we started by collecting twitter's data using twitter search API and stored this information using tweet's topic as a key in order to group them. Secondly, to address this problem we classify topics into 4 categories which are live events, news, commemoratives and group interest. After that, numerical features extracted firstly by using twitter's features such as hashtags, retweets etc. and secondly by using tweet's text and sentiment analysis on tweets in order to obtain extra information which might be useful subsequently. We experiment 3 different approaches for this classification task. The first experiment was a text based classification using bag of words approach and Naïve Bayes algorithm for text classification, the second approach was a feature-based classification using SVM classifier and the last one was a hybrid classifier approach using both textual and numerical values.

Table of contents

Table of Figures

Chapter 1

Introduction

1.1 Motivation

1.2 Contributions

1.3 Challenges

1.3 Outline Contents

Today Big Data considered as valuable information and is treated from different perspectives in order to solve various problems in many fields of expertise. The main importance of Big Data consists on the possibility to improve the efficiency of a business in many areas such as sales, customer service, products and so forth.

Along with Big Data, there is Big Data Analytics where new knowledge can be discovered through analysis, this knowledge can be acquired by monitoring dashboards, reports, and exceptions which concerning a business. Therefore, analysis supposes to help organizations to be more confident and smarter by predicting the likelihood of a future outcome based on historical data through statistical algorithms and machine-learning techniques [1].

Therefore, social media sites attract many Data scientist due to the fact that Twitter has become one of the most popular social media services where a huge amount of users visits this site on a daily basis. Twitter has some unique features that contributed to its success. First of all, the shortness of tweets, which can't be more than 140 characters as a consequence to make the creation of tweets easier and secondly the easiness to broadcast tweets to a large number of users [9].

Twitter also can be considered as a news reporting and marketing tool as you can learn about the world through another person's eyes, for example, you may have access on tweets that are related to disasters such as earthquakes, fires or tweets from a football match. Regarding marketing, many users advertise their services via twitter [9].

Another feature of twitter is that it provides the trending topics. Trending is topics which are being discussed most at the very moment on the site's fast flowing stream of tweets. Twitter define trending topics as "topics that are immediately popular, rather than topics that have been popular for a while or on a daily basis" [14].

Considering the aspects that we mentioned above we can realize that twitter has a real-time nature and that is the reason why many data scientists are interested in twitter. Thus, in order to process a huge amount of data scientist uses Machine learning methods which are an application of artificial intelligence (AI) that uses big data in order to make systems able to automatically learn and improve without any human intervention and adjust actions accordingly. Moreover, Machine learning has to do with the development of computer programs that when it exposed to new information can change and fit on that. The primary objective of Machine learning is to discover hidden patterns that a dataset contains as a consequence to improve the decisions in the future based on the given information.

For this research, we used machine learning and text mining techniques to categorize tweets in specific categories.

1.1 Motivation

Twitter has increased its popularity and this has, as a result of generating over 200 million tweets on a daily basis [14]. Moreover, the usage of social networks changed the last few years. People nowadays use social networks in order to broadcast breaking news, to describe an event or to express their opinion on a matter. In addition, twitter can be considered also as marketing tool and so forth [9]. Thus, a huge amount of useful data is generated each day.

Moreover, Twitter's fast-flowing stream of tweets represent a part of the society and also due to its real-time nature can provide information about topics that are important

at the very moment, this fact attracts companies and users that are interested in finding social trends, activities or opinion on particular fields. Thus, it is very interesting to know what topics are trending.

Nevertheless, Twitter provides a list with trending topics which are topics that are being discussed at the very moment. However, a great amount of trending topics are represented as hash tags, names or even words in a foreign language, but no further information is provided about the events that triggered this social trends, this have as a consequence the difficulty to understand of what these topics are about [14]. That issue makes almost impossible for corporations to gather information related to their interest, so finding out what triggered a social trend can help not only users to monitor trends of their interest but also it can provide useful knowledge for companies and governments.

In addition, due to the fast-flowing stream of tweets users may not be able to see topics that are interested in since these topics will be overlooked due to the fast changing stream of trends. So, with tweets categorization, we have the capability to provide more targeted information to users based on their interests.

These issues motivated us to efficiently categorize tweets into general categories in order to provide extra information about their content and as a result to increase convenience for users, companies or governments to collect or monitor information that is close to their interests. This extra knowledge could be very beneficial in many fields and business. For instance:

- Twitter users would be more satisfied with the content that they provided from Twitter.
- News media could be interested in the early discovery of tweets related to breaking news.
- Governments could be interested in monitoring an event for safety issues.
- Merchants or marketing professionals can monitor what public are interested in by monitoring topics that are related to new fashion trends or new products.

Thus, we were motivated to implement an efficient but quick procedure to cope with the fast-flowing stream of Twitter in order to increase information retrieval for topics that are being discussed.

1.2 Contribution

The task of this project is to classify tweets into general categories using text mining techniques to classify and discover hidden information between a predefined set of categories. Along with text values, extra information gathered considering twitter's features such as hashtags, retweets etc. and also by the sentiment analysis of tweets that each category contains. Furthermore, to address this problem we experimented with 3 classification approaches, the first approach is a text-based classification using text mining techniques, the second approach is a feature-based classification using data mining techniques and classifiers such as SVM and the last approach is a hybrid classification method using both textual and numerical values. The main contributions are as follows:

- Demonstrates a process to classify twitter data.
- Implementation and comparison of three different experimental approaches.

1.3 Challenges

Although, projects related to social media and Big data have many challenges. Working with Big Data can be tricky especially if we have to do with short-micro texts. Along with the growth of social media, the need for efficient techniques to overcome problems related to short texts increased as well. The main issues for this project was, firstly the huge amount of data that we dealt with in order to process a sufficient amount of data for this categorization task, that size of datasets are difficult to be collected and stored at first and even more difficult to manipulated in order to be transformed depending on task's requirements.

Moreover, dealing with textual data makes the task even more difficult due to the fact that text is unstructured and fuzzy. Therefore, another big issue is that Natural Language cannot be represented in a way which a computer can completely 'understand' human language, phrases may contain same words in totally different definitions.

In terms of Twitter, more challenges occur by using Twitter, owing to the fact that it has a fast-flowing stream that contains tweets that refer to million different topics in many

different languages. Thus, the popularity of twitter combined with its multicultural character causes these extra issues that we had to deal with.

Regarding sentiment analysis, it is not perfect at all due to the fact that Natural Language is complex. Many issues such as grammatical nuances, cultural variations, slang and misspellings that occur in tweets it can't be handled easily. Moreover, an even more complex issue is the inability for a machine to interpret the tone within a piece of writing, so term as sarcasm is difficult to defined by machines in order to identify whether the sentiment's polarity is positive or negative.

Moreover, regarding classification, the main issue is the discovery of the patterns that distinguish the differences of our predefined categories and as a consequence to classify our data properly. In addition, the ideal classification method for text is highly controversial in the field of text mining, thus it is essential to understand how this methods/algorithms operates in order to choose the most ideal for a specific task.

Last but not least, this kind of tasks requires high computation power, thus specialized tools are required as well. These tools make you able to manipulate, visualize or process large data using procedures such as MapReduce and WordCloud which are very useful for the processing and visualizing phases of the procedure.

1.4 Outline Contents

The organization of our contents is as follows.

Chapter 1. Introduction

The introduction contains general references on terms such as Big Data, Twitter and also on the fields of data science which were essential to the implementation of this project, these includes machine learning, text mining, classification and sentiment analysis. Moreover, the contribution, the motivation, and the challenges are also being attached.

Chapter 2. Related Work

Chapter 2 focus on an analytical review of the literature, which consists of work related to data mining, more specifically in the area of text classification, information extraction, and social media data mining projects.

Chapter 3. Problem Formulation

In chapter 3 we define the problem and our approach of thinking on this general categorization of tweets. Along with the problem in this chapter, we define the purpose and the scope of this project

Chapter 4. Methodology

Chapter 4 focus on the methodology that applied in order to implement that particular text mining procedure. There are references on what we did on each step and also why it is essential for this procedure. In addition, used technologies are also attached in this chapter.

Chapter 5. Experiment

In chapter 5 we described the experimentation procedure of this project, we explained precisely how we collected our data in a way which allowed us to group tweets into topics. In addition, we present how we predefined our categories and why we chose these particular. Moreover, we illustrate the extracted information about our dataset such as languages used, the number of tweets for each category per day etc. After that, there is a description of how we analyzed tweets in order to collect extra information considering twitter's features such as hashtags etc., text's features such as exclamations etc. and also by the sentiment analysis that applied on tweets. Finally, we demonstrate the algorithms that we were used to classifying tweets and we display our results and procedures for each approach that we have followed in order to address this problem.

Chapter 6. Conclusion

In this chapter, we express our conclusions about the methods that we have followed, mistakes that we could have avoided and also, the knowledge that we have gained in the research of this subject and from its implementation.

Chapter 2

Literature & Related work

2.1 Big data analytics procedure & tools

2.2 Twitter usage

2.3 Classification methods research

2.4 Sentiment analysis

2.5 Work on Twitter topic classification

2.1 Big data analytics procedure & tools

Bid Data is a term that introduced to the computing world the last 15 years due to the fact that huge amount of data generated every day, thus it has become essential for a company to collect a large amount of data in order to make the decision making process easier. E. Geanina in the paper "perspectives on Big Data and Big Data Analytics" defines the concept of Big Data, its importance and the different perspectives on its use [1]. In addition, they express the importance to analyze Big Data and how it will improve decisions making in future. They underline that Big Data is very important for fields such as information technology, customer service, for the improvement of services and products using information from social media, in the detection of fraud in a transaction and in risk assessment by analyzing the financial market. Therefore, new prepossessing methods and Big Data Analytics software are needed in order to deal with that size of data, thus, Apache Software Foundation developed Hadoop which is a fast-growing big data pre-processing platform that allow distributed processing of huge datasets. Hadoop's main subprojects are MapReduce and Hadoop Distributed File System (HDFS), MapReduce is able to distribute data workloads within hundreds of nodes in order to achieve a distributed processing and the HDFS is used to provide

reliability by replicating information in multiple nodes. This paper also underline the reason why Hadoop is ideal to process big data and that reason is that Hadoop provided a solution which is Scalable, Cost-effective, Flexible and Fault tolerance.

A more specific approach on the functionality of Hadoop MapReduce procedure described by Dittrich and Quiane-Ruiz [3]. MapReduce job mostly contains two functions: map and reduce, and uses as input set of key-value pairs <k,v>. Although, it has 3 phases, firstly map function is called for every input producing intermediate results <k',v'>, after that Hadoop MapReduce framework groups those pairs by their keys, the last step is when the reduce function is called for each distinct key producing the desired results. In order to read and write the framework utilize an HDFS which is the open source counterpart of the Google [3] file system.

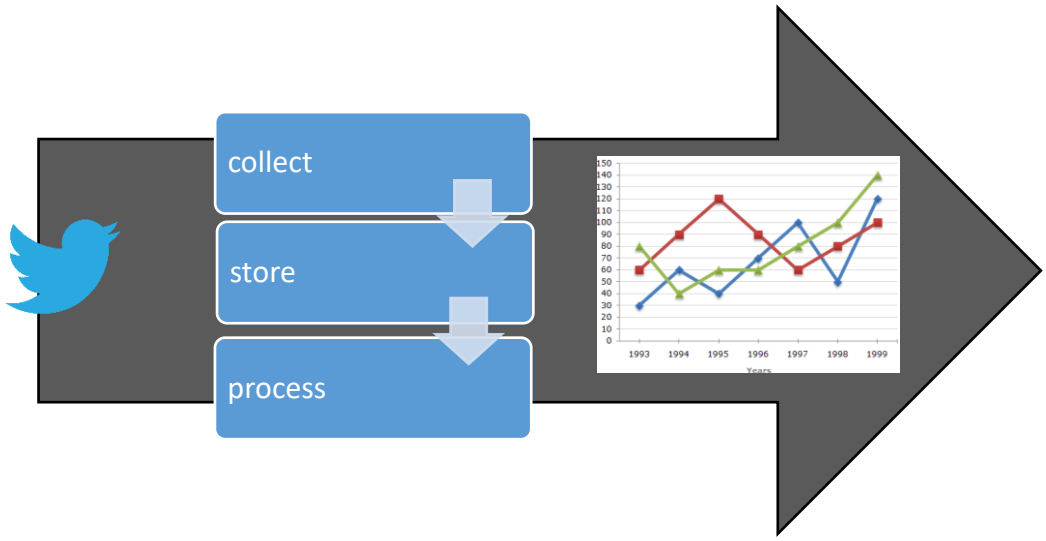


Figure 2.1 Procedure of data mining on Twitter.

In a paper by Yang and Kavanaugh [4], methodologies and open source tools introduced to help researchers to collect, analyze and visualize tweets. They explain the usage of Twitter search and streaming API, the first one can provide you tweets as old as 7 days, on the other hand, Twitter streaming API collects tweets that are posted in real time. Moreover, visualization tools such as Wordle is also described. These visualization tools help into the presentation of data sets.

2.2 Twitter usage

The main types of users' intention for generating tweets considering the paper "Why We Twitter: Understanding Microblogging Usage and Communities" are for daily chatter, conversations, reporting news and sharing information [9]. They discussed the different motivations that caused users to use twitter in their daily life. Thus, they used the motivations that causing the creation of tweets in order to categorize them into categories. This methodology ended up into a general yet precision classification.

S. Asur et al. in their paper "Trends in Social Media: Persistence and Decay" studied trending topics on Twitter [5]. The main objective was to determine the most significant factors that contribute to the creation of these trends [5]. For their experiment, they used twitter's search API to collect tweets which containing particular keywords. Then, they analysed these tweets and they concluded that trends content is mostly news and also that trends have a geometric distribution.

2.3 Classification methods research

In terms of classification, there are various machine learning algorithms and techniques which can be used to address a classification problem, S. Bird, E. Klein and E. Loper in their document called "Preprocessing Techniques for Text Mining - An Overview" defines basic terms of text classification procedures [6], in addition, they presented practical examples also on how you can build a classifier. They showed that supervised classification is a task of allocating the appropriate class label for a given input. Tasks separated into categories based on the number of labels that a text contains, for instance, if the text belongs in only one category then the task is known as single-label and if a document belongs to more than one category then is known as multi-label/multi-class classification. In addition, they highlight the importance of choosing the right features and also the enormous impact of the method that we decided to use in order encode them. Moreover, they explain the metrics that it can be used to evaluate your classifier's results. The simplest metric is Accuracy and it can be measured by dividing the correctly predicted labels with the whole set of labels that contain in the test set, also metrics such as Precision and recall are also explained. In addition, for more accurate metrics they suggest the use of cross-validation which is a solution where we subdivide

the dataset into N subsets in order to train the classifier with every subset except the one that we are going to predict the labels. However, they highlight that to consider this results as valid we have to be careful on the frequencies of each class labels in the test set. They focused on three machine learning methods such as Naïve Bayes, Maximum Entropy, and decision trees. A decision tree is a flowchart which is consists of decision nodes, which base on the feature values lead the input through the branches until we reach a leaf node which will decide the label of the input. After that, they described how a Naïve Bayes Classifier operates, Naïve Bayes is a probabilistic classifier which calculates the prior probability and the feature contributions to decide the class of the input. The last classifier that they described was Maximum Entropy Classifier, who is very similar to Naive Bayes model but rather than using probabilities to calculate the output it uses search techniques to find parameters that will increase precision. Furthermore, they point out that python provides an excellent environment for text processing and feature extraction [6].

Dr. S Vijayatani et al. discussed the applications of text mining and also discussed text mining pre-processing methods and its importance for the procedure [7]. They analysed four applications of text mining which are Information Retrieval (IR), Information extraction (IE), Categorization and Natural Language Processing.

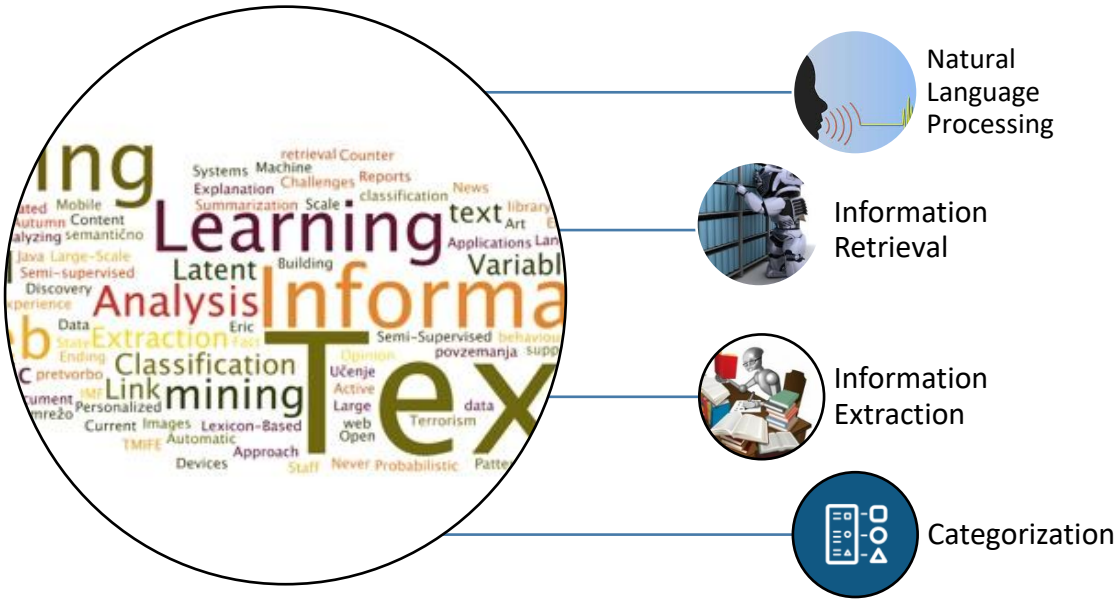


Figure 2.2 Text mining applications.

However, the main objective was to analyse the pre-processing methods. They point out a sequence of steps that have to be followed for a successful categorization. Extraction is the first step where we tokenize the text content into individual words. After that, we have to eliminate all the stop words due to the fact that are not important so we don't need that words in classification, also by removing these words reduces the dimensionality of term space [7]. The next step is to follow a stop word removal methods, they proposed 4 different methods, the classic method where you remove words that contained into a predefined list, methods based on Zipf's Law where you remove words that occur one time only in the document, Term Based Random Samplings (TBRS) developed in order to detect the stop words from web documents by iterating over pieces of data as a result to rank the terms of each chunk based on Kullback-Leibler divergence measure. They have also analysed deeply the Stemming methods and the stemming algorithms for classification processes. Stemming is a method used to identify the root of a word [7] and its purpose is to remove any suffixes, to reduce the number of words, to match the words accurately and to reduce the terms space in memory.

Moreover, A. Taspinar has also analysed the basic concepts of text classifications, he proposed a procedure of 4 steps: Tokenization, Word Normalization, Bag-of-words and classifier Evaluation pointing out some conclusions on how we are supposed to treat our text data and issues that we may have to consider. For instance, we have to consider the delimiter because in most occasions the whitespace will be fine as a delimiter but what if there are some words with a white space in them, punctuation marks are also an issue for classification yet are valuable information for sentiment analysis so they have to be treated with caution. Moreover, words inside blankets may be more appropriate to be treated as a whole. The next step is Word Normalization and is the reduction of each word to its stem form. Although there are some issues we have to consider, the first one is that all the letters have to be converted to lowercase, the second issue is how to treat apostrophes and ambiguous words such as USA, U.S.A., and the US. After the text has been segmented into words and these words have been tokenized and normalized we can use a bag of words approach to model our data [8]. This approach takes into consideration only individual words and provides a subjectivity score for each word, that score defines the polarity of every individual word into predefined categories.

In terms of text classifiers, a paper called “Text Classification and Classifier: A Survey” by V. Korde is a review focused on existing literature and explored fields of text mining [12]. Pre-processing and feature selection methods analysed, the reason to implement these methods is to reduce the high dimensionality of the terms space. Features’ importance score can be determined using metrics such as term frequency, Chi-square etc. in order to exclude words with the lowest importance. Moreover, classification algorithms brief descriptions on the way they operate presented. In conclusion, Korde observed that algorithms are greatly affected by the data source’s quality, each algorithm has its disadvantages and advantages yet support vector machine(SVM) seems to has slightly better results.

Moreover, considering the problem of multiclass classification R. Rifkin and A Klautau stated that One-Vs-All (OVA) scheme can perform as good as other approaches [13]. One-Vs-All is one of the simplest multiclass classification schemes, it trains one classifier for each category and these classifiers trained to distinguish a single class from all the remaining classes, the classifier with the largest positive value is chosen. Therefore, due to its computational simplicity, this scheme is more likely to be chosen for multiclass classifications

2.4 Sentiment analysis

Regarding sentiment analysis on Twitter, E. Kouloumpis, T. Wilson, and J. Moore focused on how to identify positive, negative and neutral tweets. The main objectives were first to examine whether their dataset is useful for sentiment tasks on Twitter and secondly to evaluate the effectiveness of the features for sentiment analysis in Twitter data [10]. For their sentiment analysis they took into consideration text, hashtags, emoticons and other informal intensifiers such as all-caps and character repetitions, they concluded that part-of-speech is not that useful for micro-texts while microblogging features such as emoticons and abbreviations are the most useful [10]. Another important effort for sentiment analysis is by A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau. They provided useful information about their resources and their pre-processing method, they point out that emoticons and acronym dictionary are also needed.

2.5 Work on Twitter topic classification

The need for information retrieval improvement attracted many scientists to deal with that problem. However, topic classification is not easy at all and due to that fact various approaches to address classification problem on twitter are proposed.

K. Lee et al. experiment 2 approaches for topic categorization, the first one was a Bag-of-Words approach and the second approach was a network based classification [14]. Their methodology consisted of four steps, Data Collection, Labelling, Data Modelling and Machine Learning. In order to collect their data, they use Twitter API and the list of the most trending topics to collect tweets that refer to these topics. After they collected their tweets, they randomly selected 768 topics as a dataset. The next step was to define their target categories, they concluded that topics can be represented in 18 categories which are art & design, books, charity & deals, fashion, food & drink, health, humour, music, politics, religion, holidays & dates, science, sports, technology, business, tv &movies, other news, and other. For the labelling two annotators used and in the case of disagreement another annotator used to define a category for the topic. The third step of this procedure was Data Modelling where they explained how they manage to model their text data into the appropriate form for the classification. Regarding Text-based Data Modelling, two pre-processing steps used, at first the documents' hyperlinks removed and in the next stage the documents converted into an array containing the word occurrences for their predefined categories, after that, the frequencies for words used to filter out the common words within the categories. On the other hand, to model their data as a network they used the algorithm from User Similarity Model in order to find the 5 most similar topics for each topic. They assumed that users generating tweets related to their interest, so if the same user tweets about ti and tb, these topics are more likely to be similar. The last stage of their procedure was experiments and results, where they experiment with various modelling algorithms for classification [15]. For text-based classification they used "Naïve Bayes Multinomial (NBM), Naïve Bayes (NB) and Support Vector Machines (SVM-L) with linear kernels" [14]. Naïve Bayes Multinomial had the highest accuracy with 65.36%. In terms of the network-based classification, they used 5 classifier k-Nearest Neighbour, SVM, Logistic Regression and C5.0 decision tree classifier, the highest accuracy reached by C 5.0 decision tree classifier with 70.96%. They concluded that network-based classifiers performed better that text-based classifiers [14], C 5.0 classifier outperformed every text-based method.

Moreover, B. Sriram et al. presented an interesting paper related to topic classification on twitter. Their primary aim was to outperform the traditional “Bag-Of-Words” strategy using meta-data to improve accuracy. Therefore, they extracted extra information which consist of one nominal (author) and seven binary features (slangs, time-event phrases, emphasis words, opinion words, percentage signs and currency). To address the problem they decided that categories such as News, Events, Opinions, Deals and Private Messages are sufficient to represent their dataset. For their experimental procedure they collected 5407 tweets from 684 authors, and the experiments conducted with the Naïve Bayes algorithm. The results showed that 8F (Bag-of-Words with features) outperformed all other classifiers, 8F reached an accuracy of 91% while Naïve Bayes without features accuracy was 70%. Thus, they concluded that using author information and features within the tweets combined with removal techniques can dramatically improve the classification task.

In addition, another approach introduced by A. Zubiaga et al. [16] they assumed that different types of trending topics will differ in terms of generation and social diffusion. They defined a method that can be used efficiently in real-time due to the fact that it requires a small and language-independent set of features that can easily obtained and it requires low computation cost as well. At first, they determined 4 categories for their categorization task which are News, Ongoing Events, Memes and Commemoratives, considering their opinion every single topic can be categorized in one of these categories. In order to collect their data they used Twitter streaming API and the list of the top 10 terms that twitter provide. They collected a total of 1036 unique trending topics which comprised of 567 452 tweets from 348 757 different users. Regarding the labelling process they used four annotators in order to read and determine which category fits most to a topic, for tweets written in a foreign language google translate used to translate these tweets in English language. Their main objective was to prove that patterns observed on users’ behaviour in terms of information diffusion distinguish depending on the motivation of the user to generate this particular tweet. Therefore, to do so they analysed twitter features such as hashtags, the length of the tweet, if tweet contains examinations or question marks, if it contains links and other diversity features related to users, vocabulary, hashtags and language. Thus, in this paper they proved that these 4 categories not only have differences in terms of keywords, text and definition but also have differences on social diffusion using language-independent set of features.

Therefore, to analyse and categorize these features the average values calculated for each topic, this resulted to the generation of 15 values that represents how the information is spread. To evaluate the classification accuracy of their established typology, they performed 2 experiments using a training set with 600 topics and a test set of 436 topics. Firstly, they used Support Vector Machines (SVM) in order classify twitter features, and secondly they used a bag-of–words approach to Classify text. Both text-based and features-based approaches achieved high precision, reaching a percentage of 75% and 78% respectively. They concluded that extra features can be significant beneficial in order to discriminate the type of a topic.

Chapter 3

Problem Formulation

3.1 Problem Statement

With the steep rise in online information, effective Information Retrieval is difficult for some particular information. However, new capabilities and opportunities appeared with the processing of these data. Hence, text classification has a significant role in effective handling and organizing such huge text collections or data.

Twitter has a fast-flowing stream that contains tweets that refer to million different topics in many different languages, this increase the difficulty for users and companies to monitor tweets that are close to their interest. With a general classification of tweets we can provide them the capability to monitor more targeted information. Thus, a precise yet quick classifier is needed to cope with the fast-flowing stream of tweets.

3.2 Problem Formulation

Given as input the Twitter activity *Ti* we are interested in the identification of the category that this tweet belong based on our predefined set of categories, Ci denotes the Category of this particular tweet. *Ti* information is represented by the vector <*ti*,*k*>, where *ti* denotes the unique number of the tweet while *k* stands for tweet related knowledge <*user_id, text, timestamp, if is retweeted, length, number of question marks,*

number of exclamation marks>. Using sentiment analysis we generated additional values for each tweet, thus, a tweet is represented by the vector <*ti,s,k*>, where *s* is the sentiment ratings for *ti* which is represented by the array <*p,n,e,c*>, where *p* denotes for positive, *n* denotes for negative, and *e* denotes for neutral sentiment rating while *c* stands as the average number of the three previous ratings. Thus, having all this information about tweets this research trying to answer the questions:

"Can we identify a tweet's *ti* category by using its text content and its sentiment ratings?"

"To what extend our classifier can predict the ideal category for tweets from our predefined set of categories?"

This can be measured by calculating the accuracy of predictions for a set of Twitter activities T. $accuracy = \frac{correct\ predictions\ made\ in\ T}{predictions\ made\ in\ T}$

In the following sections we introduce a method to answer these questions.

3.3 Objectives

In this thesis, we had to implement an efficient straightforward procedure to classify tweets into general categories in order to improve information retrieval on Twitter. Thus, we had to implement a supervised algorithm to approach this problem. To address this problem we categorized tweets in 4 categories: news, live events, commemoratives, group interest. Thus, the main goal of this project is to classify tweets into predefined categories using different algorithms and approaches in order to find the most appropriate. We experiment 3 different approaches to deal with this topic classification task. Thus, the objectives are as follows:

- Classify tweets by its text content.
- Classify tweets using extracted features.
- Classify tweets using both textual and numerical values.
- Comparison between these approaches.

3.3 Purpose

The purpose of this thesis is to improve the information retrieval of twitter, in order to increase the convenience for users, companies or governments to collect or monitor information that is close to their interests. Furthermore, the main goal is to improve the traditional text-based classification using knowledge that we can extract from tweets.

Chapter 4

Methodology

4.1 Proposed Data mining methodology

Data mining is the process of discovering interesting knowledge from large amounts of data stored in various information repositories. Yet, to discover this interesting knowledge a process of iterative sequence of steps is required. Our methodology

comprised of 7 steps which are: Data Type Determination, Data Collection, Sentiment analysis, Labelling, Data Transformation, Data Mining and Knowledge Presentation. Thus, to implement our project we took into consideration these particular stages.

4.1.1 Data Type Determination

The main objective of Data type determination is the determination of suitable data type and the determination of the relevant data for this particular analysis task. Moreover, useful features are also defined. In this stage, there are some questions that have to be answered before you make your decision on the type of the data. These questions are:

- What are the problem statement questions?
- What is the purpose of the project?
- What has the literature determined as the most appropriate data type?

Taking into consideration these three questions we decided that the most representative and relevant dataset to classify tweets is a dataset gathered through Twitter. The literature determine as the most appropriate way to collect tweets by using Twitter API. Following this collecting method it also provides the capability to store not only the text of the tweet but also some extra information related to twitter interaction features such as Replies, Retweets information, Hashtags etc.

Moreover, by analysing these tweets and twitter features more knowledge for each category could be gathered. To do so, we will take into consideration the textual content of tweets in order to discover any differences between categories related to the size, punctuation, vocabulary and so forth. Moreover, tweets will be used to discover any sentiment differences that discriminate our categories.

Thus, by selecting these type and kind of data we ensure that our requirements will be satisfied.

4.1.2 Data Collection

In order to efficiently collect a large-scale dataset of twitter data which can be manually classified, we chose to create a dataset by collecting sets of tweets where every set it refers to a topic. Therefore, we used one of the main features of twitter that provides a

list of the most discussed topics at the very moment. These topics are the so-called trending topics. Twitter uses an algorithm that focuses more on topics that are being discussed more than usual such as a breaking news story or a recently aired TV show. Trending topic can be a word, hashtag or a phrase that occur in a stream of tweets that generated in a short period of time. Figure 4.1 displays an example of a trending topic and tweets that referring to this particular topic.

Trending Topic: #TheFlash	
User	Tweet
u1	RT @CW_TheFlash: Is Central city ready for two flashes?
u2	RT @MaeAbdu: Barry being annoyed that criminals keep interrupting his and Iris' alone time. Ha! #TheFlash
u3	RT @Mickey175r: Tonight was awesome and definitely heartwarming especially watching Barry and Iris together finally!#TheFlash
u4	Two Flashes are better than one! #TheFlash
u5	#TheFlash returned in style tonight with an excellent, thoughtful episode.

Figure 4.1 Trending topic examples.

Therefore, to gather these data we used an iterating method which comprised of two parts. Firstly, the list of the most recent trending topics is collected using the API and secondly, another API method is used that allows gathering set of public twitter data which contains the given word, hashtag, or phrase. Moreover, we had a list where we were storing trends that we had already seen in order to avoid collecting the same topic twice. We were downloading the list with the trending topics every 5 minutes and we checked if any new trends appeared. As soon as new trends appeared the second method queried the search API for tweets that refers to these particular topics. Our collecting process is as follows:

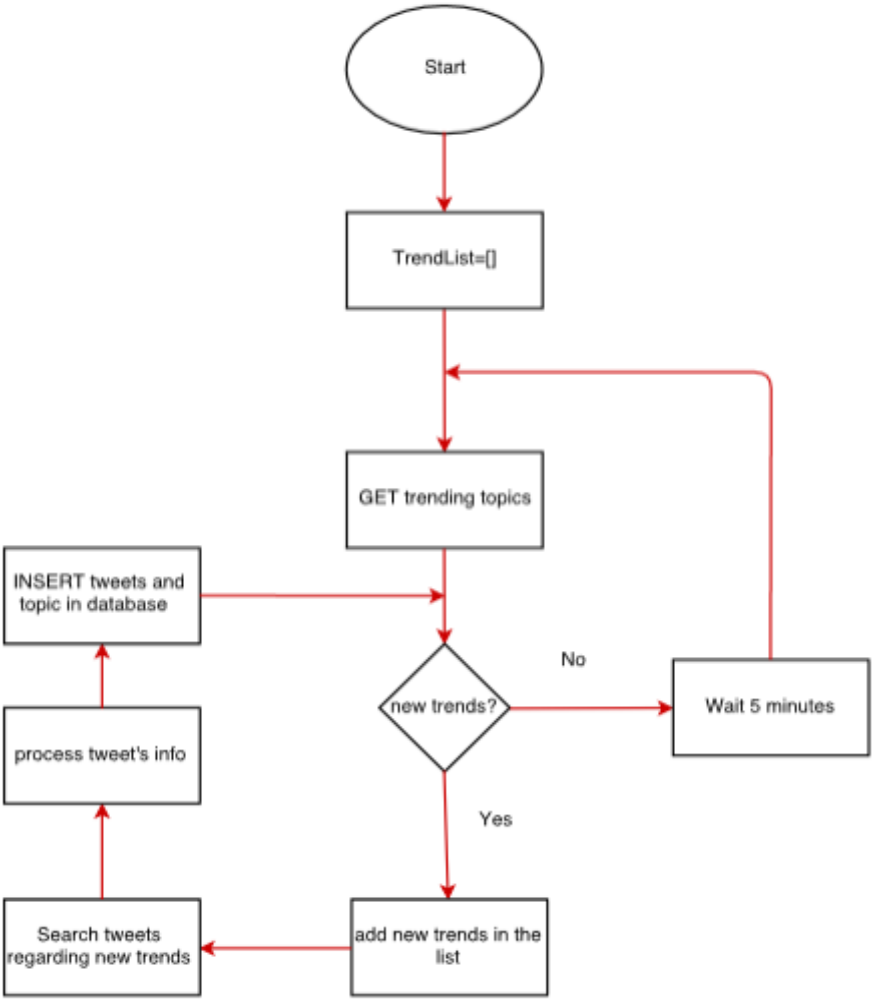


Figure 4.2 Data collecting procedure

4.1.2.1 Database

In order to store this information, we created a database with 3 entities which are tweets, topics, and categories. In tweets table, we stored an amount of information that is related to particular tweets and it provided by Twitter API such as the timestamp and the retweet count, while topic table has the names and the categories of the collected topics, finally, the category table has the names of our predefined categories.

The relation between these entities is as follows:

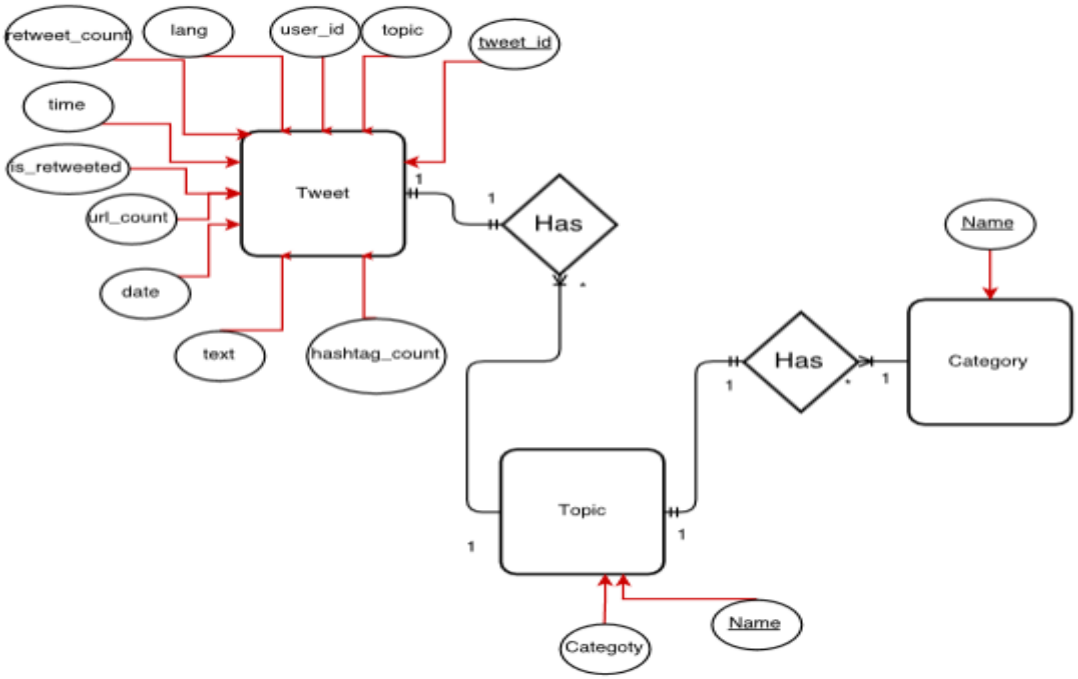


Figure 4.3 Database ER diagram

Tweet table contents:

- Tweet_id: Number of tweet(Primary Key)
- Text: textual content of tweet.
- retweet_count: How many times this tweet was retweeted.
- Lang: In what language the tweet is written.
- User_id: Id of the user that created this tweet.
- url_count: how many urls this tweet contains.
- Is_retweeted: if this user retweet this tweet
- Hashtag_count: The number of hashtags in this tweet
- Time: The time that created
- Date: the date that created
- Topic: The topic that motivated the generation of this tweet (Foreign Key).

Topic table contents:

- Name: topic name (Primary Key).
- Category: The category that belongs (Foreing Key)

Category table contents:

- Name: category name (Primary Key)

Moreover, to implement this database we used My SQL which is one of the most popular Open source SQL database management systems and is developed and supported by Oracle Corporation.

SQL databases are relational, which means that it stores information in a number of tables rather that storing all the information on a single table. The logical model of MySQL contains objects such as tables, views, rows and columns as a purpose to increase the programming environment flexibility. Moreover, it has specific rules to determine the relation between objects, these relations might be one-to-one, one-to-many, unique or required. Thus, with a well-designed database, these rules will guaranty your database's consistency.

Yet another qualification is that MySQL is scalable and easy to use.

4.1.2.2 Dataset

All the tweets were collected from the area of London using trends that appeared in this particular area. Following the above collecting process, we managed to collect 1363 unique trending topics. These topics include 516361 tweets from 311.813 different users. Along with tweets we also saved the number of hashtags and URLs that contained in this particular tweet, the language, if the tweet is retweeted, the timestamp and the topic of the tweet. Moreover, these tweets are written in 55 different languages. The majority of tweets, specifically 444.539 are written in English. In addition, every topic approximately has 378 tweets.

The graphs below illustrates the number of tweets that we gathered each day and the 6 most common languages based on tweets' timestamp and language fields.

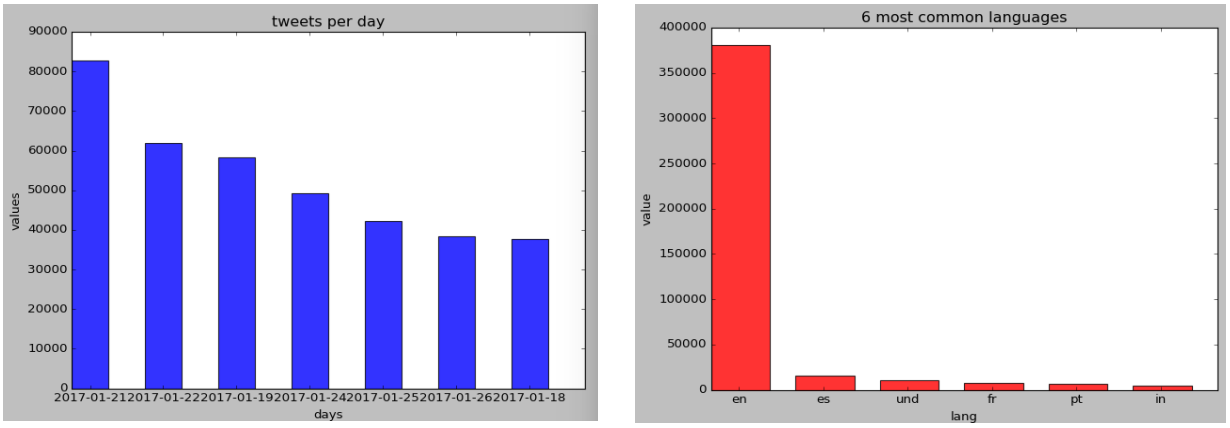


Figure 4.4 Dataset. The first bar chart illustrates the time period that tweets gathered and the number of tweets for each day. Regarding the second bar chart, the most common languages are illustrated.

Despite the fact that tweets gathered from the area of London, approximately 80.000 tweets are written in a foreign language. In addition, there is a number of approximately 25.000 tweets that its language could not be specified so it was fixed to unidentified (und).

4.1.3 Sentiment analysis

For the purposes of our research, we analysed tweets in order to determine their sentiment polarity. To do so, we used the Vader sentiment method which is specialized for sentiment analysis on social media.

Vader is a library written in python. It processes a piece of text and determines whether is positive, negative or neutral.

Sentiment analysis usually takes two forms:

Polarity based: pieces of text classified as either positive or negative.

Valence-based: the intensity of text is taken into consideration.

Vader sentiment analysis is based on lexicons of sentiment-related words. In this method, each word is rated as to how positive or negative is. Vader lexicon structures are as follows:

Piece of text	rating	sentiment
/o:	-1.4 0.66332	negative
0-|	-1.2 0.4	negative
bothers	-1.6 0.4899	negative
brave	2.4 0.8	positive

In the figure above is illustrated that more positive words have higher ratings, while more negative words have lower ratings. Thus, when is desired to analysed a piece of text Vader checks the rating of words that contained in the text in order to define its polarity.

Vader provides four sentiment metrics from a piece of text. These metrics are positive, negative, neutral and compound rating. The first three ratings are standardized to range between 0 and 1, and represent the proportion of the text that occur into those categories, while the compound is an average of these ratings taking values from -1 to 1.

Nevertheless, what makes Vader ideal for social media text? The answer is that it can cope with many issues such as multiple punctuation marks, acronyms, slang and emoticons that occur in tweets. It handles those issues by including these sorts of terms in its lexicon. Thus, Vader can handle:

- Negation.
- Punctuation emphasis & punctuation flooding.
- Capitalization difference.
- Intensifiers such as 'very'.
- Shift in sentiment. (but)
- Emoticons (:), :D)
- slang words(sux etc.)
- acronyms (lol etc.)

Twitter features such as emoticons and abbreviations are the most useful while the part-of-speech is not that useful for micro-texts [10]. Thus, this is the reason why we chose the Vader method to analyse tweets sentiment polarity.

For each tweet, four values were generated related to the extent of how positive, negative and neutral was this particular tweet, while a sum of these values was the fourth value.

However, a high proportion of tweets were unclear about their emotions and this is resulted due to the narrow size of the permitted characters.

Text	Neg	Neu	Pos	Comp
#10YearsOfOsnapitzari RT @GrandeAware: This channel gave us these heavenly videos and vocals.	0.0	0.75	0.25	0.61
HSBC to close 62 more UK branches this year, blaming online banking. Full list here. #HSBC #BranchClosures#UKbanks	0.15	0.84	0.0	-0.49
RT @njking28: C'mon City! Keep it up! #MCITOT @ManCity	0.0	1	0.0	0.0
My heart and my feels. I'm crying #10YearsOfOsnapitzari	0.27	0.72	0.0	-0.47

Figure 4.5 Example of sentiment ratings.

Sentiment analysis ratings illustrated in Figure 4.3. There are 4 examples, the first example refer on a memorial topic. This user seems to be grateful for the heavenly videos and vocals that this channel gave. This tweet has definitely a positive sentiment and it has been rated appropriately. The second example is a tweet that refers on the closing of 62 branches for HSBC which is a British multinational banking and financial services holding company headquartered in London, the sentiment in this particular tweet is definitely negative. As we have mentioned before there are many tweets which their sentiment is unclear, in this specific occasion tweets are rated as neutral. Moreover, sentiment analysis is not perfect at all. The last example is wrong annotated as negative. As we can see it contains the word ‘crying’ which in most of the times it used to express negative feelings, although, on this occasion, it used to express thrill and joy.

For the next steps of the research, we used all the values that we generated through sentiment analysis.

4.1.4 Labelling

In this step, we introduce the categories that we used in this research and the methodology that we followed to determine the category for each topic.

Firstly, we decided not to use categories that describe the actual content of topics such as football, movie, business etc. due to the fact that users discuss any kind of information on Twitter. So, in order to implement this kind of categorization many classes are needed to describe precisely the content of tweets.

As we mentioned in literature survey using the motivations that caused the creation of tweets in order to categorize tweets can ended up into a general yet precision classification [9]. Thus, analysing the motivations that caused users to generate tweets we can achieve a precise classification using few general categories. After we made an extensive research on related work on twitter we decided to identify 4 classes for this particular classification task. The categories are news, live events, group interest, and commemoratives topics. Using these general categories we can classify all the trending topics [16].

In order to perform the classification experiments and to analyse the trending topics that was gathered, first we had to manually classify topics according to its type within the set of our predefined categories. Thus, we started the annotation process, where we carefully read tweets that comprised a trending topic to determine the better fitted category. We were reading the tweets of a topic and also we were searching in the internet extra information about this topic in this specific date and time. Moreover, for tweets in foreign language the Google translate used in order to be translated into English. Furthermore, there were topics that could be assigned to more than one category, on this occasion we chose to assign them to the category that fits most.

More specifically the definition of the proposed set of categories is as follows:

News: Twitter also can be considered as news reporting site as you can learn about the world through another person's eyes [6]. Therefore, a category that contains news was essential for the procedure. We categorize a topic as news when newsworthy sites such as guardian and BBC had reported this trending topic in the same specific date. More specifically, in this category, the majority of topics refer to natural disasters (earthquakes, fires), political developments, economy and shocking real life stories. For

better understanding of the tweets that contained in news category several examples are provided:

Topic	Tweet
Papua New Guinea 22/1/2017	RT @CGTNOfficial: #BREAKING 8.0-magnitude earthquake strikes North Solomons, 36km west of Arawa, Papua New Guinea, no Tsunami alert yet
El Chapo 20/1/2017	RT @ABC: Drug lord El Chapo Guzman has arrived in U.S. after extradition from Mexico, due in Brooklyn court on Friday.
#TRUMPWALL 26/1/2017	BBC News - Trump orders wall to be built on Mexico border

Figure 4.6 News topics examples the first one refers to an earthquake that took place in Papua New Guinea. The second topic refers to the capture of El Chapo. The last one refers to the announcement of the decision for the wall between Mexico and US.

Live events: Twitter real-time nature promotes the generation of tweets that refers to live events. These kinds of topics created by a group of people who tweet about a specific live event while it happening. For instance, it is when people using the same hashtag to talk about a football match or a TV show. The majority of tweets in this particular category it refers to football matches, TV show, and festivals.

Several examples for this category are provided:

topic	Tweet
Ander Herrera 26/1/2017	RT @Devils_Related: Congratulations, Ander Herrera on your 100th appearance for United! #MUFC
#BOUWAT 21/1/2017	RT @afcbournemouth: There's the whistle! It's a point on the board, after we twice came from behind to level
#CBBUK	Is it just me who noticed that #CBBSpeidi said one minute they wished #CBBJamesC had left, but to his face say they're glad he stated #cbbuk

Figure 4.7 Live events topics examples the first one refers to Ander Herrera and his 100th appearance for Manchester United. The second topic refers to the match between Bournemouth and Watford. The last one refers to Big Brother Tv show.

Group interest: This category refers to a proportion of topics that were triggered by a viral idea initiated by a person or a group of people that were popular enough to broadcast this information widely as a result to become trend topic. More specifically, this category contains topics that weren't newsworthy enough to be reported in news media nor were live events but for a large number of users were funny and attractive. The majority of topics in this category are funny topics related to music, movies or celebrities. In addition, topics related to the releasing date for new video games and songs are also classified in this category. Figure 4.8 have example of topics that comprised in this category:

Topic	Tweet
#MyFirstWordsAsPresident 18/1/2017	RT @_lola_bee: I'm just as shocked as you are
#80sSongsForCountries 20/1/2017	Highway to Haiti #80sSongsForCountries, #80sSongsForCountries While My Qatar Gently Weeps
#noisycelebs 25/1/2017	Orlando Boom!!! #noisycelebs, Donald Trumpet #noisycelebs

Figure 4.8 Group interest topics examples, the first one asking for users to express their first words if they were elected for president, the second topic is related with 80s songs but with little changes, the third is related with celebrities that have noisy names.

Commemoratives: This kind of topics refers to a specific person or event that took place in the past on the given day. More specificity, these trends are generating due to the fact that a large number of users remembers an event that happened before and generates tweets about it. This category is the least frequent and the majority of the topics refer to birthdays, anniversaries of an event and memorial days.

Some of these topics are illustrated in Figure 4.9

Topic	Tweet
Eric Cantona 25/1/2017	RT @Sporf: 22 YEARS AGO TODAY: Eric Cantona kung-fu kicked a Crystal Palace fan.
#Happy46thGary 19/1/2017	#Happy46thGary @gary Barlow happy birthday gary have a fantastic day you truly deserve it love from all the thatters on twitter xxx
#BeliebersStayForever	RT @Bizzlesmodel_: We are unbreakable, 3 years ago today we was all tested who were the real beliebers and who wasn't #BeliebersStayForever

Figure 4.9 Commemoratives topics examples, the first one refers to Eric Cantona and its kick on a fan. The second topic is related with the birthday of Gary and the third is about Bieber.

By the end of this process 1145 topics have classified into one of our predefined categories while 217 topics couldn't be classified due to the fact that its content wasn't clear or its tweets had no coherence between them, for example, if the trending topic was a name there was chance to gather tweets that didn't talk about a same incident or event. Therefore, 527 trending topics were annotated as Live events, 289 as Group interest, 32 as Commemoratives and 297 classified as news.

Thus, following this method from the 516361 tweets that we have collected, we managed to categorize 434014 of them.

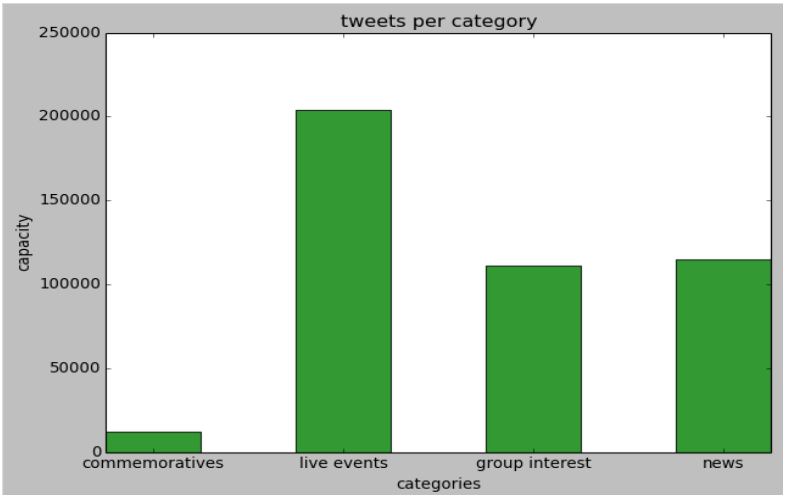


Figure 4.10 this bar chart illustrates the number of tweets per category

4.1.5 Data transformation

The purpose of this step is to convert data or information from its original format to another, most of the times into the required format for the new destination system. The primary aim of this transformation is to increase the efficiency of our predictive process. In addition, patterns can be interpreted and recognized easier. There are several strategies for data transformation such as Smoothing which work to remove noise from data using regression and clustering techniques. Moreover, another data transformation strategy is the attribute construction where new features are constructed and as a result the reduction of the number of variables that we take into consideration [17].

For this kind of tasks, appropriate tools are required in order to manipulate this size of data, thus, tools like Hadoop has developed to satisfy this need.

Hadoop is developed on two main part. The first part is a file system called Hadoop Distributed File System (HDFS) and the second part is the Map Reduce framework.

The HDFS is an optimized file system for distributed processing of large-scale data. On the other hand, Map Reduce framework operates on two main phases to process the data, the first phase it is called “map” and the second it is the “reduce” phase [2].

The MapReduce model comprised of two functions: map (ki,vi) and reduce (kj,vj). Every user can implement these functions using their own processing function by defining a customized “map” and “reduce” functions.

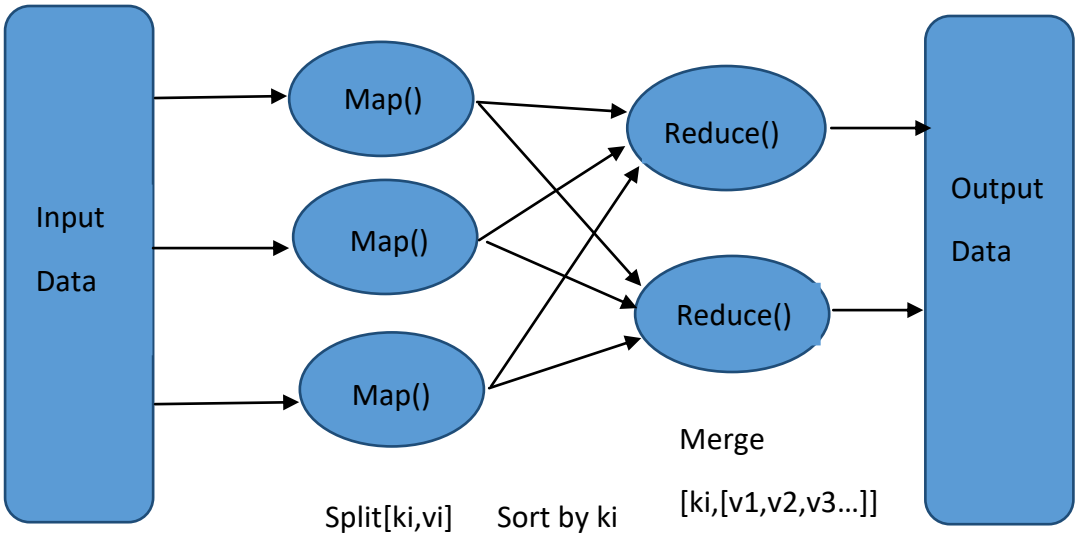


Figure 4.11 Map Reduce Procedure

In this stage, we implemented two MapReduce procedures. Firstly, for text analysis purposes we developed a word count procedure in order to display the top terms occurring in each category. Secondly, we created another MapReduce procedure to analyse twitter features.

Text analysis

In terms of text analysis, a word count MapReduce procedure developed in order to discover the most frequent and important vocabulary for each category. To consider a term important it should be frequent for only one category and also it should be a word that is related to this particular category. For instance, the word “breaking” is more likely to appear in the category that is related to news, while the word “birthday” is more likely to appear in commemoratives topics.

To implement this task we created the two main functions that are needed for this kind of procedures. Our map function was taking as input the text content of tweet and its category. Then in order to discover frequent words for each category, first we removed all the stop words that contained in tweets, secondly we remove words that tend to be frequent, because, we frequently use them in our everyday life. So, we created a list with words that we want to exclude from our analysis. That list comprised of words such as “morning”,”night”,”of” etc. The next step was to remove links, users’ mentions and tokenize the text. Map function produces an output of this form <word category, 1>.

Reduce function takes as input a list of values for each key in this occasion for each distinct word in a category and summarize all the values in the list. Using this method we managed to achieve the most frequent yet important words for each category. In the table below the 15 most frequent are displayed for each category.

News	Live events	Group interest	Commemoratives
Trump	tonight	love	happy
breaking	goal	time	today
news	win	people	day
deal	today	trump	birthday
labour	game	day	years
president	Day	Video	one
loan	match	great	ago
city	cup	why	love
west	watch	very	baby
Police	#ausopen	Think	celebrate
Brexit	team	world	Fan
signing	show	march	Year
article	penalty	game	Tribute
court	Play	help	Great
sky	ball	year	old

Figure 4.12 Important words for categories

These words have strong relations with the definition of our categories. Thus, our categories can be described with these particular words.

News related topics are more likely to talk about politics, economy, and justice, so it was no surprise to appear words like Trump, president, labour, Police or Brexit. Moreover, words like “breaking” that are frequently used to describe news also appeared.

For topics related to Live events, there are words that refers to actions like win, watch and play, moreover, there are time related words which refers to the present like tonight, today and word. As we can realize these words are strongly related to events like football and tv shows, which are being watched live

Regarding Commemoratives topics, both time-related words such as year, ago, old and celebrating words like happy and celebrate appeared. The typology of this category is clear, most of the topics have to do with celebrations.

In terms of group interest topics, this category doesn't have a clear typology due to the fact that topics in this category can cover a wide range of interest. Although, words like video and games shows that people like talking about this topics.

Twitter features analysis

Regarding twitter features, we implemented a second map-reduce procedure to generate features regarding the textual content of tweet, twitter features and sentiment ratings of tweets. We did that in order to understand whether the features we use can discriminate the type of topic. So, for each topic the average number was calculated for every feature using this equation:

$$Aver(topic\ x, feature\ j) = \frac{\sum values(xj)}{number\ of\ tweets\ in\ x}$$

Moreover, the diversity of words, hashtags, retweeted and language also calculated using Shannon's diversity index:

$$SI(topic\ x, feature\ j) = \sum pilnpi \qquad \text{N}pj = \frac{nji}{Ni}$$

That provide the Shannon's index of feature f for trending topic t where nj is the population of j value on t and N is the different values contained in t.

For this task our map function produce an output of the form <topic,[features]>. Then our reduce function group these values and produce a result of 16 values that represent a specific topic.

The table below illustrates the used features and metrics:

Feature	Description
Sentiment ratings	
Negative sentiment	Average negative rating ratio of tweets
Positive sentiment	Average positive rating ratio of tweets
Neutral sentiment	Average neutral rating ratio of tweets
Compound	Average summary of sentiment ratings of tweets
Textual content features	
Length	Average length of tweets
Examinations	Average exclamation signs in tweets
Questions	Average Question signs in tweets
Word diversity	Diversity of words in a topic
Language diversity	Diversity of languages in a topic
Twitter Features	
Links	Average number of links
Hashtags ratio	Average number of hashtags
Retweets depth	Average level of retweets depth. The retweet level of a tweet is the number of retweeting users before it reached the current state
Retweets ratio	Ratio of retweets
Hashtags diversity	Diversity of hashtags in a topic
Retweeted diversity	Diversity of users the was being retweeted
Topic repetition	Average number that the trending topic occur on tweets

Figure 4.13 Gathered features

In order to display the differences between those categories we visualized these features using box-plots.

Textual content features:

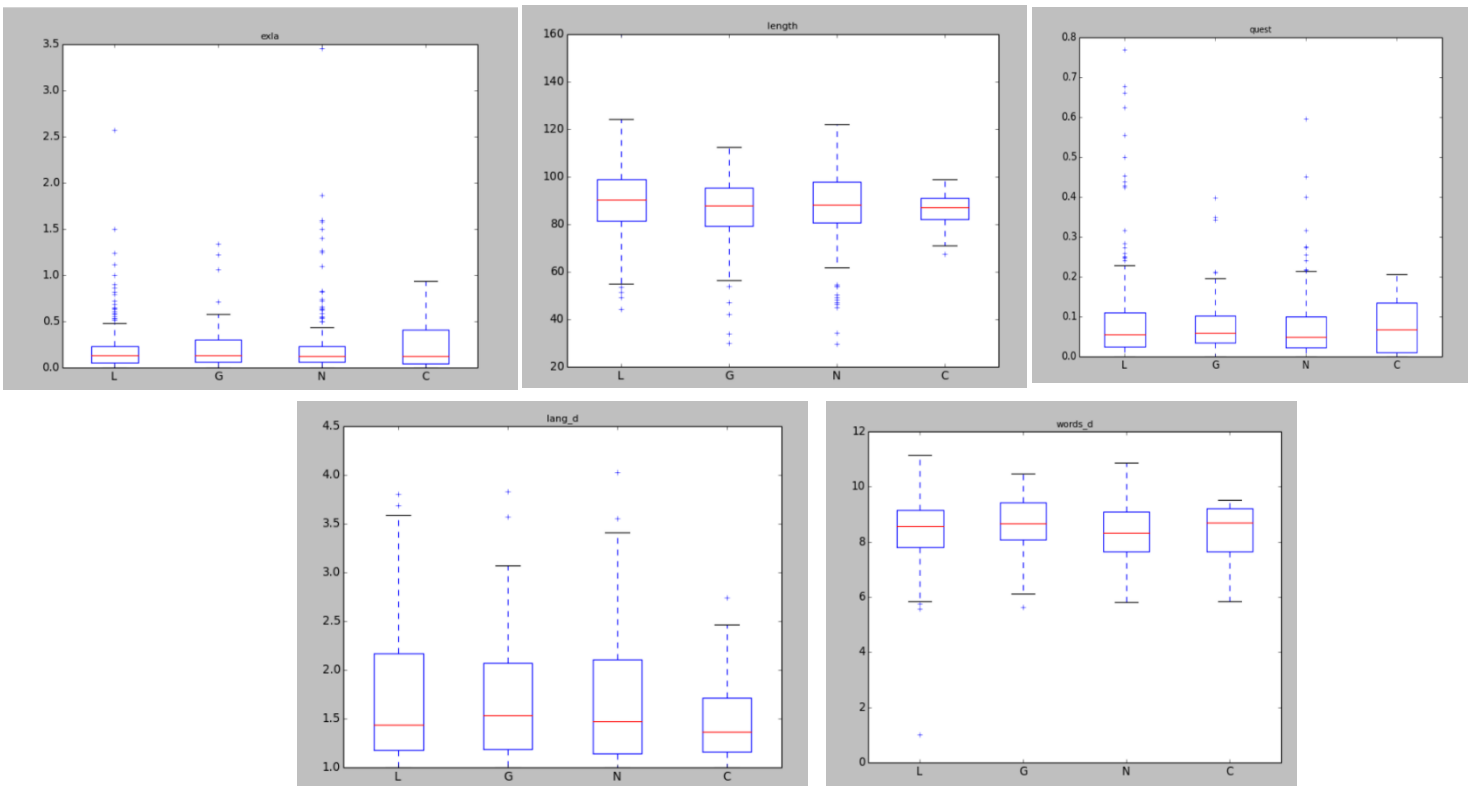


Figure 4.14 these charts illustrates the distribution for textual values in four categories (live events, group interest, news, commemoratives). The first chart refers to exclamation marks, the second refers to the length of tweets, the third is about question marks. The last two plots refers to word and language s diversity respectively.

Small differences occur in textual values. Commemorative topics seem to have a higher average number of exclamation and question marks while live events and news seems to have the smaller number of exclamation and question marks. Moreover, group interest topics are slightly shorter. In terms of diversities, more languages used to describe live events while the less number of languages is used to describe commemorative topics.

Sentiment rating features

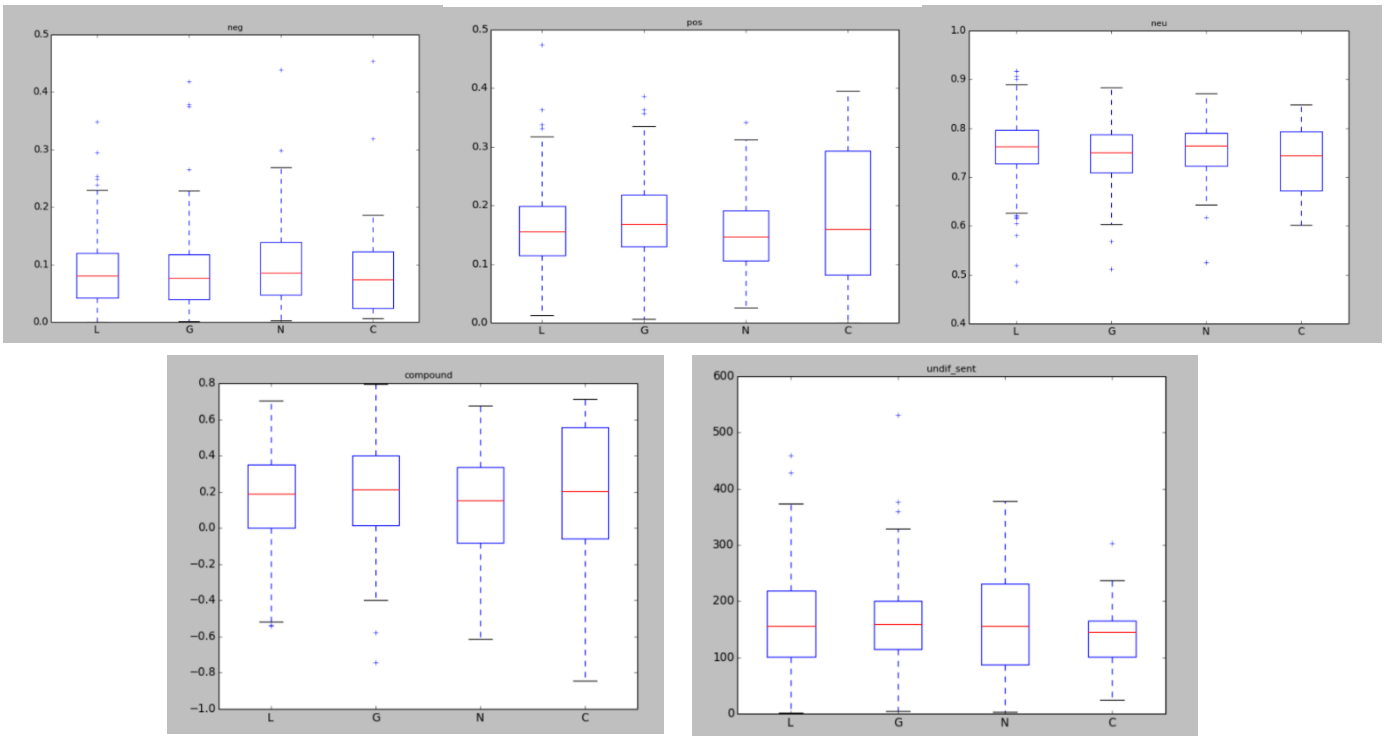


Figure 4.15 these charts illustrates the distribution for sentiment values in four categories (live events, group interest, news, commemoratives). The first chart refers to the negative values, the second refers to the positive values of tweets, the third is about neutral values. The last two plots refers to the sum of these sentiment values for each topic and to the number of tweets that didn't express any sentiment.

Sentiment distribution is different for each category. The first plot that refers to negative sentiment shows that trending news related topics have the highest average negative sentiment. However, our categories negative sentiment values are relatively low regarding the values of positive and neutral sentiment. The second box-plot show that commemorative topics comprised with high positive content, also group interest topics are more positive than topics that contained in news and live events categories. Moreover, neutral sentiment ratings are the highest, this shows that in tweets many neutral words occur. The compound box-plot show that the most positive overall value achieved by commemoratives topics while the most negative achieved from topics that contained in the news category. Regarding live events and group interest topics, its compound values seem to be greater than news and less than commemorative topics. As we mentioned in the previous steps, there is a large proportion of tweets that its sentiment was 0 due to the fact that it didn't contain any sentiment. The last box-plot

illustrates that live events have the higher number of tweets without a sentiment while commemorative topics seem to have the least.

Twitter Features:

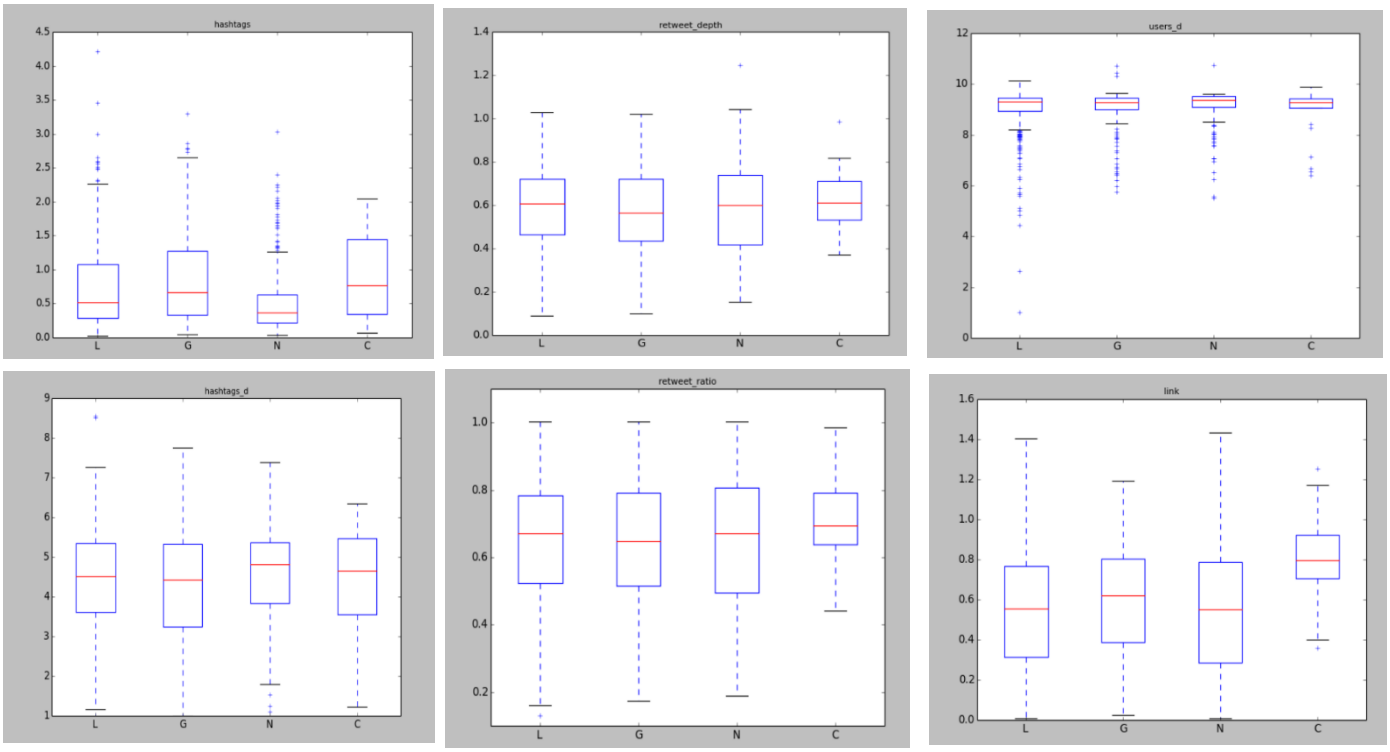


Figure 4.16 these charts illustrates the distribution for twitter in four categories (live events, group interest, news, commemoratives).

These plots shows features that gathered by observing twitter interaction syntax which includes hashtags, retweets etc. The first plot shows that the hashtag ratio for commemoratives and group interest are significantly higher than the hashtag ratio for news and live events. Retweet depth is almost the same for the live event, group interest and news categories, however, the use of retweets seems to be more frequent for group interest topics. Same as retweet depth the diversity of users is almost the same for all the categories. On the other hand, the diversity of hashtags is higher for news and live events while for commemoratives and group interest the diversity is lesser. The fifth box-plot shows the number of that were retweeted in topics. Commemorative topics seem to comprised of higher number of retweeted tweets rather than tweets in the

remaining categories. The last box-plot shows that links are more likely used in commemoratives and group interest topics.

4.1.6 Data mining

Data-Mining step involves repeated iterative applications of predictive data mining methods [17]. The primary objectives in this stage are verification and discovery. Firstly verification is the need to verify the user's hypothesis, while discovery is the goal for new patterns discovery by the system.

Moreover, data-mining involves fitting models to observed data [17]. In order to do so data-mining methods are used, these methods are tried and tested models related to machine learning, clustering, and regression.

The goal can be achieved using methods such as Classification, Regression, Clustering, summarization and dependency modelling [17]. For this project, we used a classification method in order to predict the category of the tweets. Classification is a method where a learning function after sufficient training can determine the category of data items into a set of predefined categories.

In Brief, "this step is the searching for interesting patterns in a particular representation form of the data" [17].

In this particular project we used 4 different supervised algorithms:

- Naïve Bayes
- Stochastic Gradient Descent (SGD)
- Support Vector Machine (SVM)
- Random Forest

Supervised learning is a method where you have to provide input and output variables into a learning algorithm which will start learning from this data using both values.

$Y=f(x)$

The primary aim is to map information well enough that when a new record arrive the algorithm would be able to define its output variable.

It is called supervised learning due to the fact that the process of an algorithm learning from a dataset owing the target values, can be likened with a teacher supervising the learning process. Therefore, the process stops learning after it reach a sufficient level of performance.

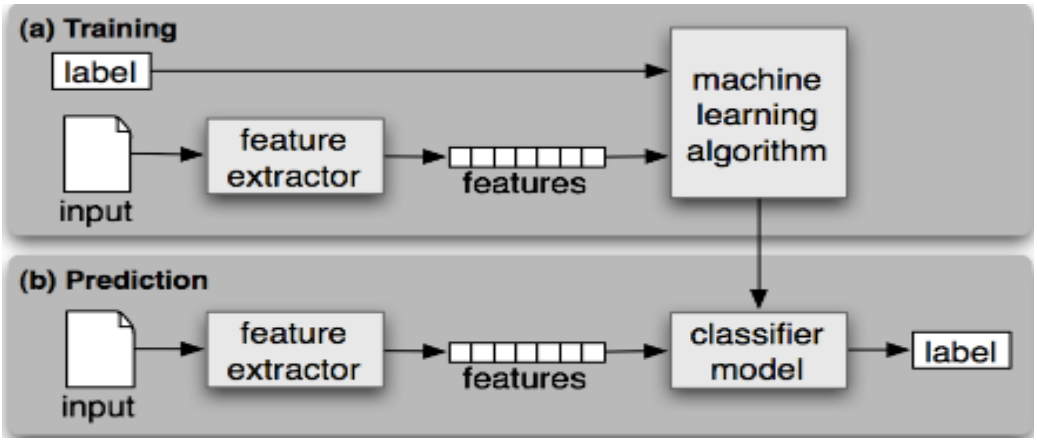


Figure 4.17 Supervised Classification. For training both features and labels are provided to the algorithm (a) . On the other hand only the features are passed through the classifier as a purpose to predict the labels. [17].

Supervised learning problems can be grouped as classification and regression problems. Classification is when the classifier is called to determine the exact category (value) of an input and regression is when the output is a real value, for instance dollars or weight.

4.1.6.1 Naïve Bayes

Naïve Bayes is a probabilistic model which represent text as a bag of words. Text classifiers often represent a text document as a bag of words which is a simple representation due to the fact that it only includes the words without to keep any information about the word order.

Naïve Bayes methods are a set of supervised learning algorithms based on Bayes' theorem with the naïve assumption of independence between every pair of features [18]. The process to choose a label from an input value is by calculating the prior probability of each category, which is determined by the frequency of each category in the training test, then it combines the contribution from each feature that the input contains to

estimate the likelihood for each category [6]. The category whose likelihood estimate is the highest is chosen to the input value. However, the probabilistic models are more than one, due to the fact that a document can be represented in several ways. For instance, two of the most used models are:

- **Bernoulli document model:** text represented as vector with binary elements assigning 1 if the word is present in the document and 0 if not.
- **Multinomial document model:** text represented as a vector with integers which illustrated the frequency of the particular word in the document

Word	Probability Category 1	Probability Category 2	Probability Category 3	Probability Category 4
W1	P(C1|w1)	P(C2|w1)	P(C3|w1)	P(C4|w1)
W2	P(C1|w2)	P(C2|w2)	P(C3|w2)	P(C4|w2)
W3	P(C1|w3)	P(C2|w3)	P(C3|w3)	P(C4|w3)
. . . .	. . . .	. . . .	. . . .	. . . .
W5	P(C1|wn)	P(C2|wn)	P(C3|wn)	P(C4|wn)
Total	∑P(C1|W)	∑P(C2|W)	∑P(C3|W)	∑P(C4|W)
Probability of Input Document	$\frac{\sum P(C1\|W)}{n}$	$\frac{\sum P(C2\|W)}{n}$	$\frac{\sum P(C3\|W)}{n}$	$\frac{\sum P(C4\|W)}{n}$

Figure 4.18 Table of words occurrences and probabilities

To be able for the classifier to make this assumption a few steps are required. At first, a table of words occurrences and probabilities is constructed which contains the probabilities of words for each category. Considering the occurrences of the table the classifier calculate the posterior probability of a particular word of being annotated in a category by using Bayes' Theorem (1):

$$P(Category|Word) = \frac{P(Word|Category)P(Category)}{P(Word)} \quad (1)$$

Thus, in order to calculate the probability of a particular word wj being in a particular category Ci, the prior probability of a category P(Ci) is required and can be computed using (2):

$$P(Ci) = \frac{Total\ words\ in\ Ci}{Total\ words\ in\ Training\ set} \qquad (2)$$

In addition, the normalizing constant of words is also needed and can be calculated from (3):

$$P(wj) = \frac{\sum occurrence\ of\ wj\ in\ all\ categories}{\sum occurances\ of\ all\ words\ in\ all\ categories} \qquad (3)$$

To calculate the probability of a particular category Ci to contain a word wj it can be calculated using equation (4):

$$P(wj|Ci) = \frac{occurrence\ of\ wj\ in\ Ci}{\sum occurrence\ of\ all\ words\ in\ Ci} \qquad (4)$$

After we manage to calculate the prior probability P(Category), the likelihood P(Word| Category) and the normalizing constant P(Word) we can use the Bayes' theorem to calculate the probability of a particular word of being annotated in a particular category and construct the table which contains this information.

After all the probability cell for each word are filled, then the probability of a new record document can be calculated by dividing the sum of each probability column with the length of the query n [19]. The equation is as follow (5):

$$P(Ci|Document) = \frac{P(Ci|w1, w2, w3, \dots, wn)}{n} \qquad (5)$$

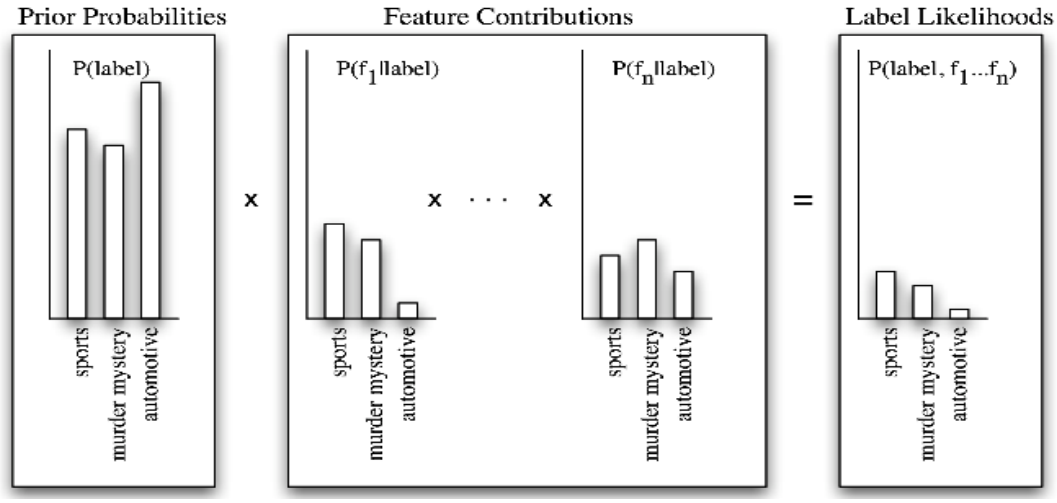


Figure 4.19 Calculated label likelihoods with Naïve Bayes. Firstly, it calculates the prior probability for each category, secondly every features it contributes to the likelihood estimation. The result is calculated by multiplying the prior probability of the particular category with the set of features' probabilities.

4.1.6.2 Stochastic Gradient Descent

It is an effective yet simple approach to discriminative learning of linear classifiers. Stochastic Gradient Descent (SGD) is a stochastic approximation of the gradient descent optimization method and its main purpose is to minimize the objective function. The objective function also is known as loss function or its negative known as reward function. Thus, depended on the objective function, SGD has to minimize the loss function or to maximize the reward function.

Moreover, this is an iterative method which iterates through the training set monitoring the true and the target output in order to make the appropriate changes, thus, in every iteration, the weights are being updated and as a result to minimize the loss function. It stops to iterate when it converge or if it reaches the target error.

In pseudocode, SGD can be represented as follows

- Insert parameters w and learning rate η.
- Iterate until it the target minimum is achieved
- For i = 1,2,…,n do:
- $w+1 := w-\eta\Delta Q(w)$

4.1.6.3 Support Vector Machines

It is another supervised machine learning algorithm which can be used for classification and regression challenges. SVM is a method where every record is represented as a point in n-dimensional space. The dimensions of the problem determined by the number of features of each input. Then, we perform a classification by finding a line, plane or hyperplane in order to separate the classes.

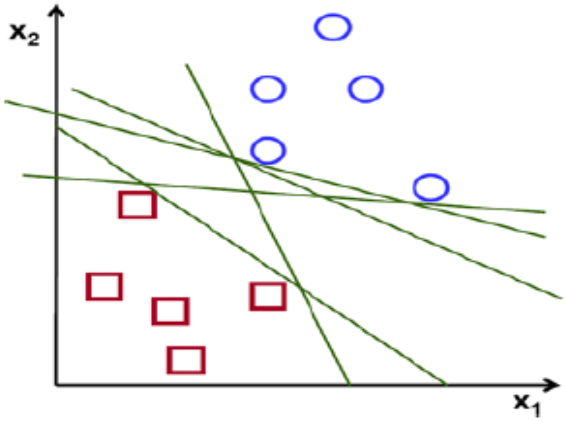


Figure 4.10 Support vector machine classification.

Therefore, it is essential for the classifier to find the optimal hyperplane in order to categorize categorise data items as good as possible. The optimal line, plane or hyperplane is that which separates categories in an optimal way and also has the biggest distance between the training examples

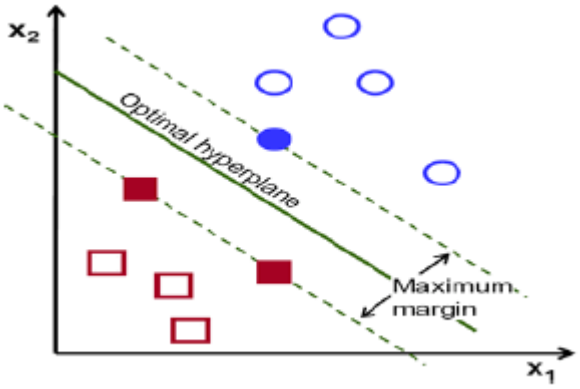


Figure 4.21 Optimal line

SVM has several parameters which might affect the accuracy of our classifier. Some of the most important parameters are this kernel, gamma and C.

- Kernels are the functions which transform a low dimensional input space into a higher dimensional space in order to transform a non-separable problem to a separable problem. These kernels functions are separated into 3 categories: linear, rbf and polynomial.

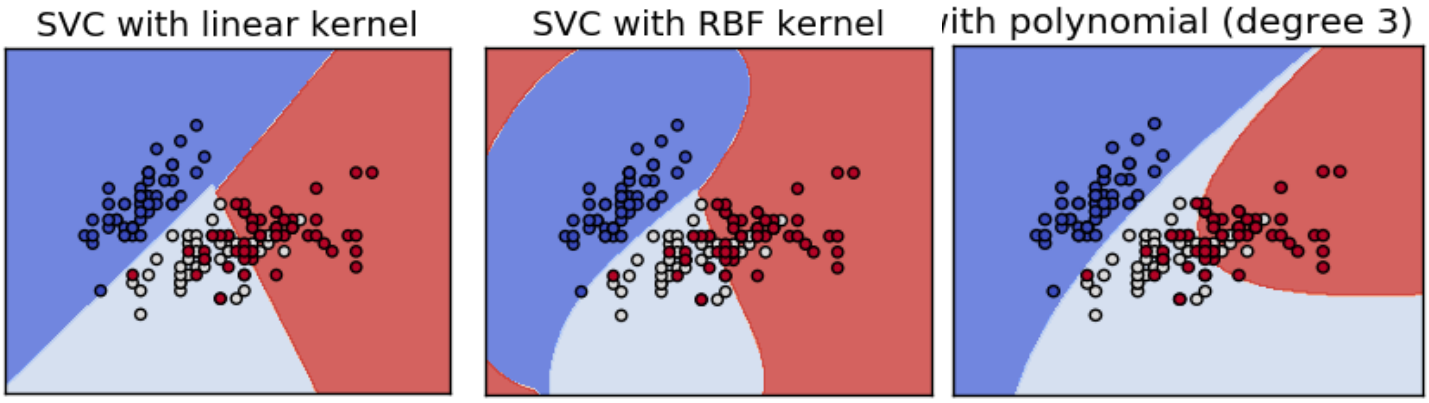


Figure 4.22 Categorization with 3 different kernel function,

http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html

- Gamma: Higher the value of gamma, the more precise fit will applied on training data. A high value of gamma may cause generalization problem and overfitting.
- C: This parameter controls the trade-off between smooth decision boundaries between different categories.

This kind of classifiers is more likely to be chosen due to the fact that they works very well with clear margins of separation, also they are very effective in high dimensional spaces and they are memory sufficient because they use a subset of training set.

Nevertheless, SVM is not perfect, it requires high training time with large data and doesn't perform well if there is noise in the data.

These classifiers such as SVM have proved extremely successful at binary classification tasks. However, to carry out multiclass classification tasks methods like One-v-All or All-v-All need to be used.

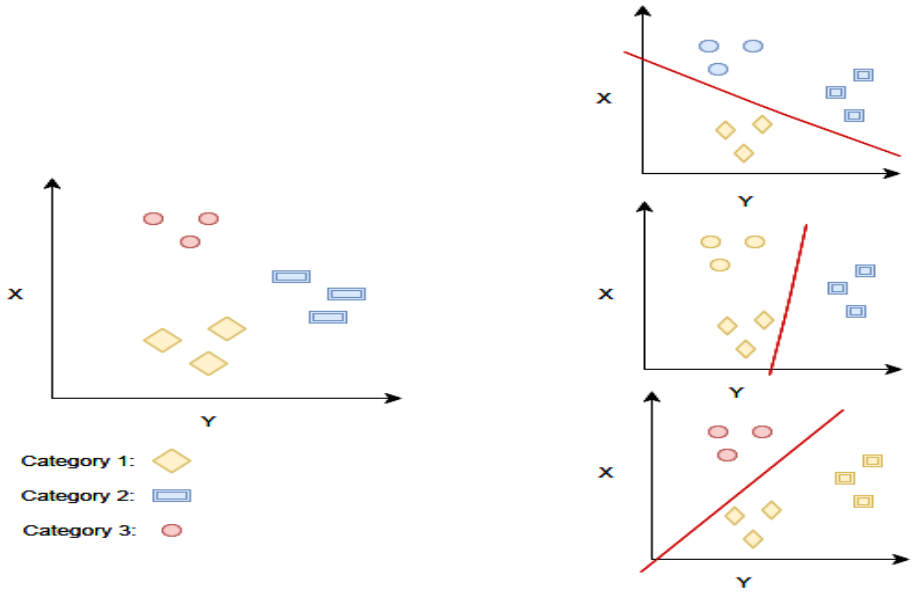


Figure 4.23 One vs All scheme for multiclass classification.

4.1.6.3.1 One vs All

This is one of the simplest multiclass classification schemes. It is implemented by training N different classifiers one for each category, classifier trained to distinguish the training data items of a category from the data items of all the remaining categories

[13]. Thus, when a new data item is needed to be classified, then all the classifiers run in order to generate a value which displays the likelihood of an input to belong in a particular category, then .

4.1.6.4 Decision Trees

Decision trees are also a very accurate supervised machine learning algorithms. A decision tree is a simple flowchart that chooses labels for input values. Decision tree comprised of 3 objects:

- Root node is the head of the decision tree.
- Decision nodes are check feature values to guide the data item to the appropriate leaf node.

- Leaf nodes are responsible to assign labels.

To choose a category we start at first from the root node, this node chooses which branch is the appropriate for this input by its input value's feature. Following that branch we arrive to a new decision node, that node will guide the input to the appropriate branch and so forth. This procedure continues until we reach a leaf node which provide a label for this particular input.

In this project we will focus on a method that uses decision trees to estimate new values. That method is called Random Forest.

This method grows many decision trees classifiers. In order to generate a result in passes the new input through every decision tree. Each tree provides its 'vote' for a category for a particular input. Forest chooses the category with the most votes.

Random Forest is ideal for classification tasks because it runs efficiently on large data bases, it can estimate that variables are important for the in the classification. It has methods to balance the error and also has proved that can be very accurate.

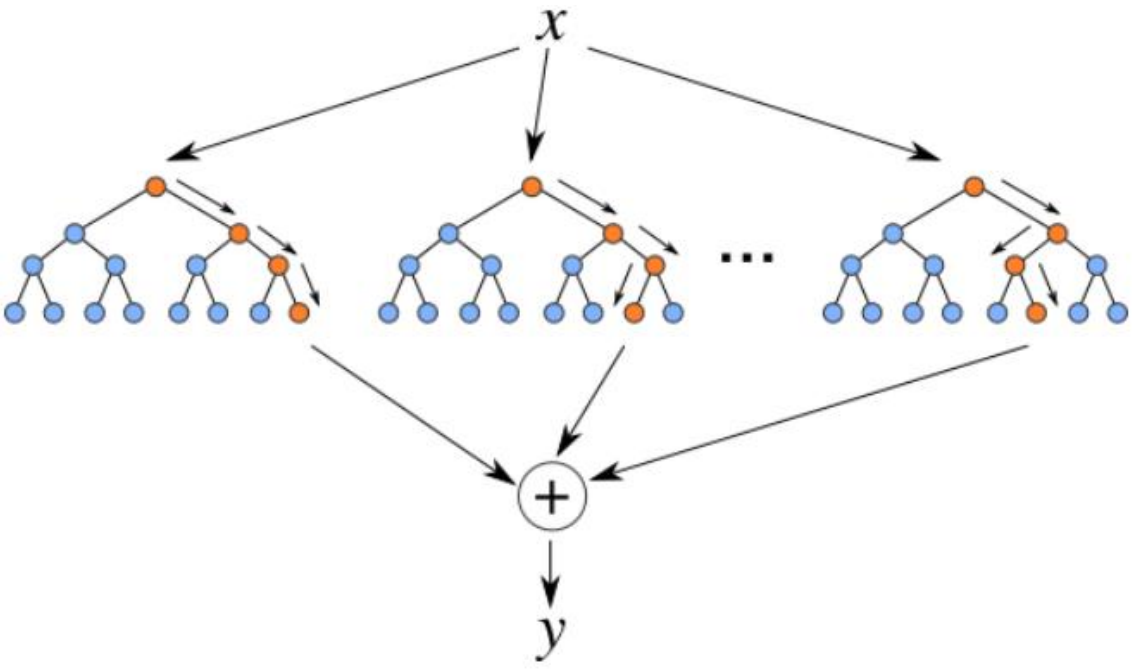


Figure 4.24 Random Forest Algorithm

4.1.7 Evaluation

This step is focussed on the interpreting of the mined patterns, after this step there is a possibility returning to any other steps for changes and improvements. Furthermore, this is the step where the visualizing phase occurs in order to display and interpret our patterns.

4.1.7.1 Performance Measures

To measure the predictive ability of classifiers we used a measurement called accuracy. Also cross validation techniques applied to increase the precision of measurement.

Accuracy:

The accuracy is the measurement where the number of correct predictions made divided by the total number of prediction made.

$$accuracy = \frac{correct\ predictions}{predictions\ made}$$

Cross validation - K-fold cross:

In order to measure the performance of our classifiers a technique called k-fold cross validation applied. This technique helped us to measure the performance of our classifiers more precisely. The main steps of this technique are as follows:

1. Randomly split your entire dataset into k"folds".
2. For each k folds in your dataset, build your model on k – 1 folds of the data set. Then, test the model to check the effectiveness for kth fold.
3. Record the error you see on each of the predictions.
4. Repeat this until each of the k folds has served as the test set.
5. The average of your k recorded errors is called the cross-validation error and will serve as your performance metric for the model.

Below is the visualization of how does a k-fold validation work for k=5.

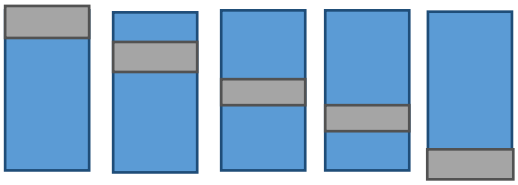


Figure 4.25 k-fold technique these rectangles represents our dataset, for each iteration the blue part is used for training and the grey part for testing.

4.1.6 Knowledge representation

In this stage, the mined knowledge are represented to people. That includes schemes for representing knowledge [17].

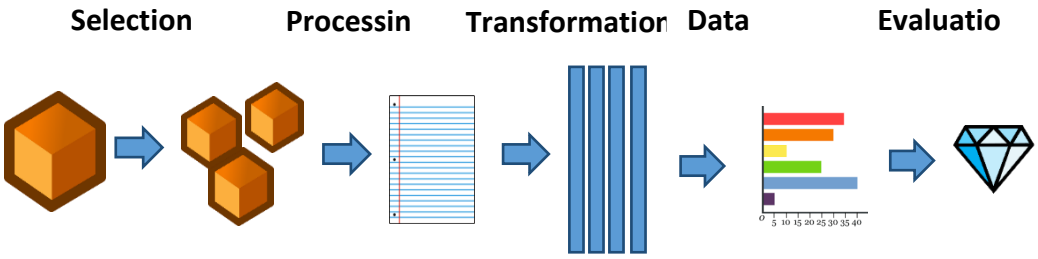


Figure 4.26 Knowledge discovery process

4.2 Python

Python used for our implementation. Python is a high level programming language, It is ideal for scripting and rapid application development in various fields due to its dynamic typing. Moreover, it becomes one of the most preferred languages for machine learning as well as for data science. The main advantage is a broad community support and comprehensive set of libraries. Thus, python provides you the capability to use these libraries in order to run some complex source code such as machine learning algorithms. In addition, python it is powerful and simple due to its simple syntax and its huge community. The main libraries for machine learning are as follows:

- Scikit-learn: It is the most popular machine learning library of python due to its huge number of features for data mining and data analytics. It is lower level than

other related libraries and acts as the foundation for Machine learning implementations.

- Pandas: is a library written for Python for data manipulation and analysis. It provides easy-to-use data structures and data analytics tools.
- NLTK: this is a library written in Python to cope with human language data. It provides interfaces to over 50 corpora and lexical resources. It provides many essential functions for text processing such as tokenization and stemming.
- Matplotlib: is a Python plotting library which generates quality figures. It can generate plots, histograms, bar charts etc.
- Tweepy: is an open source library which provides easy access to the Twitter API for Python.

Chapter 5

Experiment & Results

5.1 Test Bed

We used a virtual machine using VMware to run those experiments. The operating system was Ubuntu 14.04 LTS 32bits, Processor Intel Core i7-3632QM CPU 2.20GHz and a 40 GB Disk.

5.2 Test Data

For these experiments we used a dataset that comprised of 305,896 tweets. In this dataset there are 125,124 tweets that refer to live events, 96,734 tweets refer to news

related topics, 71,934 tweets refer to group interest topics and 12,075 tweets refer to commemorative topics. Moreover, this dataset comprise of 785 unique topics. Therefore, due to the fact that we using k-fold techniques(k=5) in order to increase our metrics precision, that means that our training set comprise of 244.717 tweets while our test set contains 61,179 tweets. In addition, it is important for a valid measurement, the tweets that refer to the same topic to be as a whole either in the training or in the test set.

5.3 Text-Based Classifier

For this approach we use only textual content to discriminate tweets within our categories. In this experiment, we categorize each tweet individually into the category that is most fitted. To cope with this task two classifiers are used the Multinomial Naïve Bayes and the Stochastic Gradient descent.

Each record (row) in the dataset that we used for this experiment comprised of three fields which are:

- The Category of tweet
- The Topic of tweet
- The textual content of the tweet.

5.3.1 Experiment setup

For this experiment, we rely on a bag-of-word approach in order to operate a classification task. Thus, to help our algorithm to classify tweets more efficiently a sequence of steps is required as a pre-processing phase.

- Remove stop word which are commonly occurring words such as “this”,”that”,”on” etc.
- Remove punctuation
- Remove URLS and user mentions
- Stemming which is the procedure to convert a word to its stem or basic word forms.

- Tokenize the text content into individual words.
- Bag-Of-Words is the representation of a document as a bag of words. This technique forms vectors to represent the count of each word.
- TF-IDF vector to obtain the frequent terms, tf-idf stands for “term frequent/inverse document frequency” and by using this method we emphasize on words that occurs frequently in a given category while simultaneously de-emphasize words that occur frequently in many categories.

The process of this approach can be visualized a follows:

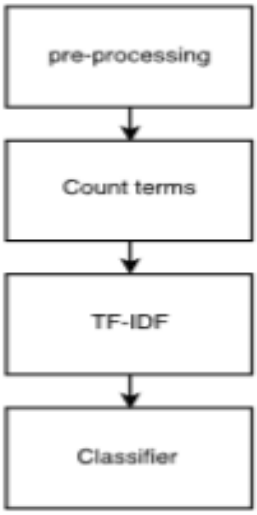


Figure 5.1 steps of this text-based approach

In this experiment, our classifiers take as input the term frequency values of our vocabulary. Considering this knowledge classifiers have to predict the category of a new tweet.

5.3.2 Accuracy

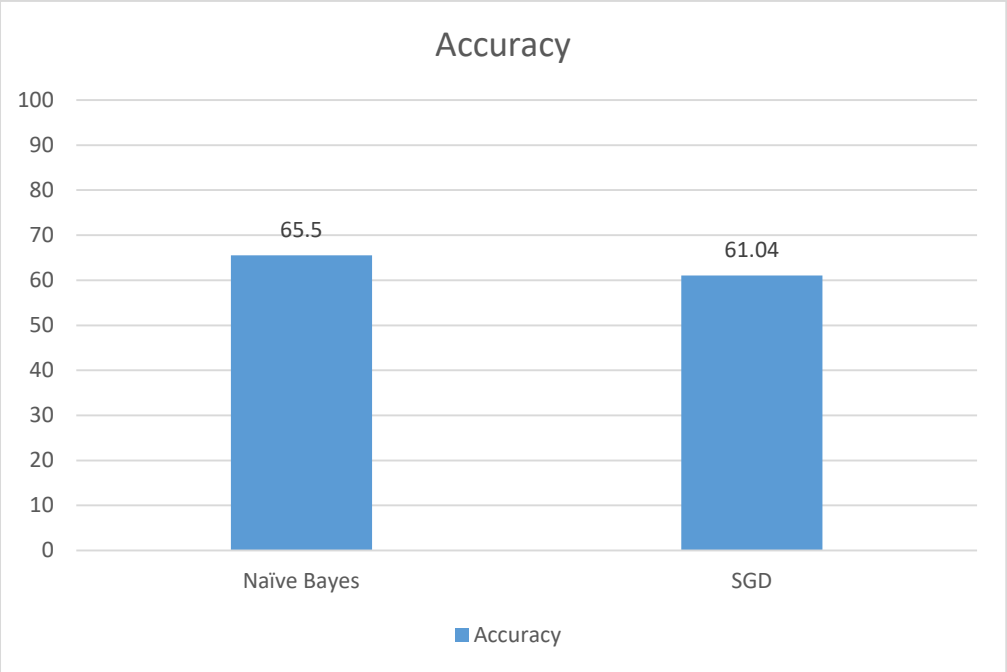


Figure 5.2 Accuracy metrics for text-based approach

5.4 Feature-based approach

For this approach, we use only features that we constructed in data transformation step in order to categorize topics as a whole instead of tweets. Thus, we created a dataset which contained the topics and its feature values for each topic. This approach let us experiment with classification models and algorithms that we couldn't use for text. For this experiment two classification methods used, the Random Forest and the Support Vector Machine model.

For this experiment, a dataset which contains 18 fields was used. We used the features that we analysed above. These features are Negative sentiment, Positive sentiment, Neutral sentiment, Compound, Length, Examinations, Questions, Word diversity, Language diversity, Links, Hashtags ratio, Retweets depth, Retweets ratio, Hashtags diversity and Topic repetition.

5.4.1 Experiment setup

The first algorithm that we used was SVM which have proved extremely successful at binary classification tasks. However, as a multiclass classification task, we had to use a multiclass classification method, so we relied on the one-vs-all scheme which explained above. Thus, in this method, we use records with 16 dimensions and the classifier is called to discover the optimum plane to discriminate these topics. SVM kernel was set to rbf.

Moreover, we also used random forest method to classify topics. We set the forest to comprise of 350 estimators.

The only prepossessing method that we used was normalization.

5.4.2 Accuracy

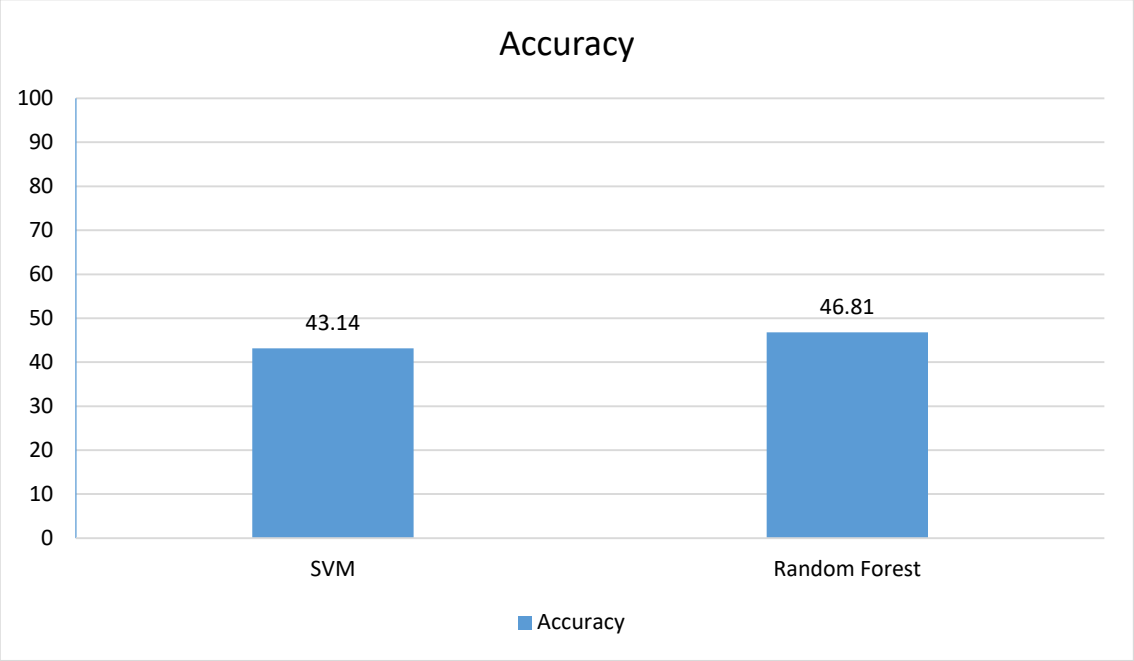


Figure 5.2 Accuracy metrics for feature-based approach

5.5 Hybrid Classification Model

For this approach, we used both textual content and numeric values that are easily extracted by a single tweet. This experiment categorizes each tweet individually into the category that fitted most taken into account both text and features. To do so, we used the same algorithms as in the first approach, Naïve Bayes, and SGD.

For this task, our dataset comprised of 11 fields was used which are the topic, the category, the text, the number of hashtags, the number of URLs, the number of exclamations and question marks and the four values that we obtained from the sentiment analysis (positive, negative, neutral, compound).

5.5.1 Experiment setup

The issue in this approach is that our data consist of heterogeneous data types so it needs different processing and feature extraction. Thus, to combine this knowledge first we have to process our data using the appropriate steps. To cope this problem we processed textual and numerical values separately. The same pre-processing as the in the first approach will be applied for text values which mean that a frequent vector will be generated. Then, we merge those vectors with the feature values vectors in order to create an input for our classifier that contains both textual and numeric values

The process of this approach can be visualized as follows:

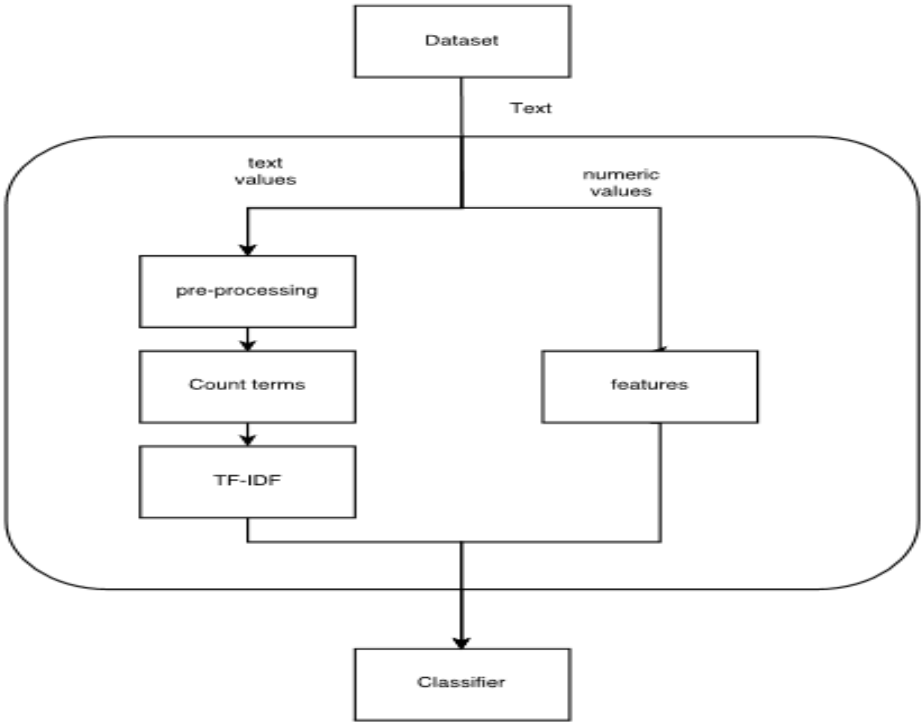


Figure 5.4 Hybrid approach procedure

5.5.2 Accuracy

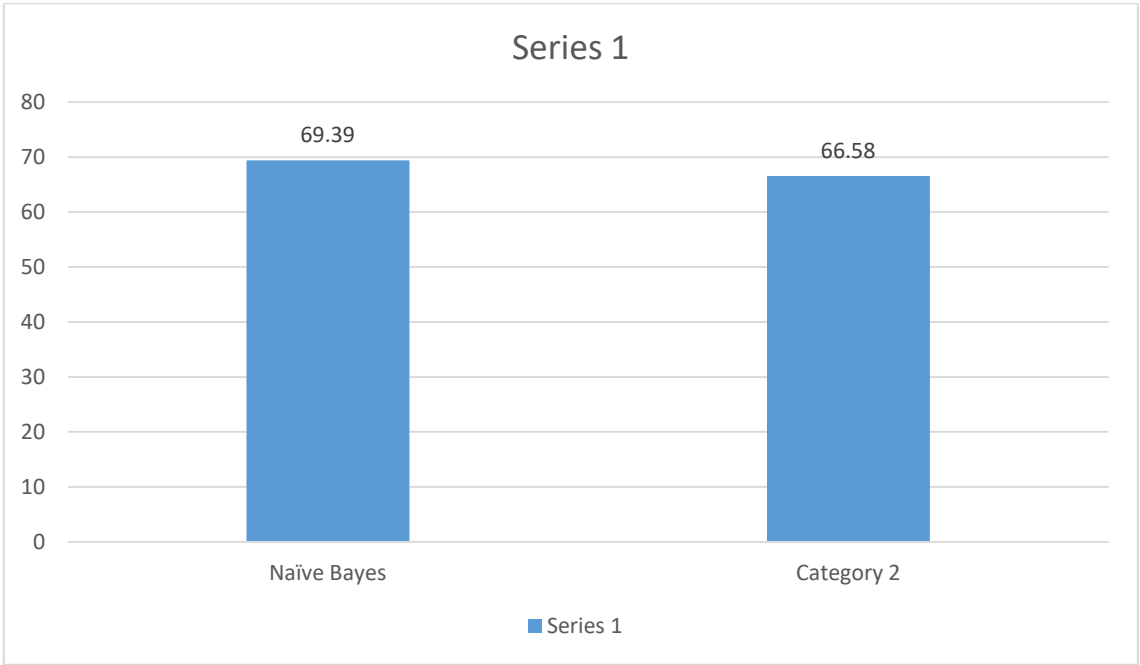


Figure 5.5 Accuracy for Hybrid approach

5.6 Method comparison

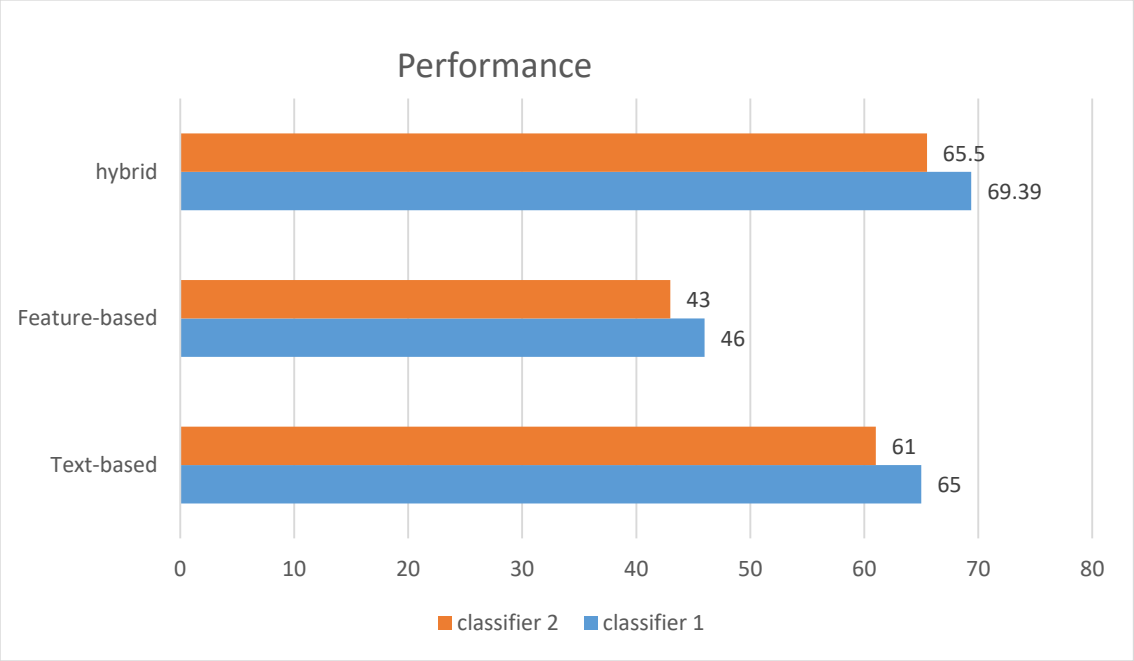


Figure 5.6 Comparison between approaches

Two classifiers used for each approach. This chart illustrates that the classifiers that took as input the most knowledge have returned the best results. Furthermore, text values seemed to be the most important due to the fact that using only numerical features have resulted in the worst accuracy. Specifically, the feature-based approach had the worst accuracy. SVM and Random forest classifiers used and provide an accuracy of 43% and 46% respectively. Almost, 20% more accurate was the text-based approach, Naïve Bayes gave an accuracy of 65% while SGD reached 61%. The most accurate approach was the hybrid, where we used Naïve Bayes and SGD again in order to observe if there was any improvement. SGD reached 65.5% which is 4.5% higher than the previous observed value and Naïve Bayes reached 69.39% which was the highest accuracy that we observed.

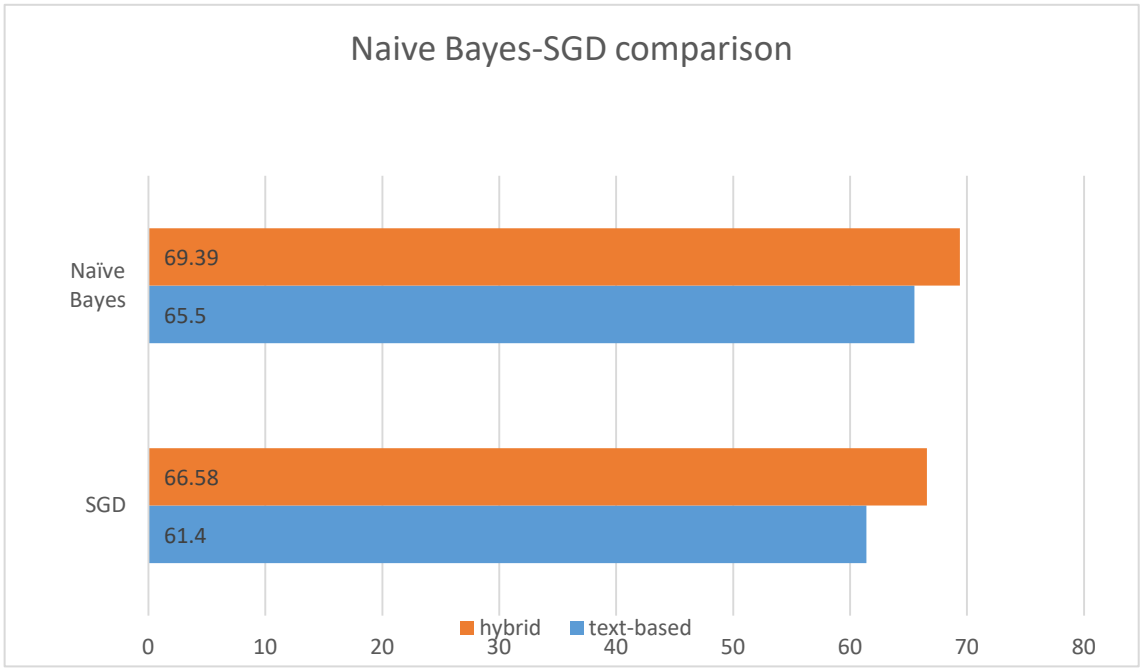


Figure 5.7 Comparison between first and third approaches

This chart shows the improvement of these classifiers by giving them extra knowledge for the tweets. Naïve Bayes improved by 3.7% and SGD by 5.18 reaching a proportion of 69.39% and 66.58% respectively.

Chapter 6

Conclusion

6.1 Conclusion

This research attempted to classify topics which gathered from social media in order to improve convenience not only for users but also for government and companies to benefit from the real-time knowledge. The main aim was to prove that extra knowledge as sentiment or other features can be beneficial for a classification task. We classify tweets by the type of the motivation that caused it using a set of categories that consist of 4 different types: news, live events, group interest, and commemoratives.

Moreover, we used three different approaches to classifying twitter data. The first approach was a text-based classification using Naive Bayes and SGD algorithm which reach an accuracy of 65.3% and 61% respectively, the second was a features-based approach using generated features to classify topics into the appropriate categories, using Random Forest and SVM algorithms which gave an accuracy of 46% and 43% respectively. The last approach was a hybrid approach where we combined both text and sentiment ratings in order to achieve a more accurate classification. For the last experiment, we used the same algorithms as in the first approach in order to observe the improvement of classifiers accuracy. SGD improved by 5.5% and reached at 66.58% while Naïve Bayes reached a percentage of 69.39%.

This experiment led us to the conclusion that sentiment and other features can provide extra useful knowledge to discriminate tweets into categories.

6.2 Future Work

Regarding future work, we strongly believe that even better results can be obtained using these general categories as long as a better sentiment analysis applied. Our sentiment analysis was taking into consideration only how positive or negative was a tweet. However, a huge number of tweets was set to neutral despite the fact that, it contained sentiment characteristic. For instance, there were a large number of tweets in live events category which contained excitement, although it has been categorized as neutral. On the other hand, in news related topics a large number of tweets contained sarcasm. Thus, with a more efficient sentiment analysis better results can be obtained.

Moreover, this is a very general categorization. However, it can be easily be extended by breaking the categories into subcategories. For example, subcategories for the news category can be politics, natural disasters and economy while for live events can be football and the tv shows.

References

1. E. Geanina, Perspectives on Big Data and Big Data Analytics, Database Systems Journal vol. III, 4/2012, http://www.dbjournal.ro/archive/10/10_1.pdf
2. C. Scarioni, Hadoop Basics—Creating a MapReduce Program, https://dzone.com/articles/hadoop-basics-creating
3. J. Dittrich, J. Quiane-Ruiz, Efficient Big Data Processing in Hadoop MapReduce, http://dl.acm.org/citation.cfm?id=2367562
4. S. Yang, A. Kavanaugh, Half-Day Tutorial: Collecting, Analyzing and VisualizingTweets using Open Source Tools http://dl.acm.org/citation.cfm?id=2037633
5. S. Asur, B. Huberman, G Szabo, C. Wang, Trends in Social Media : Persistence and Decay, http://www.hpl.hp.com/research/scl/papers/trends/trends_web.pdf
6. S. Bird, E. Klein, E. Loper, Learning to Classify Text, 7/2015, http://www.nltk.org/book/ch06.html
7. Dr. S Vijayatani, Ms. J. Ilamathi, Ms. Nithya, Pre-processing Techniques for Text Mining – An Overview, 7/2016, http://www.ijcscn.com/Documents/Volumes/vol5issue1/ijcscn2015050102.pdf
8. A. Taspinar, Text Classification and Sentiment Analysis, 11/2015, http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/
9. A. Java, X. Song, T. Finin, B. Tseng, Why We Twitter: Understanding Microblogging Usage and Communities , http://ebiquity.umbc.edu/_file_directory_/papers/369.pdf
10. E. Kouloumpis, T. Wilson, J. Moore, Twitter Sentiment Analysis: The Good the Bad and the OMG! http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2857/3251?height%3D90%%26iframe%3Dtrue%26width%3D90%
11. A. Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment Analysis of Twitter Data, http://dl.acm.org/citation.cfm?id=2021114
12. V. Korde, Text Classification and Classifier: A Survey, International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.2, March 2012, http://aircconline.com/ijaia/V3N2/3212ijaia08.pdf
13. R. Rifkin, A Klautau, In Defence of One-Vs-All Classification, Journal of Machine Learning Research 5 (2004) 101-141, http://www.jmlr.org/papers/volume5/rifkin04a/rifkin04a.pdf

14. K. Lee, D. Palsetia, R. Narayanan, Md. M. Ali Patwary, A. Agrawal, and A., Twitter Trending Topic Classification ,Choudhary, 2011 11th IEEE International Conference on Data Mining Workshops, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6137387
15. B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, Short Text Classification in Twitter to Improve Information Filtering, http://dl.acm.org/citation.cfm?id=1835643
16. A. Zubiaga, D. Spina, R. Martinez, V. Fresno, Real-Time Classification of Twitter Trends, https://arxiv.org/pdf/1403.1451.pdf
17. *U. Fayyad,, G.Shapiro, P. Smyth,* From Data Mining to Knowledge Discovery in Databases, https://www.csd.uwo.ca/faculty/ling/cs435/fayyad.pdf
18. Naïve Bayes, http://scikit-learn.org/stable/modules/naive_bayes.html
19. D. Isa, L.Hong Lee, V.P. Kallimani, and R. RajKumar, Text Document Pre-processing with the Bayes Formula for Classification Using the Support Vector Machine, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 20, NO. 9, SEPTEMBER 2008 http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4492780
20. C.J. Hutto, E. Gilbert, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

Appendix A

Collecting Procedure

```
#Import the necessary methods from tweepy library
from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import time
import tweepy
import json
import time
import MySQLdb

#Variables that contains the user credentials to access Twitter API
access_token = "Enter access_token"
access_token_secret = "Enter access_token_secret"
consumer_key = "Enter consumer_key"
consumer_secret = "Enter consumer_sercet"



#This is a basic listener that just prints received tweets to stdout.
class StdOutListener(StreamListener):

    def on_data(self, data):
        print (data)
        return True

  def on_error(self, status):
      print (status)


if __name__ == '__main__':

    conn = MySQLdb.connect(host= "localhost",user="root",passwd="px2h-r7s",db="twitter_data")
    x = conn.cursor()
    trendlistNew=[]
    trendlist=[]
    print(len(trendlist))
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)
    stream = Stream(auth, l)
    api = tweepy.API(auth, wait_on_rate_limit=True,wait_on_rate_limit_notify=False)

    while 1 :
        trends1 = api.trends_place(id=44418)
        data = trends1[0]

        trends = data['trends']
        hashtags=0
        urls=0
        retweeted=0
        names = [trend['name'] for trend in trends]
        trendlistNew=[trend for trend in names if trend not in trendlist]
        print trendlistNew
        for topic in trendlistNew:
                try:
                        first=True
                        #new_tweets=json.dumps(new_tweets)
                        for tweet in  tweepy.Cursor(api.search,q=topic,rpp=400,result_type="recent",include_entities=True).items(400):
                                #if tweet["lang"]=='en' or tweet["lang"]=='gr' :
                                try:
                                        userId= tweet.user.id
```

```
                tweetId=tweet.id
                text = tweet.text.encode('utf-8')
                text = text.replace('\n', ' ').replace('\r',' ')
                text = text.replace('\t', ' ').replace('\r',' ')
                text = text.replace('"', ' ').replace('\r',' ')
                for hashtag in tweet.entities["hashtags"]:
                        hashtags += 1
                for link in tweet.entities["urls"] :
                        urls +=1
                try:
                        tweet.retweeted_status
                except AttributeError:
                        retweeted=0
                else :
                        retweeted=1
                lang = tweet.lang
                timestamp = tweet.created_at
                timestamp=str(timestamp).split(' ')

                if first :
                        first=False
                        try:
                                x.execute("INSERT INTO `topics` (`name`) VALUES (\""+str(topic)+"\")")
                                conn.commit()
                                print "succees on topic"
                        except:
                                print "error on topic"
                                print conn.rollback()
                try:
                        x.execute("INSERT INTO `tweets` (`id`, `tweet_id`, `user_id`, `tweet_text`, `hashtags_cou
                        conn.commit()
                        print "succees on tweet"

                                print "succees on topic"
                        except:
                                print "error on topic"
                                print conn.rollback()
                try:
                        x.execute("INSERT INTO `tweets` (`id`, `tweet_id`, `user_id`, `tweet_text`, `hashtags_cou
                        conn.commit()
                        print "succees on tweet"
                except:
                        print "error on tweet"
                        print conn.rollback()
                hashtags=0
                urls=0
                retweeted=0
        except ValueError:
                print "error"
                continue
except ValueError:
        continue

trendlist += trendlistNew
trendlistNew=[]
time.sleep(60*15)
```

Appendix B

Data visualization

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from matplotlib import pyplot as plt2
from collections import Counter
import collections


csv_file=pd.read_csv("dataset.csv",sep='\t')
print len(csv_file)
c=Counter(csv_file["lang"]).most_common(6)
x_axis=[]
y_axis=[]
for val in c:
        x_axis.append(val[0])
        y_axis.append(val[1])
fig,ax = plt.subplots()
index=np.arange(len(x_axis))
bar_width=0.7
opacity=0.8
error_config={'ecolor':'0.3'}
recults=plt.bar(index,y_axis,bar_width,alpha=opacity,color='r',error_kw=error_config)
plt.ylabel('value')
plt.xlabel('lang')
plt.xticks(index+0.4/1.2,x_axis)
plt.legend()
plt.title('6 most common languages')
plt.tight_layout()

#dates
x_axis_date=[]
y_axis_date=[]
dates=Counter(csv_file["date"]).most_common(7)
print dates
for val in dates:
        x_axis_date.append(val[0])
        y_axis_date.append(val[1])
#print x_axis_date , y_axis_date
fig,ax = plt.subplots()
index=np.arange(len(x_axis_date))
bar_width=0.5
opacity=0.8
error_config={'ecolor':'0.3'}
recults=plt.bar(index,y_axis_date,bar_width,alpha=opacity,color='b',error_kw=error_config)
plt.xlabel('days')
plt.ylabel('values')
plt.title('tweets per day')
plt.xticks(index+0.3/1.2,x_axis_date)
plt.legend()
plt.tight_layout()


#categories
x_axis_cat=[]
y_axis_cat=[]
categories=[]
categories=list(collections.Counter(csv_file["category"]).items())
print categories
for val in categories:
        x_axis_cat.append(val[0])
        y_axis_cat.append(val[1])
#print y_axis_cat
#print x_axis_cat
fig,ax = plt.subplots()
index=np.arange(len(x_axis_cat))
opacity=0.8
error_config={'ecolor':'0.3'}
recults=plt.bar(index,y_axis_cat,bar_width,alpha=opacity,color='g',error_kw=error_config)
plt.xlabel('categories')
plt.ylabel('capacity')
plt.title('tweets per category')
plt.xticks(index+0.3/1.2,x_axis_cat)
plt.legend()
plt.tight_layout()
plt.show()
```

2.

```
import numpy as np
import matplotlib.pyplot as plt
from pandas import DataFrame
from sklearn.preprocessing import MinMaxScaler
from sklearn.preprocessing import Normalizer
def build_data_frame(file_name):
        with open(file_name,'rb') as tweets:
                firstline=True
                topics=[]
                rows=[]
                for topic in tweets:
                        if firstline:
                                firstline=False
                        else:
                                topic=topic.strip().split('\t')
                                features=topic[2:]
                                if topic[1] == 'live events':
                                        rows.append({"depth":features[0],"ratio":features[1],"hashtags":features[2],"length":features[3],"ekla":fe
                                elif topic[1] == 'group interest':
                                        rows.append({"depth":features[0],"ratio":features[1],"hashtags":features[2],"length":features[3],"ekla":fe
                                elif topic[1] == 'news':
                                        rows.append({"depth":features[0],"ratio":features[1],"hashtags":features[2],"length":features[3],"ekla":fe
                                elif topic[1] == 'commemoratives':
                                        rows.append({"depth":features[0],"ratio":features[1],"hashtags":features[2],"length":features[3],"ekla":fe

                                topics.append(topic[0])
                dataframe = DataFrame(rows,index=topics)
        return dataframe,topics

if __name__=='__main__':

        fs = 10  # fontsize

        data1,topics=build_data_frame('features.csv')
        labels = list('LGNC')
        #scaler = MinMaxScaler(feature_range=(0,10))

        scaler = Normalizer()
        data=scaler.transform(data1.astype(float))
        data = DataFrame(data1)
        data.columns = list(data1.columns.values)
        print data
        # demonstrate how to toggle the display of different elements:
        live_data1 = data.loc[data['category'] == 0]
        group_data1 = data.loc[data['category'] == 1]
        news_data1 = data.loc[data['category'] == 2]
        com_data1 = data.loc[data['category'] == 3]

        #l_data=live_data1["senti"]
        #g_data=group_data1["senti"]
        #n_data=news_data1["senti"]
        #c_data=com_data1["senti"]

        #normalize data
        #live_data1 = scaler.fit_transform(live_data1.astype(float))
        #group_data1 = scaler.fit_transform(group_data1.astype(float))
        #news_data1 = scaler.fit_transform(news_data1.astype(float))
        #com_data1 = scaler.fit_transform(com_data1.astype(float))

        live_data = DataFrame(live_data1)
        live_data.columns = list(data.columns.values)

        group_data = DataFrame(group_data1)
        group_data.columns = list(data.columns.values)

        news_data = DataFrame(news_data1)
        news_data.columns = list(data.columns.values)


        com_data = DataFrame(com_data1)
        com_data.columns = list(data.columns.values)

        for feature in group_data[1:]:
                fig = plt.figure()
                ax = fig.add_subplot(111)
                ax.boxplot([live_data[feature].astype(float),group_data[feature].astype(float),news_data[feature].astype(float),com_data[feature].a
                ax.set_title(feature, fontsize=fs)

        plt.show()
```

Appendix C

Classifiers

Naive Bayes

```
from pandas import DataFrame
import pandas as pd
import re
import numpy
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.cross_validation import KFold
from sklearn.metrics import accuracy_score
from sklearn.utils import shuffle
from nltk.stem.porter import PorterStemmer

#Create a dataframe which in the first column contains the text and in the second the category
def build_data_frame(file_name):
        stemmer=PorterStemmer()

        with open(file_name,'rb') as tweets:
                firstline=True
                firsttopic=True
                prevtopic=None
                flag=True
                topics1=[]
                topics2=[]
                rows1=[]
                rows2=[]
                line1=[]
                for tweet in tweets:
                        if firstline:
                                firstline=False
                        else:
                                fields = tweet.strip().split('\t')
```

```
                        if firsttopic:
                                prevtopic=fields[14]
                                firsttopic=False
                        line = fields[5]
                        line = line.replace("rt", "")
                        line = line.replace("RT", "")
                        line = re.sub(r"http\S+", "", line)
                        line = re.sub(r"([!.'():,?%]+)", " ", line)
                        line = re.sub(r"https\S+", "", line)
                        line = re.sub(r"(@[a-zA-Z0-9_]+)", "", line)
                        line = re.sub(r"([0-9]+)", "", line)
                        line = line.lower()
                        line = re.sub(r"(.)\1{1,}", r"\1\1", line)
                        line = line.strip()

                        if prevtopic != fields[14] and flag:
                                flag=False
                                prevtopic=fields[14]
                        elif prevtopic != fields[14] and not flag:
                                flag=True
                                prevtopic=fields[14]
                        if flag :
                                rows2.append({"text":line,"category":fields[1]})
                                topics2.append(fields[14])
                        else :
                                rows1.append({"text":line,"category":fields[1]})
                                topics1.append(fields[14])
        dataframe1 = DataFrame(rows1,index=topics1)
        dataframe2 = DataFrame(rows2,index=topics2)
return pd.concat([dataframe2,dataframe1])
```

```
if __name__=='__main__':
        data=build_data_frame('bb.csv')

        #data=data.groupby(['topics'],as_index=False).sum()
        #data = shuffle(data)
        print data
        #data=data.sample(frac=1) #shuffle dataset
        pipeline = Pipeline([   ('vectorizer',CountVectorizer()),
                                ('tfidf', TfidfTransformer()),
                                ('classifier',MultinomialNB())
                            ])
        #Cross-Validating
        kfold = KFold(n=len(data),n_folds=6)
        scores = []
        for train_ind,test_ind in kfold:
                train_text = data.iloc[train_ind]['text'].values
                train_y= data.iloc[train_ind]['category'].values

                test_text = data.iloc[test_ind]['text'].values
                test_y= data.iloc[test_ind]['category'].values

                pipeline.fit(train_text,train_y)
                predictions = pipeline.predict(test_text)

                score = accuracy_score(test_y, predictions)
                print score
                scores.append(score)

print ("Score: {:.2f}".format(sum(scores[])/len(scores[])*100))
```

SGD

```
from __future__ import print_function
import numpy as np
import re

from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import HashingVectorizer
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.calibration import CalibratedClassifierCV
from sklearn.linear_model import SGDClassifier

from nltk.stem.porter import PorterStemmer

##############################################################################

X_train = []
y_train = []
target_names = ['live events', 'group interest', 'news', 'commemoratives']


with open('bb.csv','rb') as tweets:
    firstline=True
    stemmer=PorterStemmer()
    for tweet in tweets:
        line1=[]

        if firstline:
                firstline=False
```

```
        else:
                fields = tweet.strip().split('\t')
                text = fields[5]
                text = text.lower()
                text = re.sub(r"rt", "", text)
                text = re.sub(r"http\S+", "", text)
                text = re.sub(r"([!.'():,?]+)", " ", text)
                text = re.sub(r"https\S+", "", text)
                text = re.sub(r"(@[a-zA-Z0-9_]+)", "", text)
                text = re.sub(r"([0-9]+)", "", text)
                text = re.sub(r"(.)\1{1,}", r"\1\1", text)

                if fields[1] == target_names[0]:
                        y_train.append(0)
                elif fields[1] == target_names[1]:
                        y_train.append(1)
                elif fields[1] == target_names[2]:
                        y_train.append(2)
                elif fields[1] == target_names[3]:
                        y_train.append(3)
                X_train.append(text)



clf = SGDClassifier(n_jobs = -1, n_iter = 100, eta0=0.1)


pipeline = Pipeline([
    ('vectorizer', HashingVectorizer(non_negative=True,n_features=(2 ** 18))),
    ('clf', CalibratedClassifierCV(base_estimator=clf, cv=5, method='isotonic'))
])

scores = cross_val_score(pipeline, X_train, y_train, cv=5, n_jobs=-1)
print("score: {:.2f}%".format(sum(scores)/len(scores)*100))
print(scores)
```

SVM/Random Forest

```
def build_data_frame1(file_name):
        with open(file_name,'rb') as tweets:
                firstline=True
                firsttopic=True
                prevtopic=None
                flag=True
                topics=[]
                rows=[]
                for topic in tweets:
                        print topic
                        if firstline:
                                firstline=False
                        else:
                                topic=topic.strip().split('\t')
                                features=topic[2:]
                                if topic[1] == 'live events':
                                        rows.append({"features":features,"category":1})
                                elif topic[1] == 'group interest':
                                        rows.append({"features":features,"category":2})
                                elif topic[1] == 'news':
                                        rows.append({"features":features,"category":3})
                                elif topic[1] == 'commemoratives':
                                        rows.append({"features":features,"category":4})
                                topics.append(topic[0])

                dataframe = DataFrame(rows,index=topics)
        return dataframe

if __name__=='__main__':

        data=build_data_frame1('features.csv')
        #data=data.sample(frac=1) #shuffle dataset
        feature = data['features'].values
        categories = data['category'].values
        #data=data.sample(frac=1) #shuffle dataset
        features=[]
```

```
count=1
for l in feature:
        l=[float(i) for i in l]
        features.append(l)
        count+=1
features=pd.DataFrame(features)
#normalize data


#feature selection
#test = SelectKBest(k=12)
#fit = test.fit(features, categories)
#features = fit.transform(features)
clf=OneVsRestClassifier(svm.SVC(kernel='linear',gamma=0.001,C=100,max_iter=-1))
#clf=OneVsRestClassifier(RandomForestClassifier(n_estimators=450,class_weight='balanced',min_samples_split=4))
scores=cross_val_score(clf,features,categories,cv=5)
print ("score: {:.2f}%".format(sum(scores)/len(scores)*100))
```