A Study on Text Mining on Twitter: Identifying Opinion and Detecting Different Forms of Speech Using Writing Patterns

by

Mondher Bouazizi

Dissertation

Submitted by Mondher Bouazizi

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Engineering (Ph.D.)

Supervisor: Prof. Tomoaki Otsuki, Ph.D.

Graduate School of Science and Technology
Keio University

August, 2019

Contents

List of Tables

List of Figures

A Study on Text Mining on Twitter: Identifying Opinion and Detecting Different Forms of Speech Using Writing Patterns

Mondher Bouazizi
bouazizi@keio.jp.
Keio University, 2019

Supervisor: Prof. Tomoaki Otsuki, Ph.D.
ohtsuki@ics.keio.ac.jp

Abstract

Over the last two decades, online user-generated content has been exponentially increasing. With its increase, a proportionally increasing interest has been attributed to this data from the research community. While several works have been targeting different types of user-generated media such as photos, videos and audio content, text has always attracted most of the attention for several reasons. To begin with, due to the unique properties of natural languages, the analysis of such data presents several challenges. Nevertheless, hitherto, average internet users still use text more than any other type of media to interact with one another.

The studies performed on online generated text cover a wide range of types of analysis. These include but are not restricted to the analysis of motivations of users to share information, the evaluation of interests in events, the identification of prominent users, etc. Sentiment analysis, in particular, presents nowadays a hot topic of research. Sentiment analysis, also known as opinion mining, refers to the automatic identification and aggregation of opinions of people towards specific topics by analyzing their online written texts and publications. Sentiment analysis has several applications, ranging from product analytics to market analysis and public opinion orientation towards events such as elections, etc. Nevertheless, it is a field that is yet to be explored, with several of its challenges are yet to be dealt with. Instances of these include fine-grained sentiment analysis, evolution of sentiments over time, aspect-based sentiment analysis, etc.

On a related context, over the last decade or so, the focus of sentiment analysis has shifted from review websites, such as movie reviews websites, or online shops such as amazon etc., towards social media and microblogging websites. This is because these (i.e., social media and microblogging websites) have become the top attraction of online users, and the most visited and consulted platforms on the internet today. Twitter, in particular, has attracted a lot of attention, due to the ease of access to its data and the nature of the relationships between its users. That being the case, in our work, our experiments will be mostly conducted on data collected from Twitter.

This dissertation explores several of the challenges of sentiment analysis on social media, notably fine-grained sentiment analysis and sarcasm detection.

Chapter 1 introduces the concept of sentiment analysis on social media, its applications and challenges. We present several of the existing work which dealt with this task. We focus mainly on works on Twitter. However, relevant works which were performed on other social media or online websites will be presented as well. This chapter also summarized the scope and contribution of this dissertation.

Chapter 2 tackles a common challenge that has always been difficult to perform, yet very important to enhance the performance of sentiment analysis systems, i.e. the identification of sarcasm on social media. We use machine learning and the concept of patterns to identify sarcastic statements on Twitter. We run our experiments on a data set of texts posted on Twitter (i.e., tweets) and compare the performance of our proposed method to that of some conventional works. We also show how the identification of such statements can enhance the performance of sentiment analysis.

Chapter 3 focuses on a different task: multi-class sentiment analysis. As yet, most of the core of research on this field has been interested in the binary and ternary classification of texts. These refer to the classification of texts into positive and negative, and into positive, negative and neutral, respectively. Instead of limiting ourselves to such a coarse-grained classification, we go into a further level of granularity and classify texts into multiple sentiments. We re-use the concept introduced in the previous chapter, i.e., patterns, to perform this task. Alongside, we introduce SENTA (SENTiment Analyzer); a tool we have built that allows to extract, out of a wide variety of features, ones that can be used for applications such as sentiment analysis or sarcasm detection, through an easy-to-use graphical user interface.

Chapter 4 discusses in more details the results obtained in the previous one, explains the limitations of the task of multi-class classification which make it inherently difficult, and in some extreme cases impossible and describes the relation between sentiments and how correlated ones can be with some others. This chapter also offers possible solutions to overcome the limitations of multi-class sentiment analysis.

Chapter 5 presents a substitution to multi-class classification, which we refer to as Sentiment Quantification. Sentiment quantification refers to the identification of multiple sentiments expressed in a text, and attributing different scores to them to reflect their importance and weight within that text. In our proposed approach we use patterns and special type of unigrams to attribute scores to different sentiments to rank them and identify which ones are present in a given text, and which are not.

Finally, Chapter 6 concludes this dissertation highlighting its key points and the contribution made within, and proposes possible venues for future research of the topic of sentiment analysis.

Acknowledgments

Undertaking this PhD has been a truly life-changing experience. It would not have been possible to do without the support, guidance and encouragement of many people.

First and foremost, I would like to express my deepest and sincere gratitude to my supervisor Prof. Tomoaki Ohtsuki for the continuous support and encouragement that he gave me. Without his patience, motivation and immense knowledge, this PhD would not have been achievable. His advice was crucial to undertake new research challenges and keep persevering even in hard times when results were hard to obtain. I am deeply grateful for all the empowerment I received under his supervision that let me define my own pace.

The committee members Dr. Tony Quek, Prof. Iwao Sasase and Prof. Masaaki Ikehara deserve a special mention, for their precious time and advice that helped improve the quality of this dissertation.

Assistance provided by Keio Leading Edge Laboratory (KLL) and NEC C&C, by offering grants to support Research during my master and PhD studies has been of a great help and deserves a special thank you.

Many Thanks to all the members of Ohtsuki Laboratory, especially Jihoon Hong, Juan Camilo Corena Bossa and Kentaro Toyoda who have always been there for me, have never let me work alone and have always known how to keep me enthusiastic and looking forward to face new challenges. Their unlimited energy is a reference for my future endeavors.

I would also like to say a heartfelt thank you to my dear friend Anthony, my brothers Hatem, Hichem and Radhouan and my mother Mdalla whose sacrifices, love and guidance have given me all the opportunities I enjoy to this day.

Last but not least, to the memory of my father, Abdelhafidh, who always believed in my ability to be successful in the academic arena. If it was not for your encouragement, undertaking a PhD would not have been a path I could have chosen all by myself. You are gone, but your belief in me has made this journey possible.

Mondher Bouazizi

Chapter 1

Introduction

This dissertation is concerned with two main topics. The first one is to tackle several challenges related to sentiment analysis in social media. The second one is how to make use of advanced techniques of sentiment analysis on several applications.

1.1 Background

Over the last few years, online social media have become a huge part of people's daily life, a phenomenon which has not been observed prior to our era thanks to the advances in the field of communications. This made the social networks of a typical person a combination of both his real-life social network, and his online one(s). These two have been strongly forged together to a point where people bring events happening with them online, discuss their daily life-related topics both online and offline. That being the case, the online User-Generated Contend (UGC) has been exponentially increasing, mainly on social media and blogging/microblogging platforms. With every tweet, every Facebook post, people tend to relate what happens to them, whether that regards their private life, or more interestingly regards a product, an idea, a person, a concept or a service they encounter. This UGC, despite being noisy, unregulated and full of redundancy, untrustworthiness and subjective and unreliable data, has attracted the attention of researchers for several reasons. As a matter of fact, researchers believe that UGC has tremendously changed the relationship between companies and customers. Several studies have shown that it has been shifting the power from firms to customers, altering the way marketing works [1]. Furthermore, it is believed that the trust one gives to his fellow users is far greater than that given to companies and firms [2]. This means that companies need indeed to pay more attention to what their customers share amongst themselves. Other than the content of posts and microblogs, etc., researchers have been also interested in the nature of interactions between the users which are unique to the cyberspace. Interacting from behind a screen is undoubtably different from face-to-face interactions, for good or for bad [3–6]. Nevertheless, researchers have been interested as well on online (potentially hidden) communities [7], mutual influence of users [8–10], public opinion changes, [11] or even the "creation" of online celebrities [12], etc.

Sentiment analysis, in particular, has been an interesting instance of a study that has been performed heavily on this content. To begin with, despite the existence of websites dedicated to reviews, or special sections on online shops dedicated to user reviews, the ratio of reviewers to users is way too low. Companies are looking for alternatives to collect opinions and reviews. Social media, for instance, present a good alternative if such reviews can be collected from them. In addition, people on social media tend to communicate their opinion in a less biased manner. On online shops review sections, users usually tend to write down their first impressions or report problems after a while. It is seldom the case that a reviewer writes down his good experience after using the product for a while. On social media, on the other hand, one can be asked by a friend an advice about the product he is using, and he would casually give his unbiased impression. Nevertheless, social media analysis goes beyond product analytics and customer service to cover user behaviour and human patterns identification.

That being said, social networks sentiment analysis could present a threat to one's life given that his private information are exposed, willingly or unwillingly. Such information are accessible

by influential entities, organizations for example, that are capable of exploiting them to manipulate public opinion. Facebook, for instance, was experimenting with the idea of using sentiment analysis to see if they could manipulate people's emotions. To do so, they altered their algorithms to inject sentimental posts (i.e., clearly negative or positive ones) more frequently into users' news feeds [1]. To reach their goal, they have used a process referred to as "emotional contagion" [13]. Their experiments have shown that it is indeed possible to influence their users' emotional output by flooding their news feeds with positive or negative posts. Even worse scenarios are those where users are not even aware of such murky behaviours. In the piece of news mentioned above, Facebook has never informed its users that they were part of an experiment and may have caused emotional distress to them in some cases [14].

In the next sections of this chapter, we will formally introduce sentiment analysis and present some of the relevant work related to this topic of research.

1.2 Sentiment Analysis Fundamentals

1.2.1 Definition and Applications

Definition

With the tremendous amount of content generated at a daily bases, not only by content creators, but also by average internet users, sentiment analysis has become a key tool for making sense of such amount of data. Sentiment analysis, is defined as the science of automatically identifying and extracting opinions from a large amount of data. In a typical scenario, a big amount of data regarding a specific subject, be it a product, a service, an event or other, is collected; and the target is to identify some overall statistics of how these data describe the subject. It is fair to affirm that sentiment analysis converts texts, which are rather descriptive or qualitative into numbers and statistics, thus bring a quantitative dimension allowing to measure more objectively opinions of people. In other words, out of an unstructured, subjective and unclean data, it is possible to extract very useful structured information. For instance, given a product, and a set of reviews, the goal would be to identify the proportion of reviews reflecting a positive opinion and the proportion of ones reflecting a negative opinion.

Despite being the key point of sentiment analysis, opinion is not the only feature extracted using sentiment analysis. Several additional information could be extracted using sentiment analysis. They include, but are not restricted to, the specific subject of opinion extracted (e.g., if the review includes information about several aspects of the product), the person or the group of people who hold the opinion, the degree of belief in the opinion shown and the evolution over time of this opinion.

[1] https://www.theguardian.com/technology/2014/jun/29/facebook-users-emotions-news-feeds

Applications

Sentiment analysis has a great impact that can be observed on several levels. For example, it has changed drastically the way companies collect and analyze feedbacks from users as we mentioned in the previous section. However, the applications of sentiment analysis are not restricted to company-consumer interactions. Following a set of examples of sentiment analysis applications:

- **Product analytics:** This is probably amongst the top applications of sentiment analysis. Upon receiving feedbacks from users, or collecting data from social media regarding their products, these are analyzed to keep track of what people like and dislike about the products, and how to appeal to them [15].

- **Customer support:** This falls in the same category as the previous one. In order to provide a good customer support service, feedbacks and comments can be prioritized based on how critically negative they are, so that very negative ones are processed more urgently by the customer support team [16].

- **Market analysis:** Performing sentiment analysis across different markets help a firm identify which market has been the most successful for them so that they can target it more. It also helps know which demographics to target and how do their product perform compared to competitors. Nonetheless, when data collected from reviews and feedbacks are scarce, social media offer a decent alternative to obtain the same data to analyze [17].

- **Brand monitoring:** Classically, companies ask users of their products about their opinion either directly or through surveys and questionnaires, which they analyze in a second stage to estimate how successful their brand is. This, nowadays, seems to be less used by companies with the spread of internet and the growth of online shopping from popular websites such as amazon or eBay. In these websites users are encouraged to give feedbacks , by simply logging to their accounts and filling in some simple forms, and share their experience with the products they have purchased [18, 19].

- **Mass event intention/opinion identification:** Unlike targeted surveys and questionnaires which is usually limited in geography or time, data collected from social media have no such restrictions. While questionnaires are usually well-prepared and have a clear goal, data collected from social media are very noisy and unstructured. However, it is possible, thanks to sentiment analysis, to extract the same information required, from a demography that is totally random, yet representative of the overall population targeted. Several works in the past have been proposed to predict election results, stock prices behaviour and more recently crypto-currency prices one. That being said, these are not necessarily accurate [20–22].

That being said, sentiment analysis has other applications which, as we mentioned in the previous section could be malicious or privacy invasive. However, such applications are out of the scope of this dissertation and will not be discussed here.

1.2.2 Techniques and Methods

Techniques of sentiment analysis in the literature are numerous. However, they can be grouped into two main categories: supervised techniques and unsupervised ones. A third category can be added, which combines both. Historically speaking, unsupervised techniques of sentiment analysis preceded supervised ones.

Unsupervised approaches: These are also referred to as rule-based approaches. They make use of the classic Natural Language Processing (NLP) techniques, to process a given text, then refer to a dictionary or a set of dictionaries, referred to as lexicons, to identify the polarity of the text. A typical example of such approaches is as follows:

1. Create two lists of words qualified as positive and negative,
2. Count in a given text the number of words of each list that occur in the text,
3. Subtract the number of occurring negative words from that of positive words,
4. If the result is positive, the overall text is judged positive, otherwise, it is judged as negative.

Such approach is obviously very naive; however, it introduces the basics of how unsupervised approaches for sentiment analysis work. In Chapter 3, we discuss more sophisticated unsupervised approaches for sentiment analysis.

Supervised approaches: Classically, supervised approaches of sentiment analysis do not rely on a set of rules like unsupervised approaches. They make use of machine learning techniques to identify the polarity of a text. The opinion identification is modeled as a classification problem whose goal is to attribute one of two classes to the text: positive or negative. Obviously, supervised approaches need a set of manually labeled data to learn how to distinguish between the different classes. The procedure of prediction of the class of a set of unknown data is shown in Fig. 1.1. Features are extracted from the training data and associated to their corresponding label. The machine learning algorithm learns automatically how to build its own rules to identify the labels using the given features. In the prediction phase, the features are extracted from a given instance of unknown data, and will go through the rules already established by the model built to predict the label of the given class.

It is fair to affirm that, despite qualified as rule-free, machine learning requires a set of rules to tell it how to extract features.

Machine learning-based approaches for sentiment analysis have attracted most of the attention of researchers since introduced by Pang et al. [23]. Their approach relied on words collected from the training set itself to build the rules of how to identify positive texts from negative ones. The attention given to supervised machine learning-based techniques came from the fact that these techniques are, overall, better and present better performance than unsupervised ones. However, as stated above, these techniques require a big amount of data manually labeled, an obstacle that might not be easily overcome.

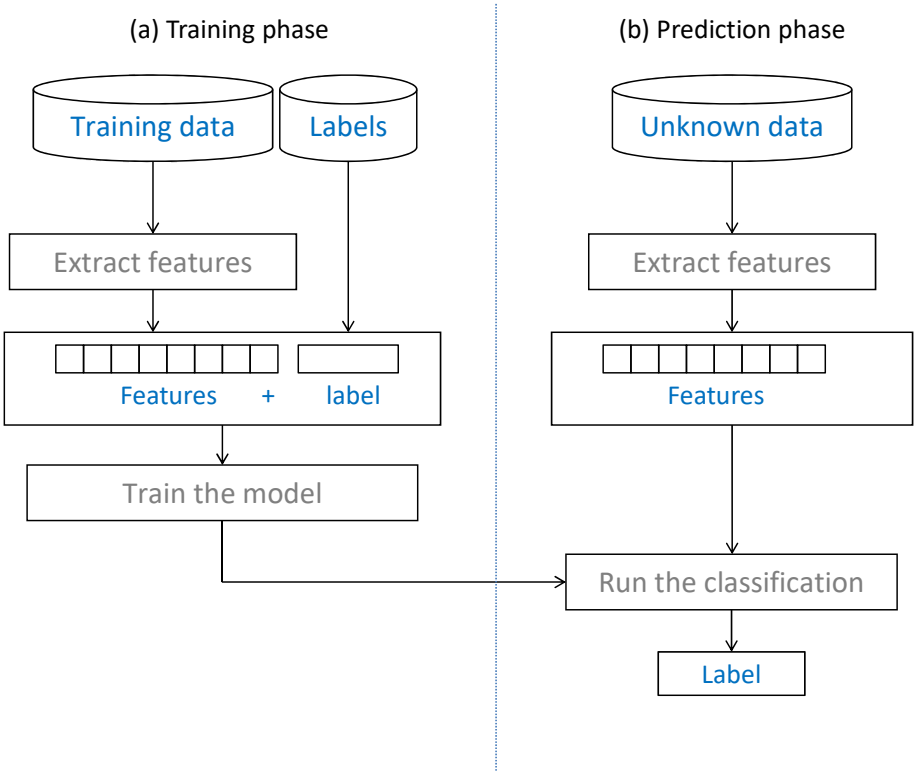

Figure 1.1: Classification Using Machine Learning

Hybrid approaches: These are approaches that combine both worlds to make use of the advantages of both. The combination of the two types of approaches can be done at different levels. For instance, given a small amount of labelled data, an unsupervised approach can be run to enrich these data to be able to run the classification using machine learning. These are sometimes referred to as semi-supervised approaches.

Another type of combination could be a voting approach that combines several approaches, supervised and unsupervised ones, to judge on the sentiment polarity of a given text.

1.3 History, Current State and Future Challenges

1.3.1 History of Sentiment Analysis

Academic research on sentiment analysis has started decades ago. However, with the spread of use of internet, the accumulation of user generated data, and more interestingly the advances in software and hardware technologies, it has become possible to process data and perform sentiment analysis on large scales. This has led to an exponential growth of deployment of sentiment analysis tools, pushing further the research in this field.

Historically, the first attempts to identify people's opinion dated back in the Greek times [24, 25]. However, these were scientifically robust studies. The first scientific journal on public opinion was released in 1937 [26]; however, works on public opinion through questionnaires and surveys preceded that [24]. These works were mostly made for political purposes [27]. Works on sentiment

Figure 1.2: Use of Hashtags in Tweets

analysis and opinion mining have since increased in number. However, the spread of internet in the last decade of the 20th century and the first decades of the new millennia made this topic of research more interesting and more attractive to the research community. According to Mäntylä et al. [24] 99% of the publication made in this field were published after 2004.

With this tremendous amount of work on the field, sentiment analysis has been divided into further narrower fields depending on its application such as customer support using sentiment analysis [15], stock price prediction using sentiment analysis [28], etc.

Techniques wise, the work of Pang. et al. [23] presents one of the most important milestones, introducing the usage of machine learning to perform sentiment analysis. Another important milestone is the appearance and spread of online social networks and microblogging websites. Facebook and more interestingly Twitter have allowed researchers to collect a tremendous amount of data that can be used to perform sentiment analysis. The introduction and spread of Hashtags by Chris Messina [2] made the task of sentiment analysis even easier. Hashtags are presonalized words or phrases preceded by the hash symbol "#" used in social media to identify messages of a certain topic (Fig. 1.2). Thanks to hashtags, companies can easily collect texts and messages posted on social media discussing their product for example to perform sentiment analysis on them.

The new breakthroughs in the field of deep learning, mainly the works of Lucen et al. [29] Hinton et al. [30] and Krizhevsky et al. [31], have marked another milestone in the field of sentiment analysis. Thanks to these works, it has become possible to train big neural network with large amounts of data in a reasonable amount of time while making sure the training converges. Despite being initially focused on computer vision and image classification, deep learning has attracted researchers from different field thanks to its potential and impressive results compared with conventional machine learning techniques. These fields include, among others, the field of sentiment analysis.

[2] https://www.hashtags.org/featured/hashtag-history-when-and-what-started-it/

1.3.2 Current State

Sentiment analysis is currently one of the hottest topics of research. The state-of-the-art approaches have reached impressive results on data collected from several sources on the internet varying from movie reviews [32, 33] and amazon reviews [34, 35] to tweets and posts collected from social media [36, 37]. Following, we introduce the most common tasks of sentiment analysis, along with some relevant works which dealt with each of them:

- **Polarity detection:** as its name indicates, this is the basic task of sentiment analysis aiming to detect the sentiment polarity of a given text.
- **Subjectivity detection:** this refers to the detection of the level of subjectivity of a given opinion expressed in a text. Excessive use of personal pronoun and opinion words (e.g., "I think") or exaggerative adverbs (e.g. "amazingly") are good indicators to detect such aspect.
- **Cross-lingual sentiment analysis:** this covers several aspects of sentiment analysis such as the use of multi-lingual dictionaries and the use of translation to improve the detection of sentiment polarity.
- **Opinion spam detection:** a certain behavior has been observed over the last decades from some companies which spread "fake" and biased reviews of their own to give an impression of having a good product. Identifying such spammy reviews has increasingly been attracting the attention of research community.
- **Measurement of review usefulness:** the objective of this branch of sentiment analysis is to evaluate which reviews shared are indeed useful and could help both consumers and companies understand the real value of a product or a service.
- **Applications of sentiment analysis:** several works have been proposed to apply sentiment analysis on very specific cases such as trying to predict the results of some elections, or highlight the impact of some event, etc. Applications of sentiment analysis vary very widely and new applications are being created every day, on the research level as well as in the industrial level.

These have been the most common tasks of sentiment analysis, but they are not the only ones. Other tasks include the identification of vagueness in opinionated texts, hate speech detection, etc.

Table 1.1 illustrates some of the relevant works on the tasks described above.

1.3.3 Challenges

In this subsection, we list several of the most challenging aspects of sentiment analysis. These include, but are not restricted to the following challenges:

- Time and space-dependent sentiment analysis,
- Identification and profiling of opinion holders,
- Identification of the aspects of the sentiment analysis,

Table 1.1: Relevant Work Related to the Defined Sentiment Analysis Tasks

Task	Related work
Polarity detection	Wilson et al. [38], Ortigosa et al. [39], Popescu and Strapparava [40], Kanayama et al. [41]
Subjectivity detection	Wang et al. [42], Banea et al. [43], Bravo Marquez et al. [44], Molina-González et al. [45]
Cross-lingual sentiment analysis	Hiroshi et al. [46], Wang et al. [47], Martín-Valdivia et al. [48]
Opinion spamming detection	Heydari et al. [49], Ott et al. [50, 51], Banerjee and Alton [52]
Measurement of review usefulness	Liu et al. [53], Krishnamoorthy [54], Purnawirawan et al. [55]
Applications of sentiment analysis	Nobata et al. [56], Sriram et al. [57], Cabanlit et al. [58], Hodeghatta et al. [59]

- Identification of sarcastic statements, and
- Fined-grained sentiment analysis.

In the remainder of this subsection, we describe in more details each of these challenges.

Time and space-dependent sentiment analysis

Amongst the most challenging tasks of sentiment analysis is to structure the data so that we obtain an overview of the geographical distribution of people's opinions. An even more challenging task is to keep track of the changes over the time of these opinion. This is in particular more challenging when performed on data collected from social media where unstructured, unreliable and untrustworthy data are posted from all over the world.

In a very recent event, Samsung has revealed their newest foldable phone which was believed to be the pioneer device of the next generation of mobile devices. This device was received with a huge hype and enthusiasm. However, with the problems that had occurred to multiple review units, the phone has had a very offensive criticism[3]. Despite this overall observation, such a big company would be very interested in studying both the hype phase and the criticism phase deeply, extracting information related to the geographic distribution of both. Such task is very difficult to perform on the cyberspace of internet, while avoiding the invasion of user's private life.

That being said, location and time-based sentiment analysis has been addressed by researchers in some recent works such as the work of Almatrafi et al. [60] which discussed the inequitable distribution of sentiment polarities over different regions of India and that of Paul et al. [61] which investigated trends based on geographical and temporal basis.

Identification and profiling of opinion holders

The term "opinion holder" refers to the person or group of people how share the same opinion expressed in a piece of text. More importantly than the individuals themselves is the profiling of

[3]https://edition.cnn.com/2019/04/18/tech/samsung-galaxy-fold-breaking-debacle/index.html

these users. A very interesting task would be the identification of the common characteristics of people who share a certain opinion. Several works have investigated this task. Some have used techniques such as conditional random fields [62], some have used convolution kernels [63], some have used Maximum Entropy models. Nevertheless, with the advances in the field of deep learning, Katiyar and Cardie [64] investigated the use of deep bidirectional LSTMs for joint extraction of opinion entities and the IS-FROM and IS-ABOUT relations that connect them, to identify, among others, opinion holder.

Identification of the aspects of the sentiment analysis

Aspect-based sentiment analysis refers to the identification of the opinion of people towards specific entities of the subject of study. A typical example is as follows: a phone manufacturer is interested in understanding the impression of users about a newly released phone. Users, when reviewing the phone, provide their opinion about several aspects of the phone: the screen, the camera, the battery, etc. Therefore, it might be interesting to identify these individual opinions regarding each of these aspect separately. This procedure is referred to as aspect-based sentiment analysis. Aspect-based sentiment analysis is a very challenging task, especially when performed on data collected from social media and microblogging websites. This is because, unlike proper reviews on review websites, data from social media are unstructured and very noisy, and there is no clear indication of what is being discussed at a given moment.

Aspect-based sentiment analysis has attracted the attention of researchers. Several works were proposed in the literature to tackle this topic.

Che et al. [65] proposed an approach that compresses complicated sentiment sentences into ones that are shorter and easier to parse and applied a discriminative conditional random field model to perform the aspect-based sentiment analysis. Similar works were proposed by Singh et al. [66]. Deep learning has also been used in this context with works such as that of Nguyen and Shirai [67] who proposed a neural network architecture they called PhraseRNN (Phrase Recursive Neural Network) which they used to run aspect-based sentiment analysis.

Identification of sarcastic statements

Sarcasm can be roughly defined as “Conveying contempt by saying the opposite of what is really meant.” In other words, the real meaning of a sarcastic statement is the opposite of what it appears to be saying. That being the case, sarcastic statements are a main reason of misclassification when sentiment analysis is performed. This is because sentiment analysis systems and tools rely on the apparent meaning to detect the polarity of a given text. Therefore, identifying sarcastic statements is of a great importance towards improving and polishing sentiment analysis.

Chapter 2 of this dissertation tackles the problem of sarcasm detection, and how it can be used to improve sentiment analysis.

Fine-grained sentiment analysis

Fine-grained sentiment analysis refers to the process of identifying a higher resolution of sentiment of a given text. In other words, instead of the binary classification of a text (i.e., guessing whether

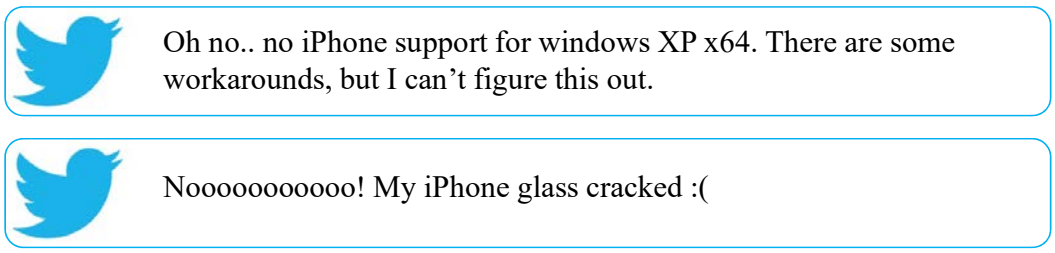

Figure 1.3: Negative Tweets with Different Emotions Expressed

a text is positive or negative), fine-grained sentiment analysis divides these two classes into further ones. For instance, it is possible to sub-divide the class "positive" into 3 different classes: very positive, positive, almost positive. The same can be done with the class "negative." A class in-between, such as the class "neutral" englobing texts with no apparent sentiment shown can be added as well.

An even more interesting task would be to identify the emotion of the opinion holder, or even the emotion a text triggers on the reader. For example, the class "positive" can be divided into multiple classes such as happiness, love, enthusiasm, etc. The class "negative" can be divided into multiple classes as well such as anger, satisfaction, sadness, etc. To concretize, given the two tweets shown in FIg. 1.3, two different sentiments/emotions are shown in them despite being both negative and discussing the same product of a well-known company. While the first shows emotions of anger and frustration, the second shows emotions of sadness. That being the case, the interpretation of these emotions are different as well from the company's perspective. Therefore, the identification of these individual sentiments is very important and could help the company prioritze one over the other.

This task has been tackled in Chapters 3 and 5 where we introduce two approaches, one to perform fine-grained sentiment classification, and the other to solve the a common issue with such systems (i.e., systems that perform fine-grained classification).

1.4 Scope and Contributions of the Dissertation

1.4.1 Summary of the Dissertation

This dissertation consists of six chapters. Chapters 2, 3, 4 and 5 present novel techniques to tackle some of the most challenging open problems in sentiment analysis. These include the identification of sarcastic statements, the fine-grained sentiment analysis and a newer way to look at the fine-grained sentiment analysis problem and deal with it. Chapters 2, 3 and 5 contain, each, a particular problem statement, relevant related work existing in the literature, a description of the proposed method to handle it and an evaluation to its efficiency. Chapter 4, on the other hand,

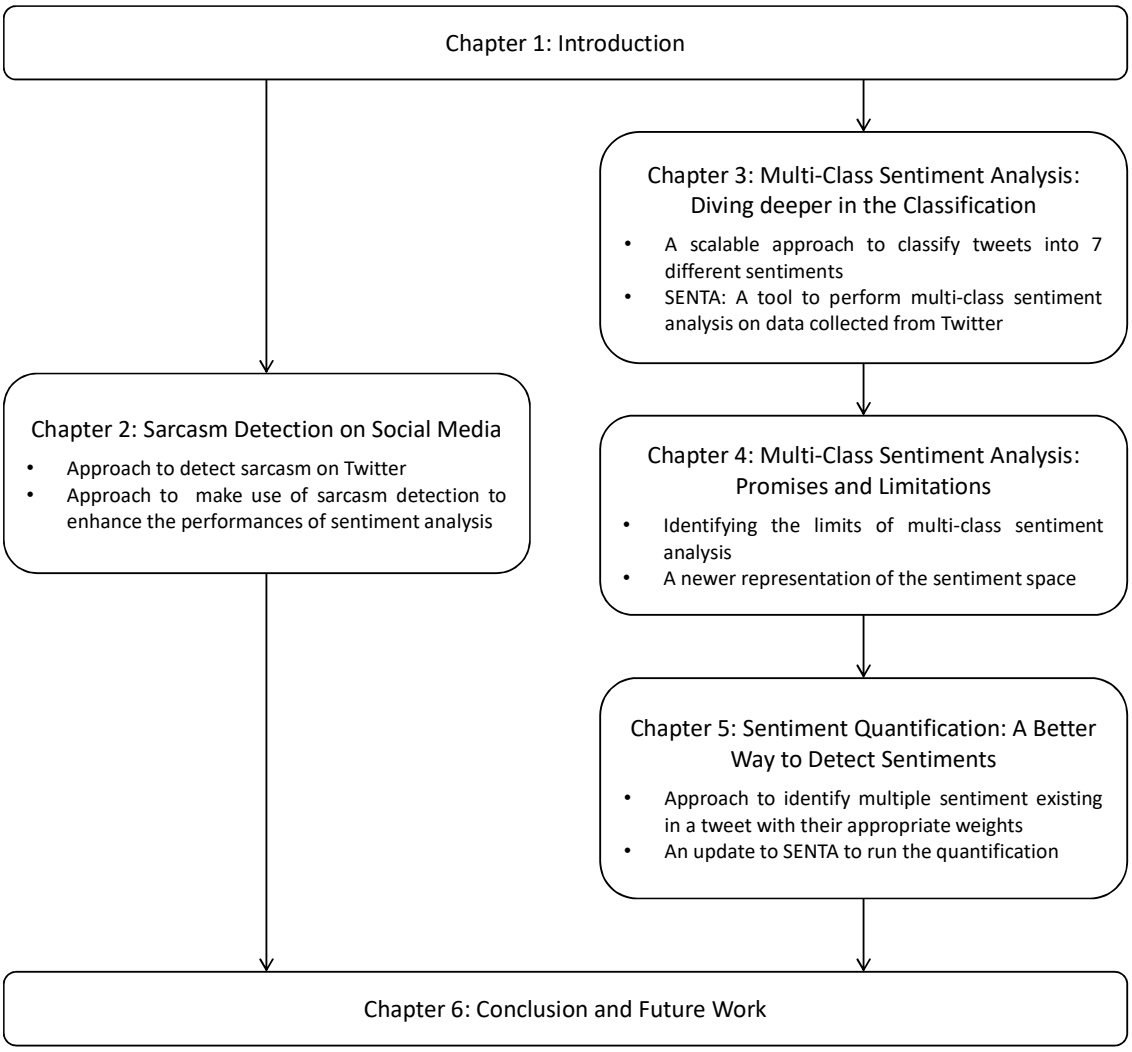

Figure 1.4: Configuration of this Dissertation

discusses the task of fine-grained sentiment analysis, also know as multi-class sentiment analysis, describes its inherently challenging problems, and introduces the task of sentiment quantification, which will be discussed in Chapter 5. The overall outline of this dissertation os summarized in Fig. 1.4.

1.4.2 Scope of the Dissertation

Sentiment analysis cover a wide range of sub-topics. As a matter of fact, sentiment analysis can be applied on different types of data. These include structured and unstructured data. While it is very useful to perform sentiment analysis on structured data, these are scarce on the internet and hard to collect. Unstructured data on the other hand are way more abundant, and are increasing in size exponentially with the amount of daily user-generated content. Unstructured data are, however, hard to analyze and present several challenges as we explained in previous sections.

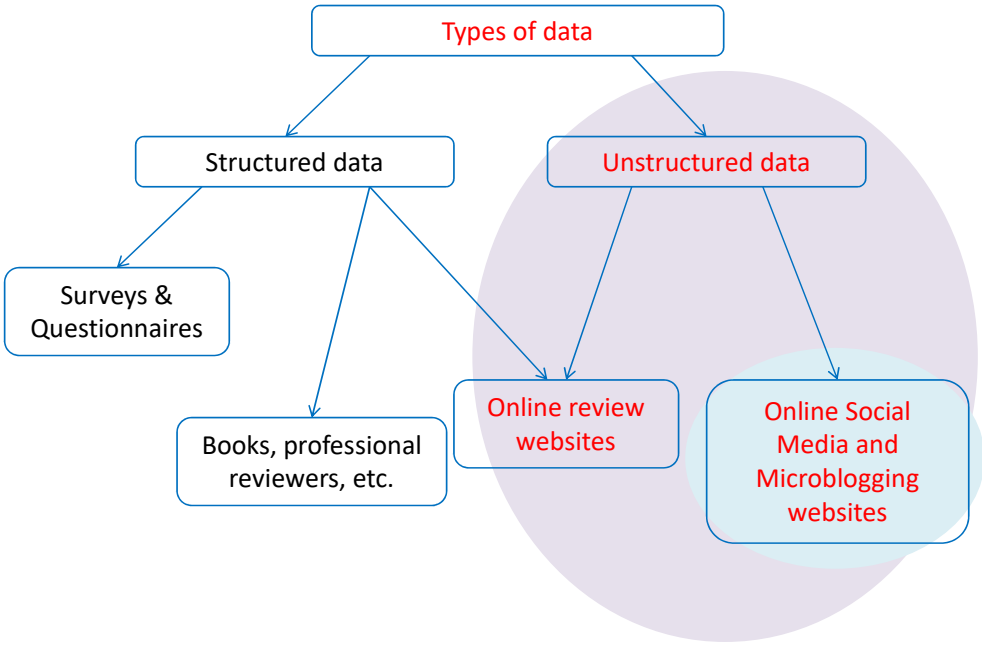

Figure 1.5: Type of Data Subject to Sentiment Analysis

This dissertation will focus on this particular type of data, more particularly on data collected from social media and microblogging websites as shown in Fig. 1.5. The proposed approaches are, nonetheless, applicable to other types of unstructured data.

In Fig. 1.6, we show some of the main challenges related to the field of sentiment analysis. These challenges have been discussed in more details previously in Section 1.3.3. They include fine grained sentiment analysis, time and space-dependent sentiment analysis, profiling of the opinion holders and handling sarcasm. These are, by no means, the only ones. There are several others challenges that have been addressed by the research community. However, we highlight the ones related to our work, and which we deal with in the remainder of this dissertation. Being one of the toughest challenges, the problem of sarcasm detection is tackled in chapter 2. This chapter also discusses how it can be used to enhance sentiment analysis. Chapters 3 and 4 tackle the problem of fine-grained sentiment analysis (multi-class sentiment analysis), whereas chapter 5 introduces a new task we refer to as sentiment quantification, and proposes a way to perform it.

In Fig. 1.7, we show the position of our approach to perform sentiment analysis and identify sarcastic statements in the literature. A wide variety of types of features and techniques have been used in several works. These include n-grams, textual and non-textual components of the text, etc. In our work, we propose a set of out-of-the context pattern features which we use in addition to other features to train a classifier. It is worth mentioning that identifying sarcastic statements independently from the time or dialogue context has always been a challenging task.

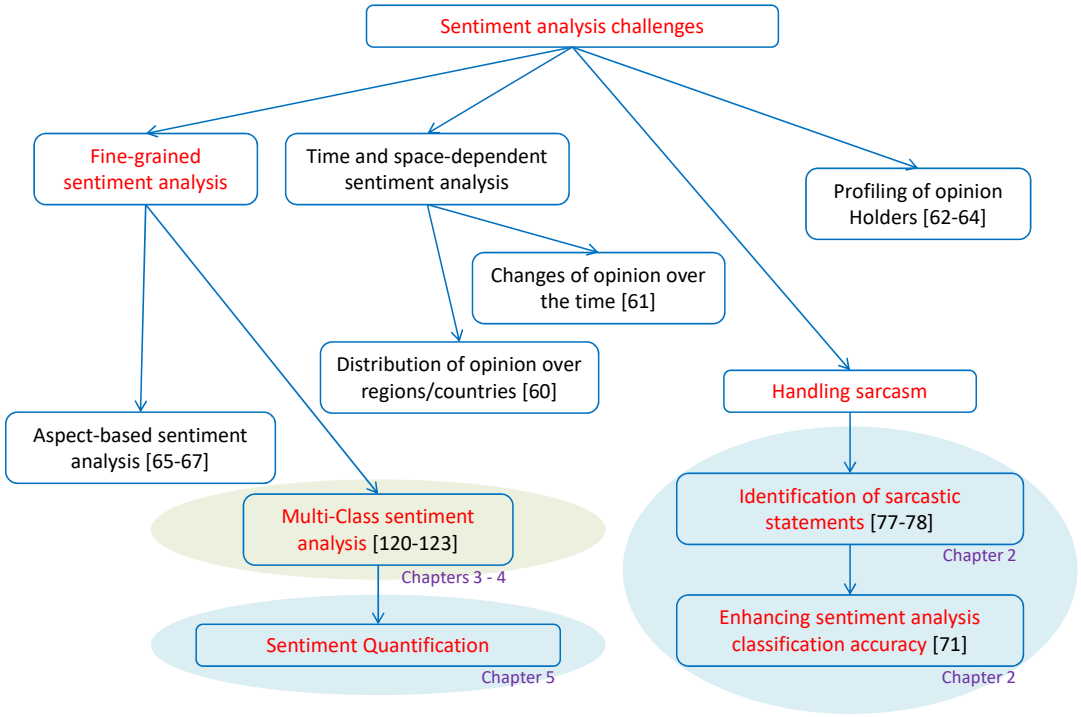

Figure 1.6: Main Challenges Related to the Field of Sentiment Analysis

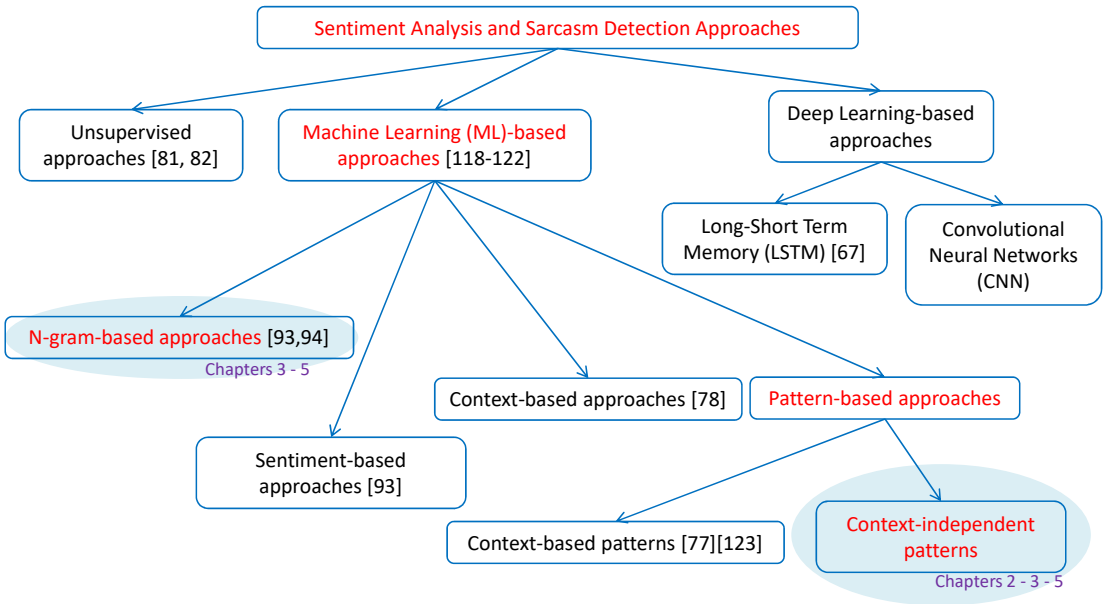

Figure 1.7: Sentiment Analysis and Sarcasm Detection in the Literature

1.4.3 Contributions of the Dissertation

This dissertations introduces the concept of usage of writing patterns as a way to detect one's sentiments and/or sophisticated forms of speech such as sarcasm. Patterns are collected based on Part-of-Speech (PoS) Tags of words. We defined a metric to measure the resemblance between different patterns and used this metric to extract several features from a given text, which we use alongside other features to train a classifier and perform the classification.

The dissertation also introduces a tool we have built and called SENTA (SENTiment Analyzer) which we used to perform the different tasks.

In Chapter 2, we propose an approach that uses out-of-context patterns, alongside with other features to perform sarcasm detection. Our approach outperforms clearly the baseline ones. In addition, in contrast to other works which assume that sarcasm is a polarity switcher, we elaborate more the idea of use of sarcasm detection to identify the polarity of a given piece of text.

In Chapters 3 and 4, we extend the concept of patterns to another dimension and make use of such type of features to perform the task of multi-class sentiment analysis. We introduce our tool SENTA which offers the possibility to extract multiple types of features, including but not limited to patterns. We also discuss the problems which make multi-class sentiment analysis very challenging.

These challenges are targeted in Chapter 5 which introduces the concept of sentiment quantification. Sentiment quantification refers to the identification of all existing sentiments in a given text and attributing a score showing how strong they are. This task is part of the novelty introduced in this thesis. Nevertheless, SENTA has been further enhanced to perform such task.

In Tables 1.2, 1.3, 1.4 and 1.2, we summarize the objectives of each of the aforementioned chapters (i.e., chapters from 2 to 5), we give a brief description of the conventional works as well as the limitations these have, and we show our proposed approaches along with their contribution.

Table 1.2: Summary of Chapter 2

Objective	• Identify sarcastic statements in Twitter posts (i.e., tweets). • Use this information (whether a given tweet is sarcastic or not) to enhance sentiment analysis accuracy.
Conventional Approaches	1. Davidov et al. [77]: • Rely on context-based patterns. • Use a dataset of 5.9 million tweets. • Use a k-nearest neighbors (KNN) classifier to classify tweets into sarcastic and non-sarcastic. 2. Riloff et al. [96]: • Propose a bootstrapping algorithm to detect a specific type of sarcasm. • Start with the seed word "love" and a set of sarcastic tweets. To learn all possible positive sentiment and negative situation phrases. 3. Rajadesingan et al. [78]: • Study the behavior of users and the psychology behind sarcasm. • Propose a system that detects sarcasm based on the history of users • Extract 5 types of features, each dealing with a specific type of historical information • Use Support Vector Machine (SVM) classifier to perform the classification
Conventional Approaches Limitations	1. Davidov et al. [77]: • Big number of features: Slow to run. • Uses a very large set of 5.9 tweets to build the model. 2. Riloff et al. [96]: • Detects only one type of sarcasm, which is not very commonly used. • Supposes that all positive expressions and negative situations are present in the training set. 3. Rajadesingan et al. [78]: • Requires a previous knowledge base for all users • Information regarding the sentiment and the sarcasm orientation are collected for all the previous tweets • Highly context dependent
Contribution	• Identify the purposes of use of sarcasm • Use Part-of-Speech (POS) tag based patterns • Use few number of features (i.e., 62 features): non contextual features • Use a fairly small training set: 6,000 tweets for training
Summary of Findings	• Faster model: - Training time: 8.12 sec. - Execution time: 2ms per tweet. • Performances that are comparable to those of the complex models (i.e., accuracy = 90.1% and precision = 91.3% during cross-validation). • It is possible to enhance Sentiment Analysis performance of fairly fast models such as the one proposed in [113] from an accuracy equal to 83.67% to 87%.

Table 1.3: Summary of Chapter 3

Objective	For a given tweet, identify, out of multiple sentiment classes, the one that represents the most the emotion shown in the tweet.
Conventional Approaches	1. Lin et al. [120, 121]: • Classifys documents into reader-emotion categories. • Emotion-based features. • Kanji- and Chinese word based features. 2. Liang et al. [123]: • Emoticon (smiley) recommendation for posted texts. • Features: Similarity measures between emoticon trajectories. 3. LIWC (Linguistic Inquiry and Word Count): • Tool to extract different types of information (features) from texts in an automated way.
Conventional Approaches Limitations	1. Lin et al. [120, 121]: • Reader-oriented: focus more on the sentiment the reader feels to show results on search engines 2. Liang et al. [123]: • Prediction of emoticons to show for the post writer (Top N emoticons). 3. LIWC (Linguistic Inquiry and Word Count): • Paid, not open Source. • Does not allow the extraction of writing patterns.
Contribution	• Develop a free, open-source and flexible tool to extract all possible information from texts: SENTA (SENTiment Analyzer) is an open-source tool that allows the extraction of different types of features from texts, including patterns to perform sentiment analysis • A set of pattern-based features, along with other features to classify tweets. • Classification of tweets into 7 different sentiment classes: love, fun, happiness, hate, anger, sadness and neutral • Detect the emotion expressed by the writer
Summary of Findings	• Binary classification: Accuracy equal to 81.3%. • Ternary classification: Accuracy equal to 70.1% • Multi-class classification: Accuracy equal to 60.2%. • High recall for some sentiments (e.g.,“Hate” and “Love” have respectively accuracies equal to 90.9% and 75.2%).

Table 1.4: Summary of Chapter 4

Objective	• Identify why Multi-Class Sentiment Analysis (MCSA) inherently a hard task. • Find a representation for sentiments that allow to identify the level of correlation of ones with the others.
Contribution	• A novel representation of the sentiment space. • A measure of the distance between the sentiments. • Identification of the main challenges and main reasons of misclassification when performing MCSA.
Observations	• Some sentiments are (highly) correlated (e.g., "Happiness" and "Fun"). • Multi-class sentiment analysis challenges: – Context Dependency and Polysemy (Words having different meanings), – Presence of multiple sentiments within a piece of text, – Closeness between some sentiments, and – Absence of sentiment indicators.
Summary of Findings	• Main challenge in MCSA: Presence of multiple sentiments within a piece of text • A possible solution: Performing Sentiment Quantification instead: identifying all existing sentiments and attributing scores highlighting their weights

Table 1.5: Summary of Chapter 5

Objective	• Instead of classifying tweets into one from multiple sentiment classes, detect and quantify all the sentiments present in each tweet. • This task is referred to as "quantification".
Conventional Approaches	• Only multi-class sentiment analysis (MCSA).
Conventional Approaches Limitations	• Even MCSA has not been well studied in the literature.
Contribution	• Introduce the task of sentiment quantification. • Propose an approach that relies on writing patterns along with other sets of features to perform a ternary sentiment classification of tweets (i.e., the classification into positive, negative and neutral). • Upon classification, the writing patterns are used again to attribute scores for each sentiment in every tweet. These scores are used to filter the sentiments we judge as being conveyed in the tweet (within the process we refer to as quantification). • The required quantification components are added to the previously introduced tool SENTA, to make it easy to run the approach.
Summary of Findings	• F1 score equal to 45.9% and 44.5% on two different test sets.

Chapter 2

Sarcasm Detection on Social Media

2.1 Introduction

Twitter became one of the biggest web destinations for people to express their opinions, share their thoughts and report real-time events, etc. Throughout the previous years, Twitter content continued to increase, thus constituting a typical example of the so-called big data. Today, Twitter has more than 330 million active users, and more than 500 million tweets are sent every day[1]. Many companies and organizations have been interested in these data for the purpose of studying the opinion of people towards political events [68], popular products [69] or movies [59].

However, due to the informal language used in Twitter and the limitation in terms of characters (i.e., formally 140 characters per tweet, 280 characters per tweet now), understanding the opinions of users and performing such analysis is quite difficult. Furthermore, presence of sarcasm makes the task even more challenging: sarcasm is when a person says something different from what he means. Liebrecht et al. [70] discussed how sarcasm can be a polarity-switcher, and Maynard et al. [71] proposed a set of rules to decide on the polarity of the tweet (i.e., whether it is positive or negative) when sarcasm is detected.

The online Oxford dictionary[2] defines sarcasm as *"the use of irony to make or convey contempt"*. Collins dictionary[3] defines it as *"mocking, contemptuous, or ironic language intended to convey scorn or insult"*. However, sarcasm is a deeper concept, highly related to the language, and to the common knowledge.

Although different from one another, sarcasm and irony have been studied as two close and very correlated concepts [72–74] or even as the same one [75–77]. The Free Dictionary[4] defines it also as a form of irony that is intended to express contempt. Since most of the focus on sarcasm is to enhance and refine the existing automatic sentiment analysis systems, we also use the two terms synonymously.

Some people are more sarcastic than others, however, in general, sarcasm is very common, though, difficult to recognize. In general, people employ sarcasm in their daily life not only to make jokes and be humorous but also to criticize or make remarks about ideas, persons or events. Therefore, it tends to be widely used in social networks, in particular microblogging websites such as Twitter. That being the case, the state of the art approaches of sentiment analysis and opinion mining tend to have lower performances when analyzing data collected from such websites. Maynard et al. [71] show that sentiment analysis performance can be highly enhanced when sarcasm within the sarcastic statements is identified. Therefore, the need for an efficient way to detect sarcasm arises.

In this chapter, we introduce an efficient way to detect sarcastic tweet. Although it does not need an already-built user knowledge base as in the work of Rajadesingan et al. [78], our approach considers the different types of sarcasm and detect the sarcastic tweets regardless of their owners or their temporal context, witch a precision that reaches 91.1%.

Therefore, the main contributions of this chapter are as follows:

1. We identify the main purposes for which sarcasm is used in social networks.

[1] https://www.statista.com/
[2] http://www.oxforddictionaries.com/
[3] http://www.collinsdictionary.com/
[4] https://www.thefreedictionary.com

2. We propose an efficient way to detect sarcastic tweets, and study how to use this information (i.e., whether the tweet is sarcastic or not) to enhance the accuracy of sentiment analysis.

3. We study the added value of the different sets of features used, in particular, in terms of precision of detection.

The remainder of this chapter is structured as follows: Section 2.2 presents our motivation for this work and Section 2.3 describes some state of the art work related to our proposed approach. Section 2.4 describes our proposed approach for sarcasm detection. In Section 2.5, we present and discuss the obtained results of the approach. In Section 2.6, we show how sarcasm can be used to enhance sentiment analysis systems and Section 2.7 concludes this chapter.

2.2 Motivations

As mentioned above, the identification of sarcasm helps enhance sentiment analysis task when performed on microblogging websites such as Twitter. Sentiment analysis and opinion mining rely on emotional words in a text to detect its polarity (i.e., whether it deals *"positively"* or *"negatively"* with its theme). However, the appearance of the text might be misleading. A typical example of that is when the text is sarcastic. In Twitter, such sarcastic texts are very common. *"All your products are incredibly amazing!!!"* might be considered as a compliment. However, considering the following tweet *"Did I say incredibly?? Well, it's true, nobody would believe that. They break the second day you buy them -_-"*, the user explicitly explains that he did not mean what he said. Although some users indicate they are being sarcastic, most of them do not. Therefore, it might be indispensable to find a way to automatically detect any sarcastic messages.

Through their work, Rajadesingan et al. [78] highlighted the limitations of some state of the art tools that perform sentiment analysis, when more sophisticated forms of speech such as sarcasm are present. They explained why sarcasm is hard to detect even by humans, and showed how the nature of tweets makes it even more complicated. Therefore arise the importance of detection of sarcastic utterances in Twitter.

However, several challenges arise and make the task complicated. Joshi et al. [79] highlighted 3 main challenges which are i) the identification of common knowledge, ii) the intent to redicule, and iii) the speaker-listener (or reader in the case of written text) context.

On a related context, even though Brown et al. [72] stated that sarcasm *"is not a discrete logical or linguistic phenomenon"*, works such as [76, 77] were proposed to identify sarcastic writing patterns to decide on whether or not an utterance is sarcastic. During our experiments as well as while manually annotating tweets, we noticed that such patterns exist, in particular among non-native speakers of English. Therefore, we focus on detecting and collecting such patterns from a manually annotated dataset, and we quantify them so that we can judge whether or not a given tweet is sarcastic by comparing patterns extracted from it to them.

Throughout this work, we present a pattern-based framework that performs the task of sarcasm detection, a framework relatively easy to implement, and that presents performances competitive to those of more complex ones.

2.3 Related Work

In the last few years, more attention has been given to Twitter sentiment analysis by researchers, and a number of recent papers have been addressed to the classification of tweets. However, the nature of the classification and the features used vary depending on the aim. Sriram et al. [57] used non-context-related features such as the presence of slangs, time-event phrases, opinioned words, and the Twitter user information to classify tweets into a predefined set of generic classes including events, opinions, deals, and private messages. Akcora et al. [11] proposed a method to identify the emotional pattern and the word pattern in Twitter data to determine the changes in public opinion over the time. They implemented a dynamic scoring function based on Jaccard's similarity [80] of two successive intervals of words and used it to identify the news that led to breakpoints in public opinion.

However, most of the works focused on the content of tweets and were conducted to classify tweets based on the sentiment polarity of the users towards specific topics. A variety of features was proposed. Not only they include the frequency and presence of unigrams, bigrams, adjectives, etc. [23], but they also include non-textual features such as emoticons [81] (i.e., facial expressions such as smile or frown that are formed by typing a sequence of keyboard symbols, and that are usually used to convey the writer's sentiment, emotion or intended tone) and slangs [82]. Dong et al. [83] proposed a target-dependent classification framework which learns to propagate the sentiments of words towards the target depending on context and syntactic structure.

Sarcasm, on the other hand, and irony in general have been used by people in their daily conversations for a long time. Therefore, sarcasm has been subject to deep studies form psychological [84] and even neurobiological [85] perspectives. Nevertheless, it has been studied as a linguistic behavior characterizing the human being [78]. In this context, researchers have recently been interested in sarcasm, trying to find ways to automatically detect it when it is present in a statement. Although some studies such as [72] highlighted that, unlike irony, sarcasm *"is not a discrete logical or linguistic phenomenon"*, many works have been proposed and present high accuracy and precision.

Burfoot et al. [86] introduced the task of filtering satirical news articles from true newswire documents. They introduced a set of features including the use of profanity and slangs and what they qualified of *"semantic validity"*; and used SVM classifier to recognize satire articles.

Campbell et al. [87] studied the contextual components utilized to convey sarcastic verbal irony and proposed that sarcasm requires the presence of four entities: allusion to failed expectation, pragmatic insincerity, negative tension and presence of a victim, as well as stylistic components.

Nevertheless, other works have been proposed to represent sarcasm. Some of these representations are given in [79] as follows:

- Wilson et al. [88] suggested that sarcasm arises when there is a situational disparity between the text and the context.

- Ivanko et al. [89] suggested that sarcasm requires a 6-tuple consisting of a speaker, a listener, a context, an utterance, a literal proposition and intended proposition.

- Giora et al. [90] suggested that sarcasm is a form of negation in which an explicit negation marker is lacking. This implies that the sarcasm is namely a polarity-shifter.

As for the task of detection itself, several goals were defined. Tepperman et al. [91] studied the occurrence of the expression *"yeah right!"*, and whether it appears in a sarcastic context or not. They proposed an approach to automatically detect sarcasm present in spoken dialogues, using prosodic, spectral and contextual cues. However, this represents the main shortcoming for their approach: absence of such components makes it impossible to detect sarcasm. In other words, although the approach itself is very effective in detecting when a specific expression is sarcastic, this approach is unable to detect any type of sarcasm that might occur. Veale et al. [92] annotated the occurrences of similes such as *"as cool as a cucumber"* into ironic or not. This works presents the same shortcoming as that of Tepperman et al. [91]. Barbieri et al. [93] proposed to classify texts into politics, humor, irony and sarcasm. Ghosh et al. [94] formulated the task of sarcasm detection as a sense disambiguation task where a word can have a literal sense or a sarcastic one, and therefore, through detecting the sense of the word, sarcasm can be detected. Wang et al. [95] suggested that, rather than trying to detect whether a tweet is sarcastic or not, it makes more sense to take into account the context: they modeled the problem as a sequential classification task. However, most of the works simply aim to classify a set of texts as sarcastic and non-sarcastic.

Davidov et al. [77] and Tsur et al. [76] proposed a semi-supervised sarcasm identification algorithm. They experimented on two data sets: one from amazon and the other from Twitter. The results they obtained were interesting, though their approach relies on the frequency of appearance of words which might be misleading if the training set is not balanced in terms of topics it deals with or if the data are not big enough. In addition, it treats what is called *"Context Words"* in the same way regardless of their grammatical function. It also does not make difference between sentimental words and non sentimental words. Patterns that do not consider the emotional content of words, or discard some emotional words because of their low presence might reduce the potential of the approach.

Maynard et al. [71] relied on hashtags that Twitter users employ in their tweets to identify sarcasm in Twitter. They also studied how the detection of sarcasm can highly enhance the sentiment analysis of tweets, and proposed a rule to decide on the polarity of the tweet (i.e., whether it is positive or negative) depending on the apparent sentiment of the tweet and the content of the hashtag.

Riloff et al. [96] proposed a method to detect a specific type of sarcasm, where a positive sentiment contrasts with a negative situation. They introduced a bootstrapping algorithm that uses the single seed word *"love"* and a collection of sarcastic tweets to automatically detect and learn expressions showing positive sentiment and phrases citing negative situations. Their approach shows some potentials. However, most of the sarcastic tweets in Twitter do not fall in the aforementioned category of sarcasm. In addition, the approach relies on the existence of the all possible *"negative situations"* on the training set, which makes it less efficient when dealing with new tweets.

Rajadesingan et al. [78] went deeper and dealt with the psychology behind sarcasm. They introduced a behavioral modeling for detecting sarcasm in Twitter. They identified different forms of sarcasm and their manifestation in Twitter, and demonstrated the importance of historical information collected from the past tweets for sarcasm detection. Although, it has proven to be very

efficient, the approach is less performant when there is no previous knowledge about the user. Most of the features extracted rely on data collected from previous tweets to judge. For a realtime stream of tweets, where random users are posting tweets, it is hard to run the approach, the size of the knowledge-base grows very fast, and the training should be redone each time based on the new tweets collected (i.e., since the previous tweet has the highest impact on the current one, the new tweet should be taken into consideration for the next iteration).

Muresan et al. [97] proposed a method to construct a corpus of sarcastic Twitter messages, where the author of the tweet provides the information whether or not a tweet is sarcastic. Throughout their work, they investigated the impact of lexical and pragmatic factors on machine learning performance to identify and detect sarcastic tweets and ranked the features according to their contribution to the classification.

Fersini et al. [98] introduced a Bayesian Model Averaging ensemble that takes into account different classifiers, according to their reliability and their marginal probability predictions to make a voting system more sophisticated than the conventional majority voting one.

Bharti et al. [99] proposed two approaches for detecting sarcastic tweets: the first one is a parsing-based lexicon generation algorithm and the second one uses the occurrences of interjection words.

In general, and based on the method and features used, we can classify these works into 3 categories:

- **Rule-based approaches** such as the work of Maynard et al. [71] and that of Ghosh et al. [94],
- **Semi-supervised approaches** such as the works proposed by Tsur et al. [76], that proposed by Davidov et al. [77] and that proposed by Bharti et al. [99],
- **Supervised approaches** such as the work of Muresan et al. [97], that of Wang et al. [95] and that of Rajadesingan et al. [78].

As for the features used in the supervised approaches they fall mainly into 3 sets:

- n**-gram-based features**, which have been used along with other features in the majority of the works such as the works of Barbieri et al. [93], Riloff et al. [96] and that of Ghosh et al. [94],
- **Sentiment-based features** such as the works of Reyes et al. [100, 101] and Joshi et al. [102],
- **Saracstic pattern-based features** such as the works of Tsur et al. [76], Davidov et al. [77] and Riloff et al. [96], etc.

Other works added the contextual features to enhance the classification, whether the context is the historical context as in [78], the conversation context as in [102, 103] or the topical context as in [95].

In our work, we opt for a supervised approach that learns sarcastic patterns extracted based on the part-of-speech of words used.

2.4 Proposed Approach

Given a set of tweets, we aim to classify each one of them depending on whether it is sarcastic or not. Therefore, from each tweet, we extract a set of features, refer to a training set and use machine learning algorithms to perform the classification. The features are extracted in a way that makes use of different components of the tweet, and covers different types of sarcasm. The set of tweets on which we run our experiments is checked and annotated manually.

2.4.1 Data

Throughout the period ranging from December 2014 to March 2015, we collected tweets, using Twitter's streaming API. To collect sarcastic tweets, we queried the API for tweets containing the hashtag *"#sarcasm"*. Although Liebrecht et al. [70] concluded in their work that this hashtag is not the best way to collect sarcastic tweets, other works such as [77] highlighted the fact that this hashtag can be used for this purpose. However, they also concluded that the hashtag cannot be reliable and is used mainly for 3 purposes:

- to serve as a search anchor,
- to clarify the presence of sarcasm in a previous tweet, as in *"I forgot to add #sarcasm so people like you get it!"*,
- to serve as a sarcasm marker in case of a very subtle sarcasm where it is very hard to get the sarcasm without an explicit marker, as in *"Today was fun. The first time since weeks! #Sarcasm"*.

In total, we collected 58 609 tweets with the hashtag *"#sarcasm"*, which we cleaned up by removing the noisy and irrelevant ones, as well as ones where the use of the hashtag does fall into one of the two first uses of the three described above.

As for non-sarcastic tweets, we collected tweets dealing with different topics and made sure they have some emotional content.

We prepared 3 data sets for our work as follow:

- **Set 1:** this set contains 6000 tweets, half of them are sarcastic, and the other half are not. The tweets on this data set are manually checked and classified depending on their level of sarcasm from 1 (highly non-sarcastic) to 6 (highly sarcastic). The manual annotation is done by two people with no background about the tweets or the users who posted them. They have been asked to attribute the scores. It is important to note that the manual labelling is subject to the annotators' own opinion. Therefore, it is taken into account that the classification is not perfect. However, a sarcastic tweet is never labeled as non-sarcastic, and vice versa. Therefore, this set contains a trustworthy knowledge base that can be used to train our model. Tweets having level of sarcasm equal to 3 are mostly ones that, without the hashtag *"#sarcasm"*, are very close those of level 4 or 5. In other terms, it is very hard for a human, with no background about the tweet, to tell whether it is sarcastic or not. The hashtag *"#sarcasm"* has not been removed yet when the annotation is done. This first set is used to train our model. Therefore, in the rest of this work, it will be referred to as the *"training set"*. The number of sarcasm levels is also referred to as N_S and is equal to 6.

- **Set 2:** this set contains 1128 sarcastic tweets, and 1128 non-sarcastic ones. Sarcastic tweets are collected as described above (i.e., by querying Twitter API). Yet, no manual check is done, which makes it a very noisy data set. However, to reduce the noise, we filtered-out the non-english tweets, very short tweets (i.e., that have less than 3 words), and those which contain URLs. In most of the cases, URLs refer to photo links. We believe that part of the sarcasm is included in the photo, therefore we discard them. This data set is used during our experimenting process to optimize the parameters we defined for our features. In the rest of this work, we will refer to this set as the *"optimization set"*.

- **Set 3:** this set contains 500 sarcastic tweets, and 500 non-sarcastic ones. All tweets are manually checked and classified as sarcastic and non-sarcastic. This set will serve as a test set, and will be used to evaluate the performances of our proposed approach. Therefore, in the rest of this work, it will be referred to as the *"test set"*.

None of the tweets of any of the aforementioned sets is re-used in another. In addition, during our work, we removed the hashtag *"#sarcasm"* from all the tweets.

2.4.2 Tools

To perform the different Natural Language Processing (NLP) tasks (i.e., tokenisation, lemmatization, etc.), we used Apache OpenNLP[5]. However, OpenNLP PoS tagger performs poorly with the given model to tag tweets, due to the irrelevant content and the use of slangs, etc., we used Gate Twitter part-of-speech tagger [104]. This PoS-tagger reaches an accuracy of 90.5% on Twitter data.

To perform the classification, we used the toolkit weka [105] which presents a variety of classifiers. We used libsvm [106] to perform the classification using Support Vector Machine (SVM).

2.4.3 Features Extraction

Being a sophisticated form of speech, sarcasm is used for different purposes. While annotating the data, the annotators concluded that these purposes fall mostly, but not totally, in one of three categories: sarcasm as wit, sarcasm as whimper and sarcasm as avoidance.

- **Sarcasm as wit:** when used as a wit, sarcasm is used with the purpose of being funny; the person employs some special forms of speeches, tends to exaggerate, or uses a tone that is different from that when he talks usually to make it easy to recognize. In social networks, voice tones are converted into special forms of writing: use of capital letter words, exclamation and question marks, as well as some sarcasm-related emoticons.

- **Sarcasm as whimper:** when used as whimper, sarcasm is employed to show how annoyed or angry the person is. Therefore, it tempts to show how bad the situation is by using exaggeration or by employing very positive expressions to describe a negative situation.

[5]https://opennlp.apach.org

- **Sarcasm as evasion:** it refers to the situation when the person wants to avoid giving a clear answer, thus, makes use of sarcasm. In this case, the person employs complicated sentences, uncommon words and some unusual expressions.

Unlike [107], which classifies sarcasm into 4 different types based on how sentiments appear in the text, the observations and classification are done based on why sarcasm is used. Although theses observations are likely to be biased and depend on the annotator's own opinions, we rely on these assumptions to build our model. During our work, we rely mainly on writing patterns to detect sarcastic statements; however, other features are extracted and that help to obtain higher classification precision and accuracy. The distinction of purposes highlights the use of some features as we will describe next.

Four families of features are extracted: sentiment-related features, punctuation-related features, syntactic and semantic features, and pattern features.

Sentiment-related Features

A very popular type of sarcasm that is widely used in both regular conversations as well as short messages such as tweets, is when an emotionally positive expression is used in a negative context. A similar way to express sarcasm is to use expressions having contradictory sentiments. This type of sarcasm which we qualified as "whimper" is very common in social networks and microblogging websites. Riloff et al. [96] show that this type of sarcasm can be identified and detected when a positive statement, usually a verb or a phrasal verb, is collocated with a negative situation (e.g., *"I love being ignored all the time"*). They built a lexicon-based approach that learns the possible positive expressions and negative situations and used it to detect such contrast in unknown tweets. However, learning all possible negative situations requires a big and rich source and might be infeasible because negative situations are unpredictable.

In our work, we opt for a more straight-forward, yet more general approach. We consider any kind of inconsistency between sentiments of words as well as other components within the tweet. Therefore, to identify and quantify such inconsistency, we extract sentimental components of the tweet and count them. For this purpose, we maintain two lists of words qualified as "positive words" and "negative words". The two lists contain respectively words that have positive emotional content (e.g., *"love"*, *"happy"*, etc.) and negative emotional content (e.g., *"hate"*, *"sad"*, etc.). The two lists of words are created using SentiStrength [6] database. This database contains a list of emotional words, where negative words have scores varying from -1 (almost negative) to -5 (extremely negative) and positive words have score varying from 1 (almost positive) to 5 (extremely positive). Using these two lists, we extract two features we denote respectively pw and nw by counting the number of positive and negative words in the tweet.

Adjectives, verbs and adverbs have higher emotional content than nouns [108]; therefore positive and negative words that have the associated PoS-tag, shown in TABLE 2.1, are counted again and used to create two more features that we denote PW and NW and which represent the number of highly emotional positive words and highly emotional negative words.

[6] http://sentistrength.wlv.ac.uk

Table 2.1: PoS-Tags for Words Considered as Highly Emotional

Part of Speech	**Part of Speech Tag**
Adjectives	"JJ", "JJR", "JJS"
Adverbs	"RB", "RBR", "RBS"
Verbs	"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"

We then add three more features by counting the number of positive, negative and sarcastic emoticons. Sarcastic emoticons are emoticons used sometimes with sarcastic or ironical statements (e.g., *":P"*). These emoticons are used sometimes when the person is trying to be funny or to show that he is just making a joke (i.e., when sarcasm is used as wit).

Hashtags also have emotional content. In some cases, they are used to disambiguate the real intention of the Twitter user conveyed in his message. For example, the hashtag employed in the following tweet: *"Thank you very much for being there for me #ihateyou"* tells that the user does not really want to thank the addressee, he was rather blaming him for not being there for him. Therefore, we count also the number of positive and negative hashtags.

In addition to the aforementioned features, we extract features related to the contrast between these sentimental components. We first calculate the ratio of emotional words $\rho(t)$ defined as

$$\rho(t) = \frac{(\delta \cdot PW + pw) - (\delta \cdot NW + nw)}{(\delta \cdot PW + pw) + (\delta \cdot NW + nw)} \tag{2.1}$$

where *t* is the tweet, *pw*, *PW*, *nw* and *NW* denote respectively the number of positive words (other than highly emotional ones), that of highly emotional positive words, that of negative words (other than highly emotional ones) and that of highly emotional words. δ is a weight bigger than 1 given to the highly emotional words. In case the tweet does not contain any emotional word, ρ is set to 0. In the rest of this work, δ is set to 3.

We then define 4 features that represent whether there is a contrast between the different components. By contrast we mean the coexistence of a negative component and a positive one within the same tweet. We check the existence of such contrast between words, between hashtags, between words and hashtags and between words and emoticons and use these information as extra features. The final sentiment-related feature vector has 14 features.

Punctuation-Related Features

Sentiment-related features are not enough to detect all kinds of sarcasm that might be present. In addition, they do not make use of all the components of the tweet. Therefore, more features are to be extracted. As mentioned before, sarcasm is a sophisticated form of speech: not only it plays with words and meanings, but also it employs behavioral aspects such as low tones [110, 111], facial gestures [112] or exaggeration. These aspects translate to a certain use of punctuation or repetition of vowels when the message is written. To detect such aspects, we extract a set of features that we qualify as punctuation-related features. For each tweet, we calculate the following values:

- The number of exclamation marks,

- The number of question marks,
- The number of dots,
- The number of all-capital words, and
- The number of quotes.

We also add a sixth feature by checking if any of the words contains a vowel that is repeated more than twice (e.g., *"looooove"*). If such a word exists, the feature value is set to *"true"*, otherwise, it is set to *"false"*.

The excessiv use of exclamation marks or question marks, or the repetition of a vowel, particularly in an emotional word, might reflect a certain tone that the user intends to show; however, this tone is not always sarcastic. We believe that these features can be highly correlated with the number of words in the tweet. Some very short tweets which end with many exclamation marks might show surprise rather than sarcasm. Following two examples of tweets in which the use of exclamation marks has two different use cases:

- *"Thank you @laur3en, it was amazing !!!"*
- *"Thanks for another amazing day with your amazing boyfriend!!!!"*

In the first case, the exclamation marks are used to show sincere feelings of gratitude. However, in the second, the exclamation marks serve as an indication of annoyance; the user has no real intension to thank his friend. Although the use of exclamation is not relevant in itself and might not show whether the user is expressing sarcasm or any other emotion; combined with other features, this feature is expected to add value to the classification. We then define one last feature by counting the number of words in the tweet. In total, 7 punctuation-based features are extracted.

Syntactic and Semantic Features

Along with the punctuation-related features, some common expressions are used usually in a sarcastic context. It is possible to correlate these expressions with the punctuation to decide whether what is said is sarcastic or not. Besides, in other cases, people tend to make complicated sentences or use uncommon words to make it ambiguous to the listener/reader to get a clear answer. This is common when sarcasm is used as "evasion", where the person's purpose is to hide his real feeling or opinion by using sarcasm. Hence, we extract the following features that reflects these aspects:

- The use of uncommon words,
- The number of uncommon words,
- The existence of common sarcastic expressions,
- The number of interjections, and
- The number of laughing expressions.

Table 2.2: Expressions Used to Replace the Words of GFI

PoS-tag	**Expression**
"CD"	[CARDINAL]
"FW"	[FOREIGNWORD]
"UH"	[INTERJECTION]
"LS"	[LISTMARKER]
"NN", "NNS", "NNP", "NNPS",	[NOUN]
"PRP", "PRP$"	[INTERJECTION]
"MD"	[MODAL]
"PB", "RBR", "RBS"	[ADVERB]
"WDT", "WP", "WP$", "WRB"	[WHDETERMINER]
"SYM"	[SYMBOL]

In particular, the feature "Existence of common sarcastic expression" is extracted in the same way we extract the features qualified as *"pattern-related"* (this will be described in detail in the next subsection). Here we used a noisy set of 3000 tweets having the hashtag *"#sarcasm"* (the set has been discarded later and has not been used neither for training nor for test). We extracted all possible patterns of length varying from 3 to 6, we selected the patterns that appeared more than 10 times. Being few in number, we manually checked the list and removed the irrelevant ones. We obtained a list of 13 main patterns including [*love* PRONOUN *when*] (e.g., *"I love it when I am called at 4 a.m. because my neighbour's kid can't sleep!"*), [PRONOUN *be* ADVERB *funny*] (e.g., *"You are incredibly funny -_-"*), etc.

Pattern-Related Features

The patterns selected in the previous subsection, and qualified of *"common sarcastic expression"* are very common, even in spoken language. However, their number is small, they are not unique and most of the tweets in both our training and test sets do not contain them. That being the case, we dig further and extract another set of features. The idea of our pattern-related features is inspired from the work of Davidov et al. [77]. In his approach, the author classified words into two categories: high-frequency words and content words based on their frequency of appearance in his data set and defined a pattern as an *"ordered sequence of high frequency words and slots for content words"*. This approach, although it has some potential to detect sarcasm, presents many shortcomings as shown in Section 2.3.

Therefore, we propose more efficient and reliable patterns. We divide words into two classes: a first one referred to as *"CI"* containing words of which the content is important and a second one referred to as *"GFI"* containing the words of which the grammatical function is more important. If a word belongs to the first category, it is lemmatized; otherwise, it is replaced it by a certain expression. The expressions used to replace these words are shown in TABLE 2.2. The classification into classes is done based on the part of speech tag of the word in the tweet. The list of part-of-speech tags, their meaning and to which category we classify them is given in TABLE 2.3.

Table 2.3: Part-of-Speech Tag Classes

POS Tag	Description	Class
CC	coordinating conjunction	CI
CD	cardinal number	GFI
DT	determiner	CI
EX	existential there	CI
FW	foreign word	GFI
IN	prep./sub. conjunction	CI
JJ	adjective	CI
JJR	adjective, comparative	CI
JJS	adjective, superlative	CI
LS	list marker	GFI
MD	modal	GFI
NN	noun, singular or mass	GFI
NNS	noun plural	GFI
NNP	proper noun, singular	GFI
NNPS	proper noun, plural	GFI
PDT	predeterminer	CI
POS	possessive ending	CI
PRP	personal pronoun	GFI
PRP$	possessive pronoun	GFI
RB	adverb	CI
RBR	adverb, comparative	CI
RBS	adverb, superlative	CI
RP	particle	CI
SYM	Symbol	GFI
TO	to	CI
UH	interjection	GFI
VB	verb, base form	CI
VBD	verb, past tense	CI
VBG	verb, gerund/present participle	CI
VBN	verb, past participle	CI
VBP	verb, sing. present, non-3d	CI
VBZ	verb, 3rd person sing. present	CI
WDT	wh-determiner	GFI
WP	wh-pronoun	GFI
WP$	possessive wh-pronoun	GFI
WRB	wh-abverb	GFI

We generate the vector of words for each tweet according to the rule defined. For example, the following PoS-tagged tweet *"@gilbert:_NN you_PRP are_VBP crazy_JJ ,_, who_WP told_VBD you_PRP I_PRP want_VBP to_TO drink_VB with_IN you_PRP !!!!_."* gives, the following pattern vector [NOUN PRONOUN *be crazy who tell* PRONOUN PRONOUN *want to drink with* PRONOUN.]

We define a pattern as an ordered sequence of words. The patterns are extracted from the training set and are taken such as their length satisfies

$$L_{Min} \leq Length(pattern) \leq L_{Max} \tag{2.2}$$

where L_{Min} and L_{Max} represent the minimal and maximal allowed length of patterns in $words$ and $Length(pattern)$ is the length of the pattern in $words$. The number of pattern lengths is $N_L = (L_{Max} - L_{Min} + 1)$. Therefore, from the example mentioned above, we can extract the following patterns:

- [NOUN PRONOUN *be crazy*],
- [PRONOUN *be crazy*],
- [*be crazy who tell* PRONOUN PRONOUN *want to*],
- etc.

Only patterns that appear at least N_{occ} times in our training set are kept; the others are discarded. In the rest of this work, N_{occ} is set to 2: the value 1 gives lower accuracy and precision and higher values decrease remarkably the number of patterns, and consequently presents lower accuracy. In addition, a pattern that appears in a sarcastic tweet and in a non-sarcastic tweet is discarded. This step is done to filter out patterns that are not related to sarcasm. After the selection, we divide the resulted patterns into N_F sets, where

$$N_F = N_L \times N_S. \tag{2.3}$$

We create N_F features, as shown in TABLE 2.4. Each feature F_{ij} of the table represents the degree of resemblance of the tweet to the patterns of degree of sarcasm i and length j. Therefore, given a tweet t, we calculate the resemblance degree $res(p, t)$ of each pattern in the training set p to the tweet t, defined as:

$$res(p,t) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order,} \\ \alpha \cdot n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appears in the tweet.} \end{cases}$$

Given N_{ij} the number of patterns collected from the training set having a sarcasm degree i and a length j, we focus, among them, on the K patterns $(p_1, \cdots, p_k)$ that resemble the tweet the

Table 2.4: Pattern Features

		Pattern length			
		L_1	L_2	$\cdots$	L_N
Sarcasm level	1	F_{11}	F_{12}	$\cdots$	F_{1N}
	2	F_{21}	F_{22}	$\cdots$	F_{2N}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	6	F_{61}	F_{62}	$\cdots$	F_{6N}

most. The value of the feature F_{ij} is

$$F_{ij} = \beta_j * \sum_{k=1}^{K} res(p_k, t) \tag{2.4}$$

where β_j is a weight given to patterns of length L_j (regardless of their level of sarcasm). We give different weights for each length of pattern since longer patterns are more likely to have higher impact. F_{ij} as defined measures the degree of resemblance of a tweet *t* to patterns of level of sarcasm *i* and length j. K in our work is set to 5, and represents the K closest patterns among the N_{ij} ones described above.

Extension of the training set patterns: Being relatively small in size (i.e., only 6000 tweets), our training set cannot cover all possible sarcastic patterns. Therefore, we enrich it to obtain more patterns. We collected 18 959 tweets containing the hashtag *"#sarcasm"* and 18 959 tweets that do not. We checked if the tweets having the hashtag *"#sarcasm"* contain any of the sarcastic patterns we already extracted from the training set and that have a length equal to or more than 4. If that is the case, we extract the different patterns from the tweet and add them to the list of patterns of the training set keeping in mind the rule we made for the selection of patterns (i.e., if the pattern exists in a non-sarcastic tweet, it is discarded). Although the added tweets are not as reliable as those of the initial training set, we believe that filtering the tweets that contain at least one pattern that is identical to a reliable one is reliable enough given it already contains the hashtag *"#sarcasm"*. We then did the same to the non sarcastic tweet. Thus, we enriched our data set with more patterns. This step has been done only to get more patterns, therefore, none of the other families of features is concerned by the enrichment.

Pattern-related features as defined give a high flexibility to optimize depending on their contribution. In total we have the following parameters to optimize:

- L_{Min} and L_{Max}
- α
- $\beta_1, \cdots, \beta_{N_L}$

To optimize L_{Min} and L_{Max}, we fixed α and β_i ($i = 1, N_L$) as follow and tried different values of pattern lengths:

$$\begin{cases} \alpha & = 0.1, \\ \beta_1 = \cdots = \beta_{N_L} & = 1.0. \end{cases}$$

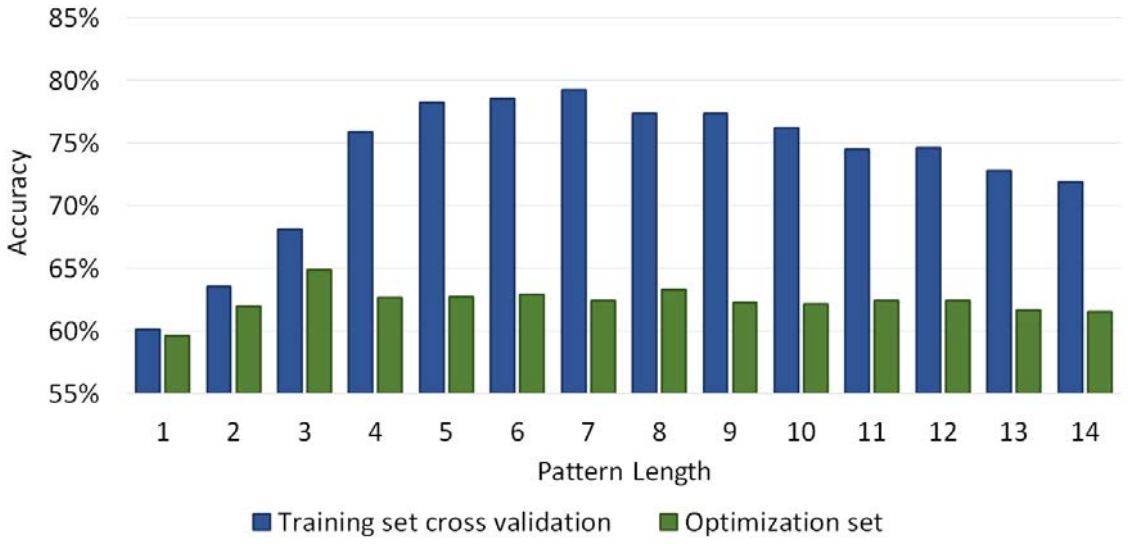

Figure 2.1: Accuracy per Pattern Length for Fixed Values of $\alpha, \beta_1, \cdots, \beta_{N_L}$

We ran a first simulation on our training set (6000 tweets) and optimization set (2256 tweets), for each pattern length. We obtained the results shown in Fig. 2.1. The results present the accuracy of the classification of tweets as sarcastic and non-sarcastic. The obtained results show that the patterns having a length are from 4 to 10 give the highest accuracy (i.e., more than 75% accuracy during 10-folds cross validation). Pattern length 3 gives the highest accuracy on our optimization set. Given that the average number of words per tweet is equal to 11.48, we set the parameters L_{Min} and L_{Max} respectively to 3 and 10.

Afterwards, we set L_{Min} and L_{Max} as mentioned, kept the values of $\beta_1, \cdots, \beta_{N_L}$ as they are (i.e., equal to 1). We tried different values of α. We ran different simulations on the same data sets using pattern features, for different values of α. Results of the test are given in Fig. 2.2.

The accuracy of classification varies highly depending on the value of α, that is, the lower the value is, the better the performances are during the cross validation. This is due to the unicity of the patterns. In other terms, in the training set, the patterns derived from each tweet will have the highest score. Thus, the tweet will be classified as the closest to its own patterns. However, in the optimization set, the accuracy is the highest when $\alpha \in \{0.01, 0.1\}$. The highest accuracy we obtained was for $\alpha = 0.03$ as shown in Fig. 2.2

Finally, for $\beta_1, \cdots, \beta_{N_L}$, we tried different combinations maintaining the following condition

$$\beta_1 \leq \cdots \leq \beta_{N_L}. \tag{2.5}$$

The observed results are not very different for all the combinations we tried although we noticed that the closer the values to 1, the better the performances are. The best performances we obtained were observed when

$$\beta_n = \frac{n-1}{n+1}. \tag{2.6}$$

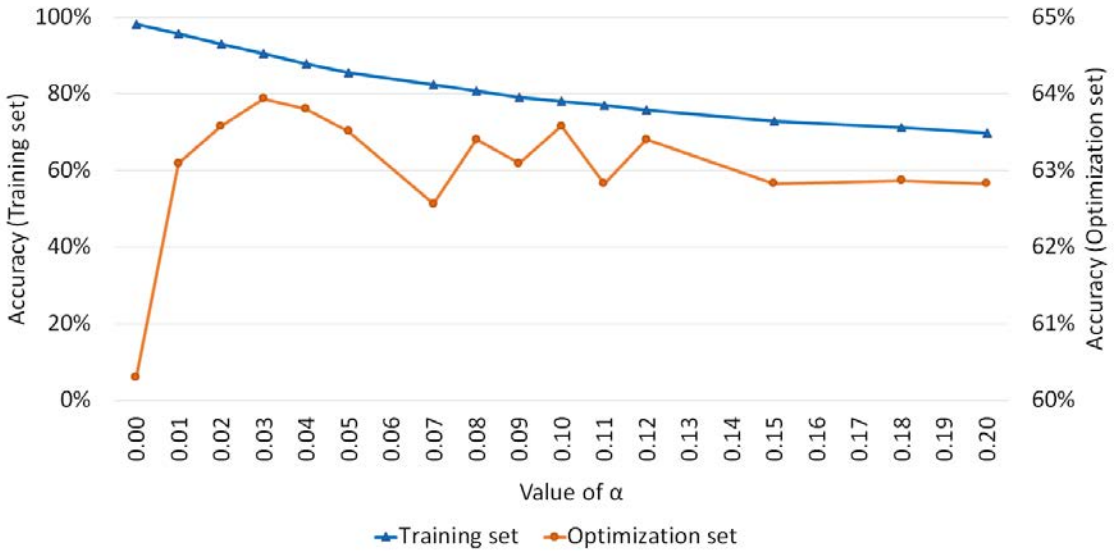

Figure 2.2: Accuracy of Classification for Different Values of α

The final values of parameters we set for pattern-related features are as follow:

$$\begin{cases} N_{occ} &= 2, \\ L_{Min} &= 3, \\ L_{Max} &= 10, \\ \alpha &= 0.03, \\ \beta_n &= (n-1)/(n+1) \qquad \forall n \in \{3, \ldots, 10\}. \end{cases}$$

In the next section, we evaluate the model we built and present the results of our experiments.

2.5 Experimental Results

Once the features are extracted, we proceed to our experiments. The Key Performance Indicators (KPIs) used to evaluate the approach are:

- **Accuracy:** it represents the overall correctness of classification. In other words, it measures the fraction of all correctly classified instances over the total number of instances.
- **Precision:** it represents the fraction of retrieved sarcastic tweets that are relevant. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of tweets classified as sarcastic.
- **Recall:** it represents the fraction of relevant sarcastic tweets that are retrieved. In other words, it measures the number of tweets that have successfully been classified as sarcastic over the total number of sarcastic tweets.

We ran the classification using the classifiers *"Random Forest"* [109], "Support Vector Machine" (SVM), "k Nearest Neighbours" (k-NN) and "Maximum Entropy". Table 2.5 presents the performances of the classifiers on the dataset.

Table 2.5: Accuracy, Precision, Recall and F1-Score of Classification Using Different Classifiers

	Overall Acc.	Precision	Recall	F1-Score
Rand. Forest	83.1%	91.1%	73.4%	81.3%
SVM	60.0%	98.1%	20.4%	33.8%
k-NN	81.5%	88.9%	72.0%	79.6%
Max. Ent.	77.4%	84.6%	67.0%	74.8%

The overall accuracy obtained reaches 83.1% using the classifier Random Forest for an F1-score equal to 81.3%. This accuracy is obtained when setting the parameters of the classifier as follows [109]:

- Number of Features: 20
- Number of Trees: 100
- Seeds: 20
- Max Depth: 0 (unlimited)

SVM, on the other hand, presents a precision equal to 98.1% for a low F1-score equal to 33.8%. This means that most of the tweets that were classified as sarcastic are indeed sarcastic. However, a very few percentage of the sarcastic tweets were detected (almost 20%). In other words, SVM is capable of detecting sarcasm with a high precision and the output can indeed be used to refine sentiment analysis, however, it does not cover all the sarcastic tweets. In a real stream of tweets, the number of sarcastic tweets is quite lower than that in the dataset used; therefore, the results obtained mean that only one out of five sarcastic tweets will be detected. Classifiers such as k-NN and Maximum Entropy present a high accuracy and F1-scores, however, the results using Random Forest are the highest. During the preliminary experiments (i.e., parameters optimization) as well as for the rest of our analysis, the results used are those returned by the classifier Random Forest.

2.5.1 Performances of Each Set of Features

We first checked the performances of classification of each set of features apart. Figs. 2.3 and 2.4 present the performances of the different sets of features.

During cross-validation

Fig. 2.3 shows the performances of classification during cross-validation. We notice that the performances of the pattern-related features is very high during cross-validation. This has been discussed in the previous section: the value of α as chosen makes each tweet in the training set the closest to itself. This explains the very good results obtained by Davidov et al. [77].

On the other hand, we notice that the syntax-related features present a very low accuracy and recall. The features seem to be not very efficient, if used alone, to classify the tweets as sarcastic and non-sarcastic. One reason is the low presence of these features in the data set. TABLE 2.6 shows the existence rate of each of the features in the training set. In addition, due to the informal language used in Twitter and the noise it has, the PoS-tagger performances are lower than when

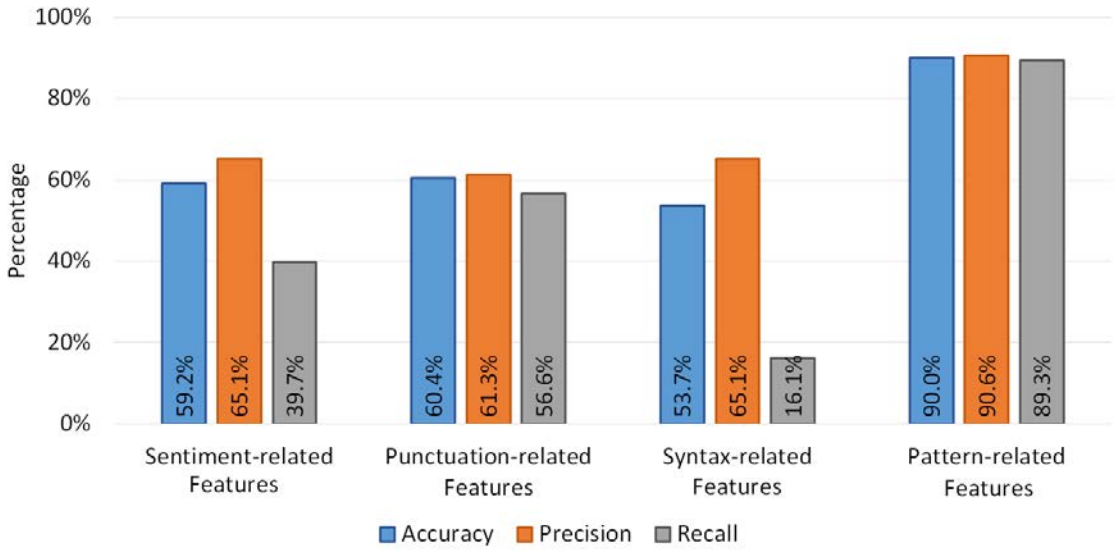

Figure 2.3: Accuracy of Classification During Cross-Validation for each Family of Features

Table 2.6: Ratio of Presence of Syntax-Related Features in the Training Set

	True	False	Ratio
Presence of uncommon words	243	5757	4.05%
Presence of common sarcastic patterns	115	5885	1.92%
Presence of interjections	410	5590	6.83%
Presence of laughters	224	5776	3.73%

applied to a formal text. In particular, the PoS-tagger is not very efficient to detect interjections, it classifies them in many cases as nouns. However, the precision given by this set of features, and which exceeded 65% shows the importance of such features to detect sarcastic components. It refers to the number of sarcastic tweets over the number of tweets judged as sarcastic. Although, they perform poorly, these features might have higher added value when correlated with other features, or if their presence is more frequent.

Punctuation-related features and sentiment-related features have higher prediction rate. They are more efficient, though they perform worse than pattern-related features. They both give an accuracy almost equal to 60%. Furthermore, the precision of sentiment-based features is remarkably higher than the accuracy. In other terms, from the tweets that have been classified as sarcastic, the prediction rate is high. This can be explained by the fact that tweets having contrasting emotional content are likely to be sarcastic. Thus, if detected, they would be classified as sarcastic.

On a test set

Fig. 2.4 shows the performances of classification on our test set. Performance of the classification on unknown data is clearly lower than that during cross-validation. However, we can notice that the sets of features that have the highest merit during cross-validation are the same ones that have the highest merit during the classification of test set tweets.

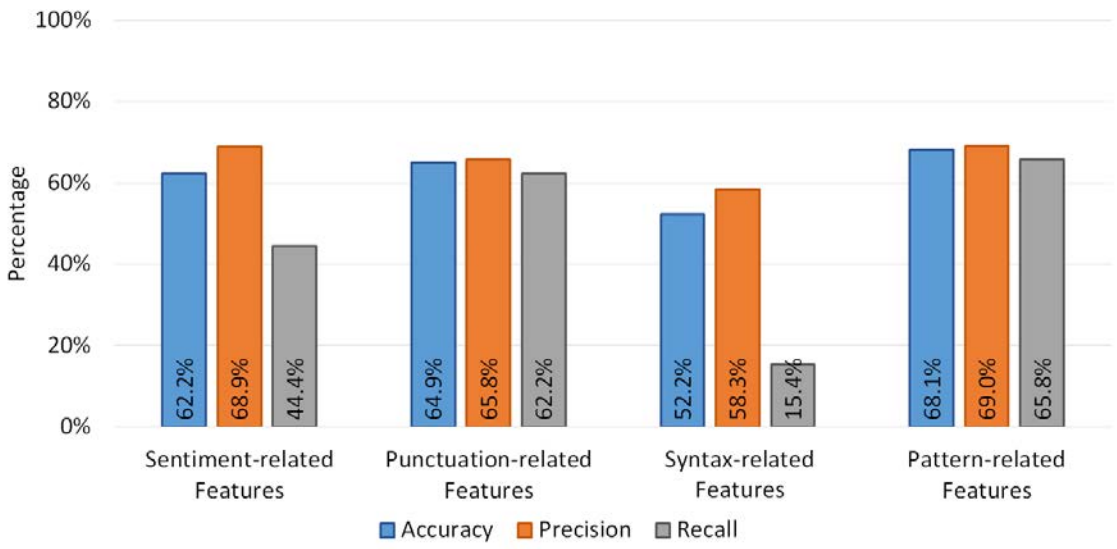

Figure 2.4: Accuracy of Classification of the Test Set for each Family of Features

The low accuracy of syntax-related features is due to their low presence in the test set too. As for Pattern-related features, they have higher performances. Accuracy and precision have very close values. This can be explained by the fact that, contrarily to sentiment-based features for example, which check the existence on some characteristics related to sarcasm in the tweets, patterns are extracted from both sarcastic and non-sarcastic tweets, and the closeness to these patterns is checked.

2.5.2 Overall Performances of the Proposed Approach

Together, the features perform better than each one by itself. Fig. 2.5 shows the performance of the proposed approach when all the features are used.

During cross validation, both the accuracy and precision are higher than 90%. The recall is lower than 89%. More interestingly, the accuracy obtained for the test set, before enrichment of the patterns, exceeds 72% with a precision higher than 73%. This shows that, if combined, the different sets of features, perform better. Although our data set contains many sarcastic tweets that are hard to identify even by humans (we referred to the hashtag *"#sarcasm"* to classify them), the accuracy obtained is high. The enrichment process added more potential to the approach and increased the accuracy of the classification noticeably. The precision also increased compared to that without enrichment. It reflects the fact that most of the tweets that have been classified as sarcastic really are. Recall, on the other hand, has a lower value, though still better than before enrichment. It shows that, many of the sarcastic tweets were not well classified. As mentioned before, tweets of sarcasm level 3 are very difficult to be distinguished from the non sarcastic ones, therefore, we believe that many of the sarcastic tweets that were not classified as sarcastic fall in this category. Nevertheless, this can be enhanced if we use more tweets for enrichment or in the training set.

To measure the potential of our method, we consider the approach proposed by Riloff et al. [96] as well as the n-gram-based approaches as our baseline. In addition to the aforementioned

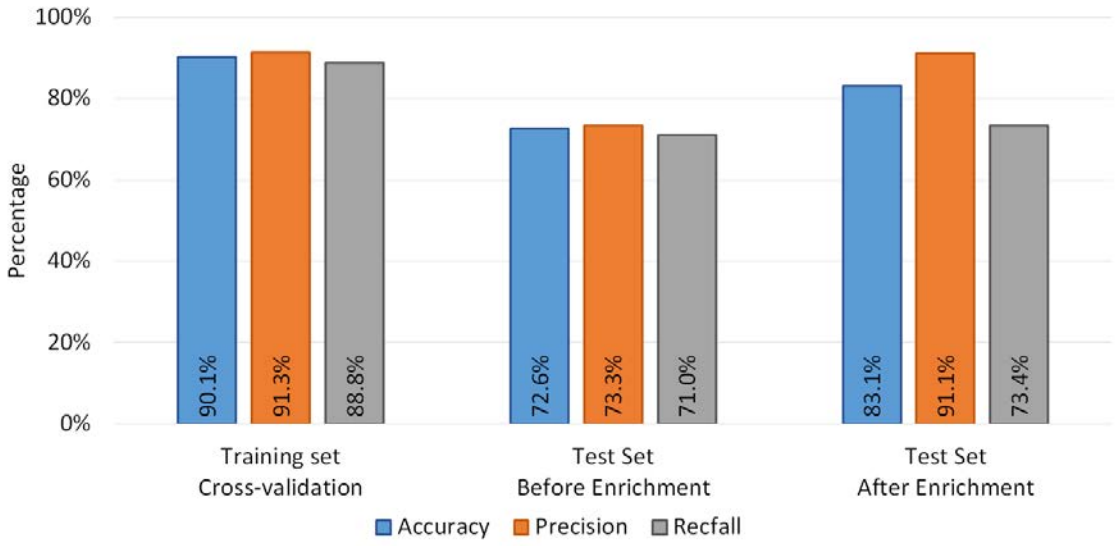

Figure 2.5: Accuracy of Classification Using all Features During Training Set-Cross-Validation and on the Test Set

KPIs, we define a fourth one, which is the F_1 score defined as follow:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2.7}$$

It combines the precision and recall, therefore it represents a more reliable KPI to compare different approaches.

The results of the comparison of our approach with the baseline ones are given by TABLE 2.7. Our proposed approach clearly outperforms the baseline ones, for the used data set: not only it has a higher accuracy and precision, our method's F_1 score is neatly higher than that of the baseline ones.

To begin with, although it performs well when detecting a specific type of sarcasm, the approach proposed by Riloff et al. [96], performs poorly in our data set since most of the sarcastic tweets do not fall in the type of sarcasm where a positive sentiment contrasts with a negative situation. This explains the high precision of that approach and its low recall. In other words, if judged as sarcastic using this approach, a tweet is very likely to be indeed sarcastic. However, judging one as such is less likely to happen. To recall, the approach proposed by Riloff et al. [96] starts with a single seed word "love", build up two dictionaries of positive verbs and negative situations, and uses these dictionaries to identify the case of sarcasm where a positive verb co-occur with a negative situation. This is obviously not the only way to express sarcasm, thus the low recall.

The n-gram approach on the other hand lacks behind for more obvious reasons: while it is always possible to identify sentimental sentences (i.e., positive or negative ones) by relying on individual words or expressions composed of 2 or 3 words, it is very difficult to identify sarcastic statements just be relying on such limited amount of words. Sarcasm detection requires an understanding of the entire sentence or at least a longer expression.

Compared to more sophisticated approaches such as that proposed by Davidov et al. [77] or Rajadesingan et al. [78], our approach, although it does not require a big training data set, or

Table 2.7: Performance of the Proposed Approach Compared to the Baseline Ones

	Accuracy	**Precision**	**Recall**	**F-Score**
n-grams	65.9%	82.2%	40.6%	65.9%
Riloff et al. [96]	59.4%	65.0%	40.8%	50.1%
Proposed approach	**83.1%**	**91.1%**	**73.4%**	**81.3%**

a knowledge base of the users, presents competitive results. The two approaches were not re-implemented and run on our data set for the reason that we do not have a previous knowledge of the users as in [78], nor do we dispose of 5.9 million tweets to classify words into context words and highly frequent words as in [77]. However, our proposed presents an F1 score close to that of the approach [77] which is 82.7% (on the Twitter data set) and an accuracy close to that of [78] which is 83.46%.

2.6 Use of Sarcasm to Enhance Sentiment Analysis Performance

Given a set of tweets dealing with different topics, we want to deduce for each one of them if it deals "positively" or "negatively" with its theme. Therefore, from each tweet, we extract a set of features, refer to a training set and use machine learning algorithms to perform the classification.

2.6.1 Data

The tweets are collected from a big Twitter dataset, publicly available to be used in academia[7]. We selected a collection of tweets "classifiable" by humans to positive or negative. Tweets which are irrelevant or emotionless are discarded. We then manually annotated them into "positive" and "negative". Tweets are also selected in a way to belong to one of the following topics: politics, phone reviews, sports, movie reviews and electronic products (other than phones) reviews. We added an extra topic we called "general" for tweets that do not belong to any of the aforementioned topics.

We created two datasets as described: a set of 20 000 tweets for training, and a set of 1200 tweets for test. TABLE 2.8 shows how tweets are split into training and test sets. For the purpose of our study, we selected 5% of the tweets in a way to be sarcastic.

2.6.2 Features Used

Sarcasm, despite its definition, does not always mean that what is said is the opposite of what is meant. Therefore we should not assume the polarity of the tweet automatically to the opposite when sarcasm is employed. Therefore, we extract the following features as described in the previous section:

- The contrast features described in Section 2.4,
- The presence of a repeated vowel in an emotional word,

[7] http://help.sentiment140.com/for-students

Table 2.8: Structure of the Dataset Used

	Positive		Negative	
Topic	Training	Test	Training	Test
General	5165	120	5165	120
Sports	670	120	670	120
Phone reviews	815	120	815	120
Movie reviews	1405	96	1405	96
Electronic products reviews	1371	96	1371	96
Politics	574	48	574	48
Total	10 000	600	10 000	600

- The use of uncommon words,
- The number of laughters,
- The number of interjections, and
- The existence of sarcastic patterns.

These features have been added to a set of features we proposed in a previous work [113] to perform sentiment analysis. The features introduced in [113] are the following:

- The number of positive words and that of negative ones,
- The number of highly emotional positive words and that of highly emotional negative ones,
- The ratio of emotional words,
- The number of positive hashtags and that of negative ones,
- The number of positive emotions, that of negative ones and that of neutral ones,
- The number of question marks and that of exclamation marks.

The total number of features used is 23. We run our experiments using the new set of features and compare the results of this approach to the one where sarcasm-related features are not included.

2.6.3 Experiment Results

Classification is conducted using Naive Bayes, SVM, and Maximum Entropy algorithms. Table. 2.9 shows the accuracy of sentiment classification before and after taking sarcasm-related features into consideration. The results show a noticeable enhancement after taking the sarcasm into consideration. Albeit the low number of sarcastic tweets in our test set (i.e., less than 5%), our approach helped enhance the results.

In addition, since most of the sarcastic tweets are basically negative tweets that have been classified as positive, we focus on the recall of negative tweets. We compared the recall before and after taking sarcasm into consideration. TABLE 2.10 shows that the recall has noticeably increased after taking sarcasm into consideration. In other words, many of the tweets, previously classified wrongly as positive are now well classified.

Table 2.9: Accuracy of Sentiment Analysis Before and After Adding Sarcasm-Related Features

Classifier	Naive Bayes	SVM	Max Entropy
Before	82.94%	83.67%	82.45%
After	84.92%	87.00%	83.7%

Table 2.10: Recall of Negative Tweets Before and After Adding Sarcasm-Related Features

Classifier	Naive Bayes	SVM	Max Entropy
Before	83.9%	85.7%	82.3%
After	85.9%	92.0%	83.8%

2.6.4 Discussion

Presence of sarcasm has always been one of the main misclassification reasons. Throughout this section, we have demonstrated that is always possible to enhance sentiment analysis accuracy just by identifying sarcastic statements. Despite being fast, the identification of sarcastic statements prior to sentiment analysis might not be very practical. It is always a better option to evaluate how much sarcasm is present in the data set and decide whether or not to add this extra step.

In addition to sarcasm, several other misclassification reasons exist, namely the absence of sentiment indicators, the context-dependency, etc. In such case, if the individual pieces of text don't matter, and the overall accuracy is what matters, it is always possible to use other techniques to get an accurate overview of the opinion of users. For example, it is possible to use several small test sets, run the classification (regardless of how good the results are), compare the results to the expected ones, and learn how to adjust the ratio of the different classes. This allows, when performing the classification on unknown data sets to evaluate accurately the distribution of the different classes. This process is referred to in several works as "quantification", however, it is out of the scope of the current thesis.

2.7 Conclusion

In this chapter, we described our method to detect sarcasm in Twitter. The proposed method makes use of the different components of the tweet. Our approach makes use of Part-of-Speech-tags to extract patterns characterizing the level of sarcasm of tweets. The approach has shown good results, though might have even better results if we use a bigger training set since the patterns we extracted from the current one might not cover all possible sarcastic patterns.

We also proposed a more efficient way to enrich our set with more sarcastic patterns using an initial training set of 6000 Tweets, and the hashtag *"#sarcasm"*.

We then have demonstrated the importance of detection of sarcastic statements to enhance sentiment analysis and opinion mining: we proposed a method to detect sarcastic tweets, and proved how the recognition of sarcastic tweets helps boosting the sentiment classification.

Chapter 3

Multi-Class Sentiment Analysis

3.1 Introduction

Twitter, as well as other Online Social Networks (OSN) and microblogging websites became literally the biggest web destinations for people to communicate with each other, to express their thoughts about products [114] [58] or movies [59], share their daily experience and communicate their opinion about real-time and upcoming events, such as sports or political elections [68], etc.

While new platforms such as Snapchat[1] focused on video- and multimedia-based communication, Twitter, for its properties that we have introduced in Chapter 1, remains a ver interesting subject of data mining. Thanks to these properties, this ecosystem presents a very rich, source of data to mine. However, due to the limitation in terms of characters (i.e. 140 characters per tweet), mining such data present lower performance than that when mining longer texts. In addition, classification into multiple classes remains a challenging task: binary classification of a text usually relies on the sentiment polarity of its components (i.e., whether they are positive or negative); whereas, when positive and negative classes are divided into subclasses, the accuracy tends to decrease remarkably.

In this chapter, we propose an approach that relies on writing patterns, and special unigrams to classify tweets into 7 different classes, and demonstrate how the proposed approach presents good performances (i.e., classification accuracy and precision). The main contributions present in this chapter are as follows:

1. We introduce SENTA (SENTiment Analyzer), a user-friendly tool that allows the extraction of a wide set of features from texts that cover both the content and the form,

2. We introduce, in addition to some conventional features, writing pattern-related features to help enhance the accuracy of classification,

3. We use SENTA to extract a set of features to classify tweets into 7 different sentiment classes.

The remainder of this chapter is structured as follows: In Section 3.2 we present our motivations for this work and in Section 3.3 we describe some of the related work. In Section 3.4, we present SENTA, our tool to extract different features from tweets, and that we will use later on to perform the multi-class classification. In Section 3.5 we describe in details the proposed method. In Section 3.6 we detail our experiments and the results obtained. Section 3.7 concludes this chapter and proposes possible directions for future work.

3.2 Motivations

3.2.1 Why Multi-Class Sentiment Analysis?

Social networks and microblogging websites such as Twitter have been the subject to many studies in the recent few years. Automatic sentiment analysis and opinion mining present a hot topic of study. Social networks present a huge source of data representing the opinions of a significant, yet totally random, proportion of users and customers who are using a product of a service. However,

[1] https://www.snapchat.com

due to the informal language used, the presence of non-textual content and the use of slang words and abbreviations, classification of data extracted from such microblogging websites is rather a challenging task. Ghag et al. [115] defines *"Hidden Sentiment Identification"* which is the identification of the real feeling rather than the sentiment polarity, *"Handling Polysemy"* which is the existence of multiple meanings that might have different sentiment polarity for the same word, and *"Mapping Slangs"* which is the identification of the meaning and the polarity of slang words, among others as the most challenging tasks facing the sentiment analysis of short microblog texts.

On a related context, the state of the art proposed approaches are mostly focusing on the binary and ternary sentiment classification. In other words, they classify texts either into "positive" and "negative", or into "positive", "negative" and "neutral". However, to study the opinion of a user, it would be more interesting to go deeper in the classification, and detect the sentiment hidden behind his post. Following two examples of tweets which are negative, however, reflect two completely different aspects:

- *"Damn damn.. no iPhone support for windows XP x64. There are some workarounds, but I can't figure this out."*

- *"Noooooooooo! My iPhone glass cracked :("*

In the first example, the user is expressing his fury towards the absence of support of his phone on an operating system. However, in the second he is expressing some feeling of sadness because of a physical problem his phone faced. The first example shows some important information regarding the satisfaction of the user, therefore, it might be more important to study. However, in general, both information can be used, yet, they have to be distinguished from each other.

3.2.2 The Need for an Open-Source Tool for Feature Extraction from Tweets

Nowadays, a variety of tools such as LIWC [116] offer the option to extract advanced features for different languages from texts, most of these tools are paid and require some programming knowledge to use.

In addition, to the best of our knowledge, none of these tools offer the possibility to extract, in a flexible way, writing patterns, that can be used to enhance the performances of classification tasks such as the detection of sarcasm or, as in the current work, the multi-class sentiment analysis.

Therefore, arises the need for a more flexible, yet easy to use and user-friendly tool that allows the extraction of multiple types of features, while offering the possibility to customize them depending on the use case, to obtain performance as high as possible.

In this chapter, we present the first version of SENTA, an open-source tool that performs the extraction of features so they can be used by tools such as Weka [105] to perform the classification.

This tool, as described, is publicly open for any contribution, and hopefully makes a start point for an open-source efficient tool to perform text classification for any purpose.

3.3 Related Work

Twitter data mining has been a hot topic of research in the last few years. Nature of the data mined varies widely depending on the aim and the final result expected. Consequently, the techniques used to process data and extract the needed information are different.

Akcora et al. [11] proposed a method to determine the changes in public opinion over the time, and identify the news that led to breakpoints in public opinion. In a related context, Sriram et al. [57] proposed a method to classify tweets depending on their natures into a set of classes including private messages, opinions and event, etc.

However, most of the work has been focusing on the content of the tweets and how to extract opinions of users towards specific topics or objects. The work of Pang et al. [23] presented the pioneer work for the use of machine learning to classify texts based on their sentiment polarity. In their work, the authors used unigrams, bigrams and adjectives in different ways to classify a set of movie reviews into positive or negative. Other works iterated more on the idea, and new types of features have been used for the classification, depending on the aim and application: Boia et al. [81] and Manuel et al. [82] proposed two approaches that, respectively, rely on emoticons to detect the polarity of tweets and on slang words to assign a sentiment score to online texts. These two works proved how non-textual components can be used to detect the polarity of a text.

More recent works went deeper, and new models have been built: Gao et al. [117] proposed a recent approach that focus in the repartition or the frequency of sentiment classes in the set they analyze. Moving from classification to quantification, the authors concluded that using a quantification-specific algorithm presents a better frequency estimation than using regular classification-oriented algorithms.

Few works have been conducted on the multi-class sentiment analysis. Most of them focused on assessing the sentiment strength into different sentiment strength levels (e.g., *"very negative"*, *"negative"*, *"neutral"*, *"positive"* and *"very positive"*) or simply give numeric sentiment scores to the texts [118, 119]. Nevertheless, other works were conducted to classify texts into different sentiment classes: Lin et al. [120, 121] proposed an approach that classifies documents into reader-emotion categories. They relied on what they qualify as similarity features and word emotion features along with other basic features. The approach, although it shows some potential, is oriented towards the reader rather than the writer. Therefore, the sentiment classes proposed are different from what a writer might intend to show. Similarly, Ye et al. [122] studied the problem of emotion detection of news articles from reader's perspective, and tried various multi-label classification methods and different strategies for features selection to conclude which are to be adopted to solve the problem. Liang et al. [123] proposed an emoticon recommendation system that recommends emoticons for posted texts to help to author decide which emoticon to insert to show what he intends.

3.4 SENTA - A Tool for Features Extraction from Texts

SENTA is a user-friendly tool we developed to extract different features from the tweets, and texts in general, to perform in a later step the classification of tweets/texts into different classes. The features extracted vary widely, and cover the context as well as the form of the text.

SENTA has several graphical interfaces that allow the user to easily input his data, choose the features he wants to extract, and save the output in different formats. In this work, we have used SENTA to extract the necessary features that we used to perform the task of multi-class sentiment analysis in Twitter.

3.4.1 Tools

SENTA was built using Java and Java FXML. While many libraries were used to build this program, mainly OpenNLP was exploited in most of the tasks. OpenNLP has been used to perform the NLP basic tasks such as the tokenization, Part-of-Speech (PoS) tagging and the lemmatization of the texts (i.e., tweets in our case).

3.4.2 Convention

For the rest of this Section, the user of the program SENTA will be referred to as "the user", while the Twitter user whose tweet is processed will be referred to as "the twitterer"

In addition, by interface, we mean a graphical user interface of SENTA.

3.4.3 Pre-Processing of Tweets

During this work, we pre-process each tweet as shown in Fig. 3.1: we start by removing the URLs, tags at the beginning of the tweets and irrelevant content. We then use OpenNLP to tokenize the tweet, get the PoS tags of the obtained tokens, and refer to both (tokens + PoS tags) to get the lemmas of all the words. We then generate what we call a negation vector of the tweet. A negation vector is a vector having the same length as that of the tokens. If the tweet contain a negation word (e.g., *"not"*, *"never"*, etc.), all the tokens (words) that come after, until the next punctuation mark are considered as negated, and are attributed a value equal to 1 in the corresponding negation vector. This will help later detect which words are positive and which are negative. Obviously, many works such as [124] present better solutions to handle the presence of negation and polarity shifting in sentiment analysis, however, we opted for this more straight-forward, yet less complex and faster approach.

We also made an internal tool that decomposes the hashtags into words referring to a dictionary of words occurrence probability as we will describe later on in this work. This decomposition is used also for detecting any sentiment hidden in the hashtags. On a small set of hashtags (i.e., 100 different hashtags) our tool reached a good accuracy of decomposition that reached 88%.

3.4.4 Graphical User Interfaces

Main windows

Project type window As mentioned above, SENTA was developed as a user-friendly tool to extract different possible features from texts. Therefore, to assist the user all over the process, different interfaces are present.

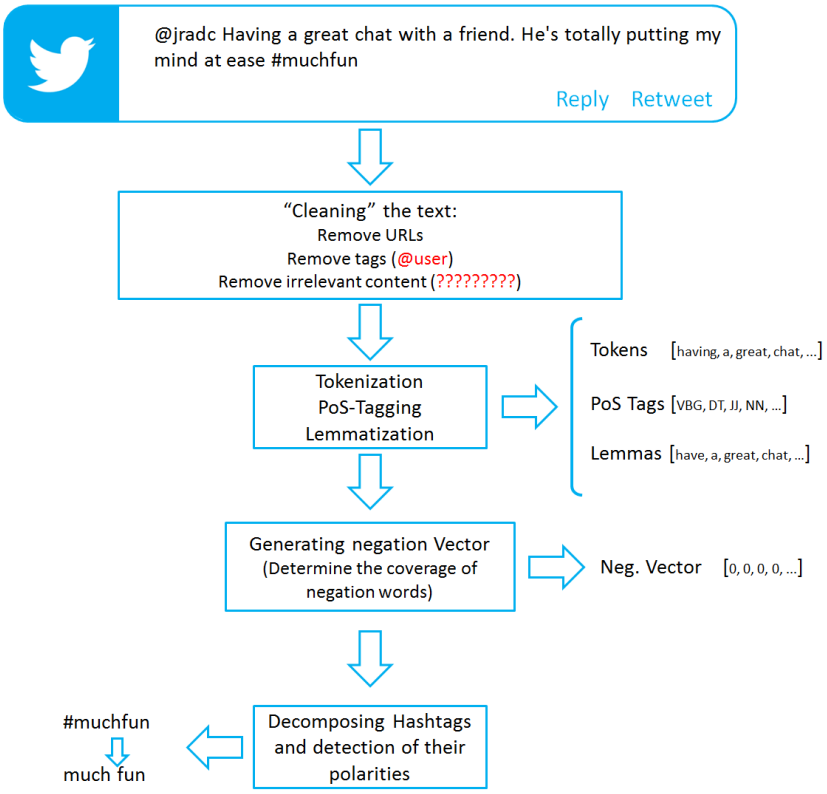

Figure 3.1: Pre-Processing of Tweets

From the first window shown in Fig. 3.2, the user chooses whether he wants to open an already existing project, import features from an existing file (and eventually add them to the ones he will extract once he goes to the next step), or start a new project.

Import project window The import of an existing project supposes that a project has already been created. SENTA allows the user to save an existing project in a file with the extension "*.senta", along with the different files required to load the project.

Fig. 3.3 shows the interface displayed when the user chooses to open an existing project. He has the choice to browse his computer to look for a project, or to select directly one of the recently opened/created projects.

After the selection of the file, the user needs to click *"Get"* to collect the different options, parameters and features to be collected.

- Project type: this refers to whether the sets used in the existing project are a training set and a test set or a training set and a non-annotated set. The difference between a test set and a non-annotated set will be explained later in this section.

- Project name: the name of the project as saved earlier, and this cannot be changed for the existing project, but when saving the current project, the user might choose a different name.

- Training and test files: these are the data sets used previously.

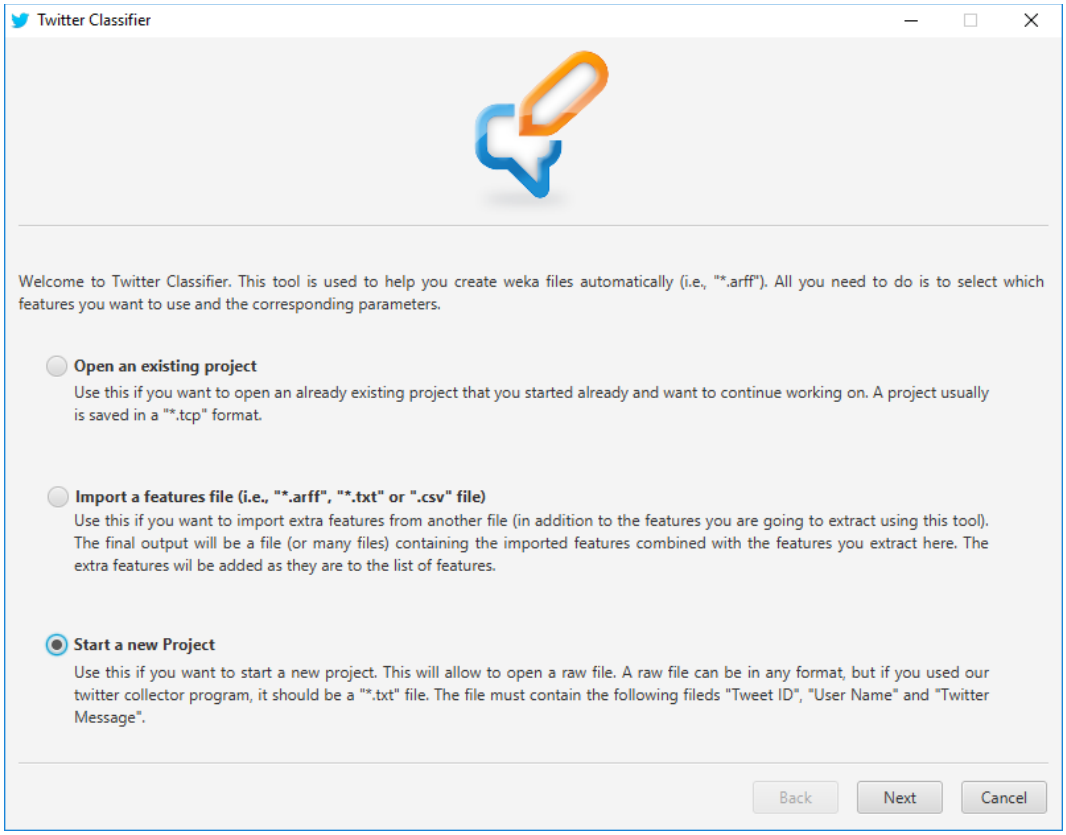

Figure 3.2: The “Main” Window of SENTA

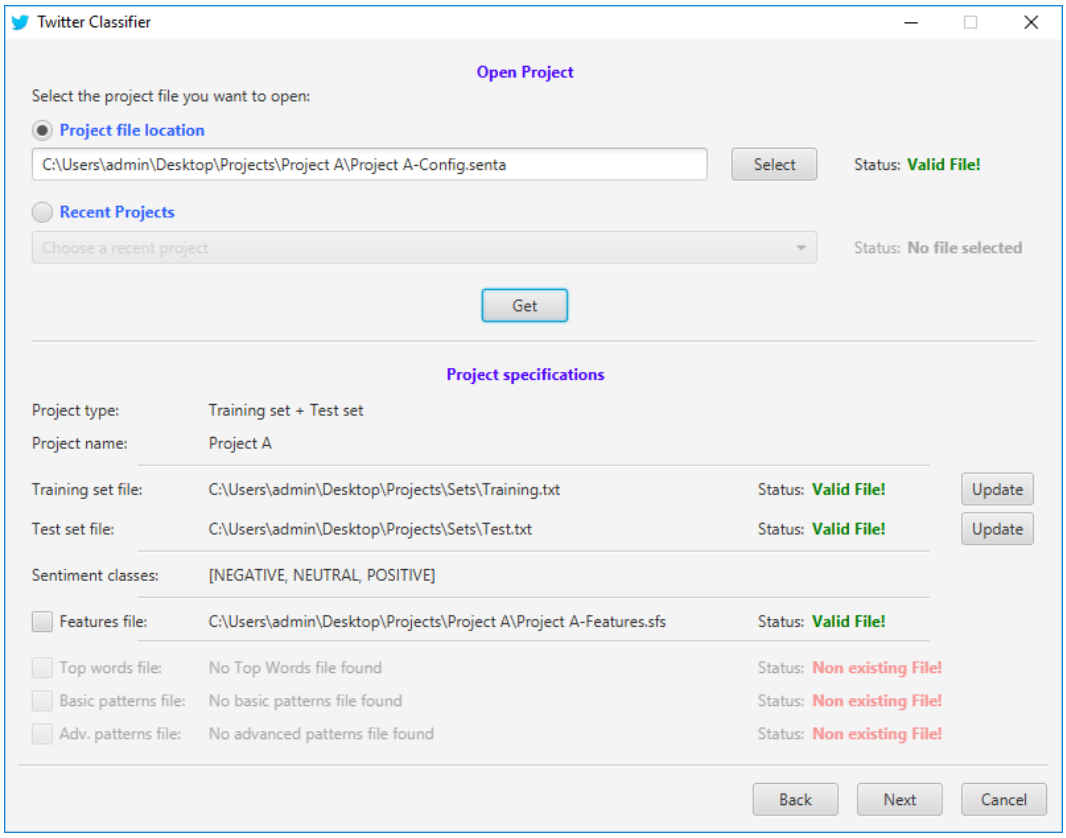

Figure 3.3: The “Open an Existing Project” Window

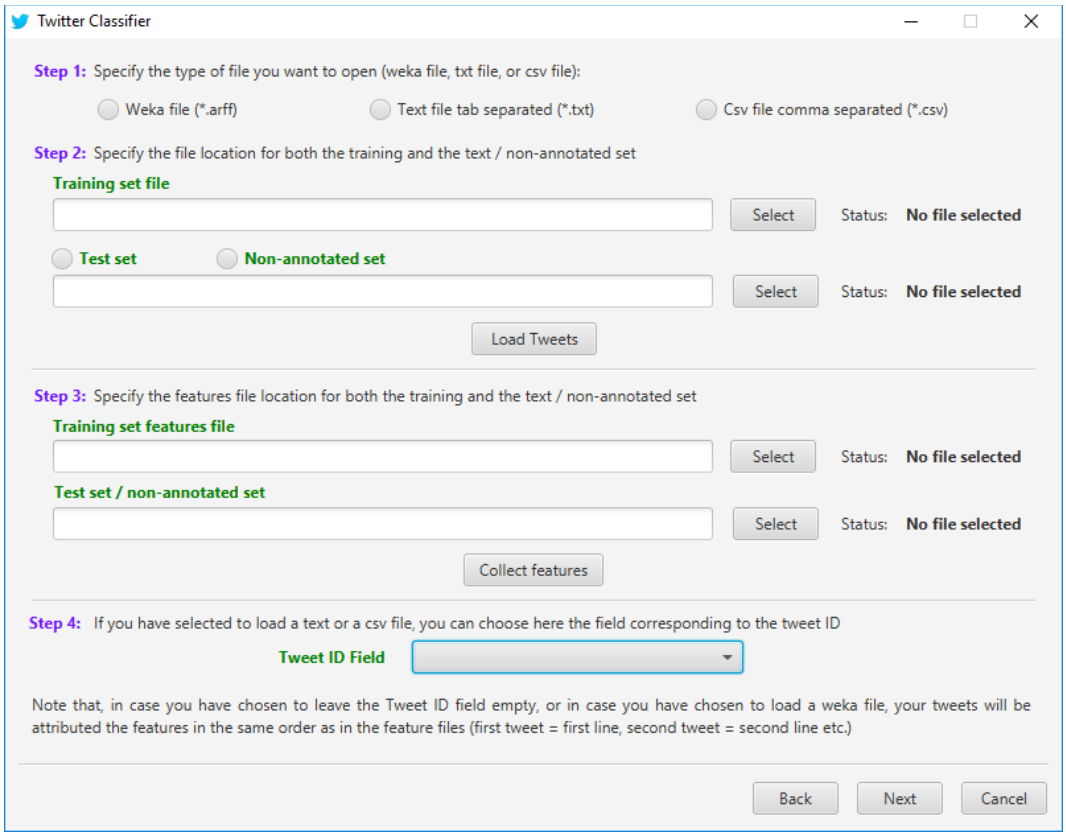

Figure 3.4: The "Import Features" Window

- Sentiment classes: these are the classes that the tweets are supposed to be classified to (extracted from the training set)

- Features file: the different sets of features and feature parameters as selected previously for the opened project.

- Extra files: these are used to make the feature extraction faster, if they have previously been extracted and saved in the corresponding files. These will be explained further later.

For the same project, the user can choose a different training and/or test set (or non-annotated set). He can also choose not to use the old set of features, and select new ones.

Import features window As stated above, in addition to the extraction of features, SENTA allows the import of extra features, which have been extracted using external tools) so that they are added to the set of features extracted by SENTA. Fig. 3.4 shows the window where the extra features can be imported.

In addition to the training and the test/non-annotated sets themselves, the user inputs 2 files corresponding to the extra features.

The user needs to specify the format of the file. Only a Weka file (i.e., "*.arff"), a text file (i.e., "*.txt" tabulation separated) or a CSV file (i.e., "*.csv" comma separated), can be imported.

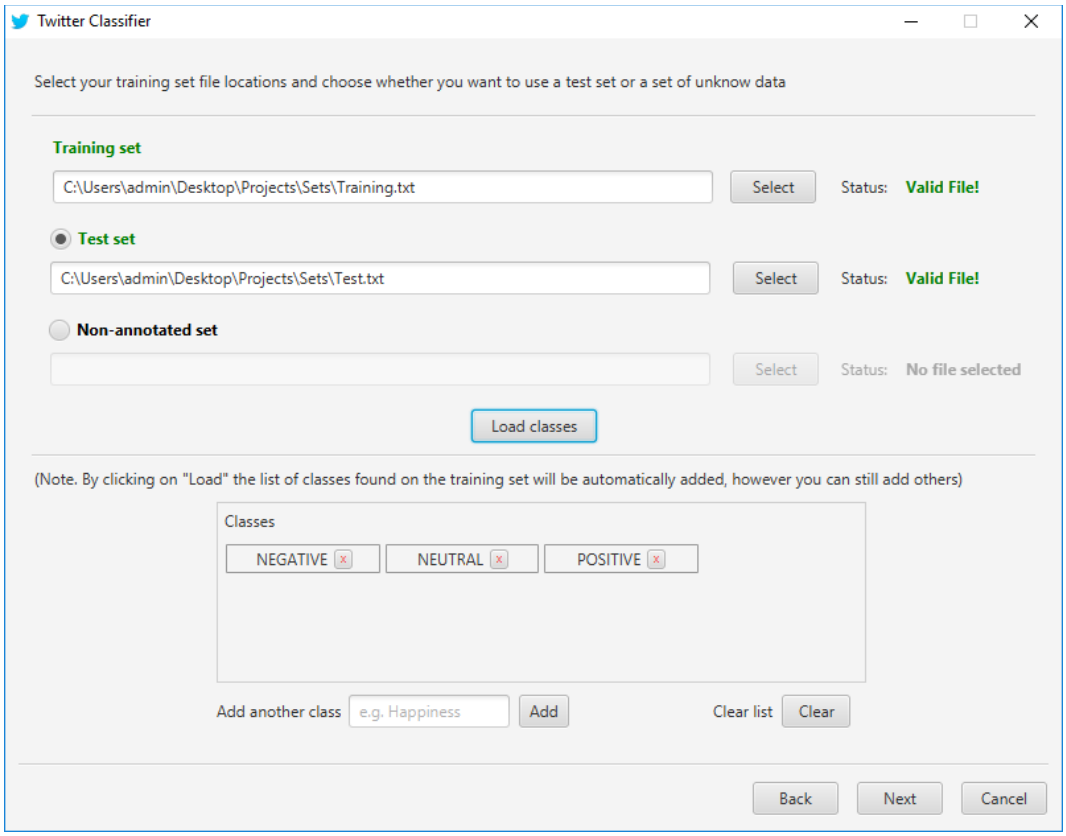

Figure 3.5: The "Start a New Project" Window

The extra features extracted from both the training and the test/non-annotated set need to be provided for all the instances (tweets). In case one of the files is missing or in case of inconsistency in terms of number of instances, the extra features will be dismissed entirely.

Once the user specifies the location of all the files, he needs to click on *"Collect features"* to get the tweets and their features. The training and test/non-annotated sets have a specific format required that will be discussed later on. However, regarding the extra features files, they are highly recommended to contain the Tweet ID field so that the features can match the actual tweets collected from the data sets. If such a field does not exist, the features will be attributed automatically to the tweets in the same order. Obviously in case of inconsistency (e.g., the number of lines in the data set file and the features file are not equal) the features file will be dismissed.

Creation of a new project window However, during this work, no features, other than the ones extracted with SENTA are used. Therefore, we opt for the creation of a new project. To start a new project, the user is supposed to provide two datasets: a training set and either a test set or a non-annotated set as shown in Fig. 3.5. The training set and test set have to contain at least the following attributes:

- Tweet ID: this is the unique ID of the tweet, that will be used in the rest of the work to identify the tweet and that will be used later to save the tweets features.

- Username: the name of the twitterer who posted the tweet. While this information is not used for any purpose during this work, this information might be needed in a future extension (e.g., to detect the gender/location of the user as extra features).
- Tweet message: the content of the tweet itself.
- Class: the user-defined class of the tweet.

The last attribute supposes that the tweets have already been manually annotated by the user, and therefore can be used for training and/or testing. For the same reason, if the user decides to opt for a non-annotated set, in which case he will extract the features and try to perform the prediction of the classes of the different tweets, this attribute is not supposed to be provided, and if given is simply ignored.

Once the files containing the data sets are selected, the user can check the different classes by selecting *"Load classes"*. The user has also the possibility to add extra classes. While this might seem irrelevant and meaningless at this point, these extra classes can be used later to extract extra features (e.g., Unigram features), to enhance the accuracy of classification. This will be discussed later on in this Section.

Feature selection window After the collection of the training tweets and the test/non-annotated tweets, the user is supposed to select the features he wants to extract. The features that can be extracted using SENTA are divided into 7 different sets as shown in Fig. 3.6 that we will cover later on. However, note that all the interfaces that manage the extraction of features are similar.

The 7 sets of features are:

- Sentiment-related features
- Punctuation features
- Syntactic and stylistic features
- Semantic features
- Unigram Features
- Top words
- Pattern-related features

To select a set of features, the user has to check it, and then customize it. The small question mark button next to the name of the set of features opens a help window that explains what the set of features does, and how to configure it.

The features selection along with their parameters can be exported and re-imported for a future project any time.

Once the features and their associated parameters are set, on the main window, the number of features to be extracted for each family of features is displayed.

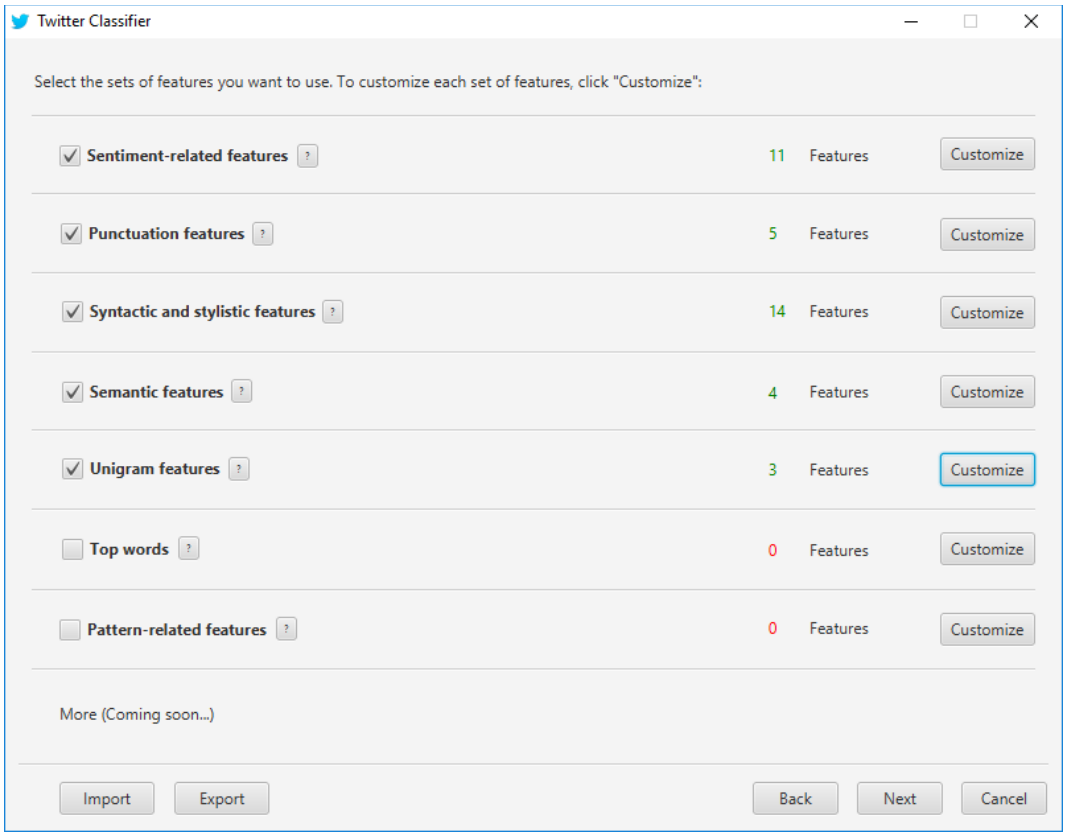

Figure 3.6: The "Features Selection" Window

Save project window The user is then called to choose the different options to save his project as shown in Fig. 3.7, where he has to specify a name for his project, a location for it to be saved, along with the different save options including the type of output and whether some extra data are to be saved or not.

Inside the project directory specified, a subfolder will be created and named after the project name.

The features qualified as *"Top words"* and *"Pattern-related features"* require the extraction of some words, expressions or patterns from the training set (or an independent set other than the test/non-annotated set) as we will discuss later. However, given the fact that this procedure takes some time, or that the user might prefer to extract these dictionaries from an independent set, SENTA offers the option to let the user import these from a different source (and checks if they are valid or not). SENTA also allows him to save the patterns and/or top words at this stage that will be extracted from the current training set (this requires that the user already selected these features to be extracted).

The features, once extracted, can be saved in different formats: a Weka file (i.e., "*.arff"), a text file (i.e., "*.txt" tabulation separated) and/or a CSV file (i.e., "*.csv" comma separated).

Start extraction window Once the project details have been set, the user can start the feature extraction, and keep track of which task is currently being run as well as the tasks already finished

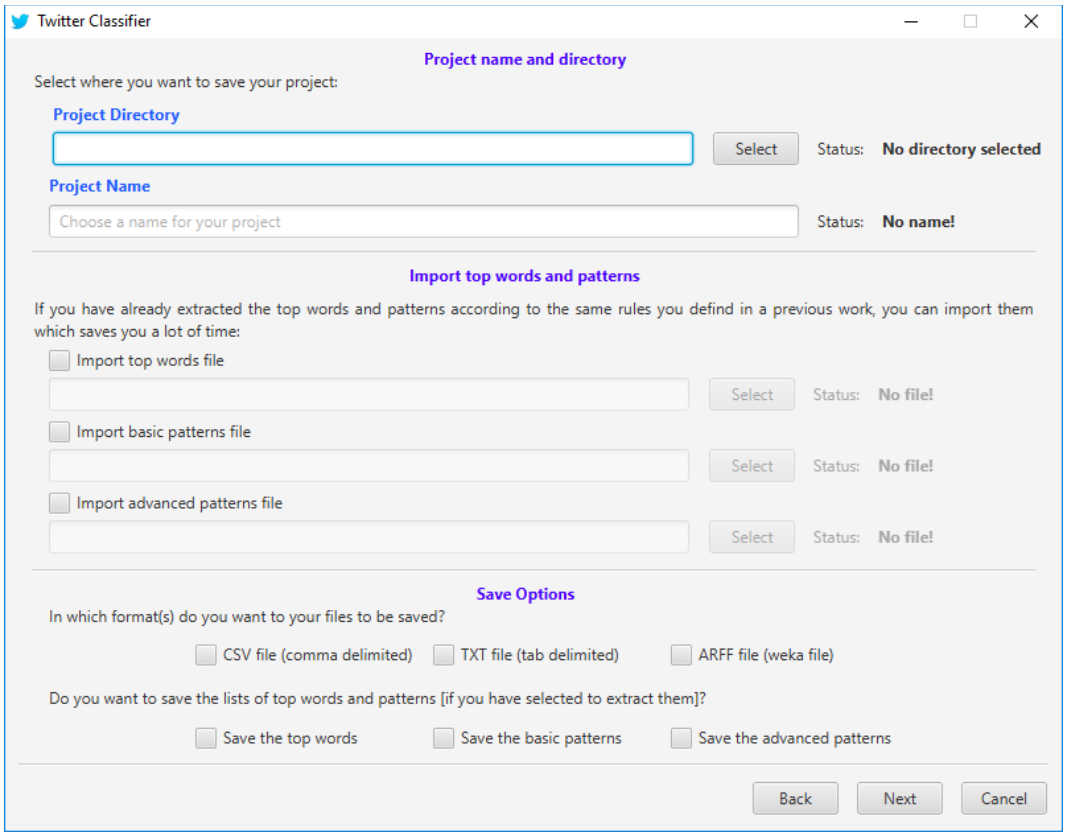

Figure 3.7: The "Save Project" Window

as shown in Fig. 3.8. The time displayed is in seconds (s). The user can also pause the task any time but this will not free any space in the memory neither free the thread being run.

Project Summary window The last interface in the main windows is a recapitulation of the project along with the output files is displayed as shown in Fig. 3.9.

The recapitulation includes in addition to the project name, directory and type, the location and size of the training and test sets, and the files generated along with the project file.

From this point the user can go to the previous interface, go back to the main interface or open in the system explorer the project directory to browse the saved files.

Feature customization windows

Feature customization window appears when a user presses the button *"customize"*. For all the sets of features, we added the button *"Default"* that selects by default the features that we used to perform the multi-class classification in the rest of this work to make it easy to replicate.

Sentiment features Sentiment features are features which rely on the sentiment polarities of the different components of the text such as the words themselves, emoticons, hashtags, etc. These features are extracted using already-built dictionaries and small sub-tools we use internally. Noticeably we referred to SentiStrength to build our dictionary of emotional words, however, we are

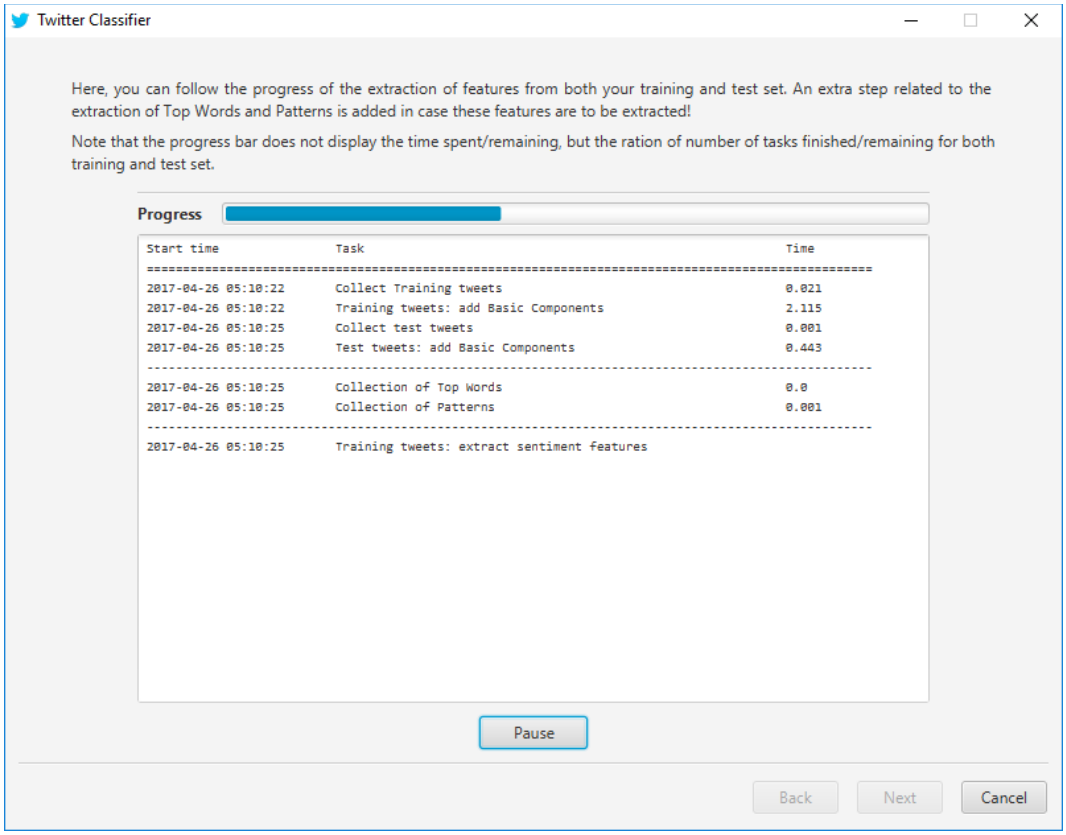

Figure 3.8: The “Start of Collection and Project Progress” Window

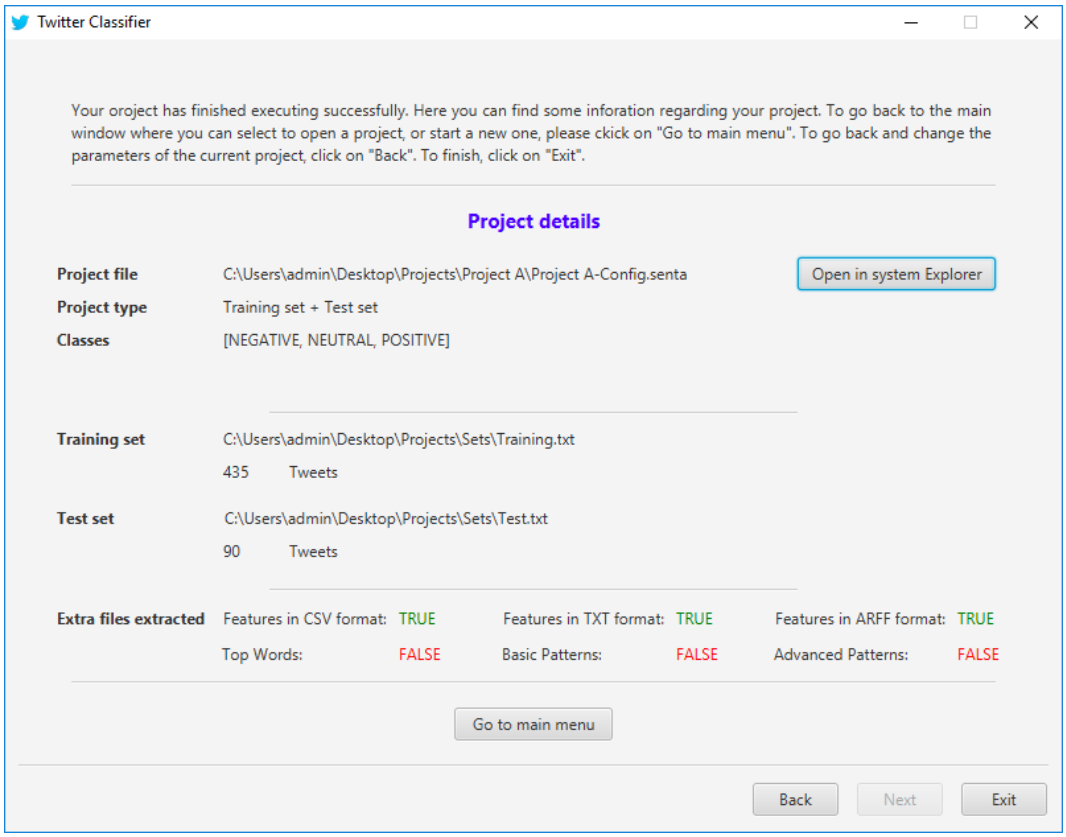

Figure 3.9: The Window Displaying the “Summary of the Project”

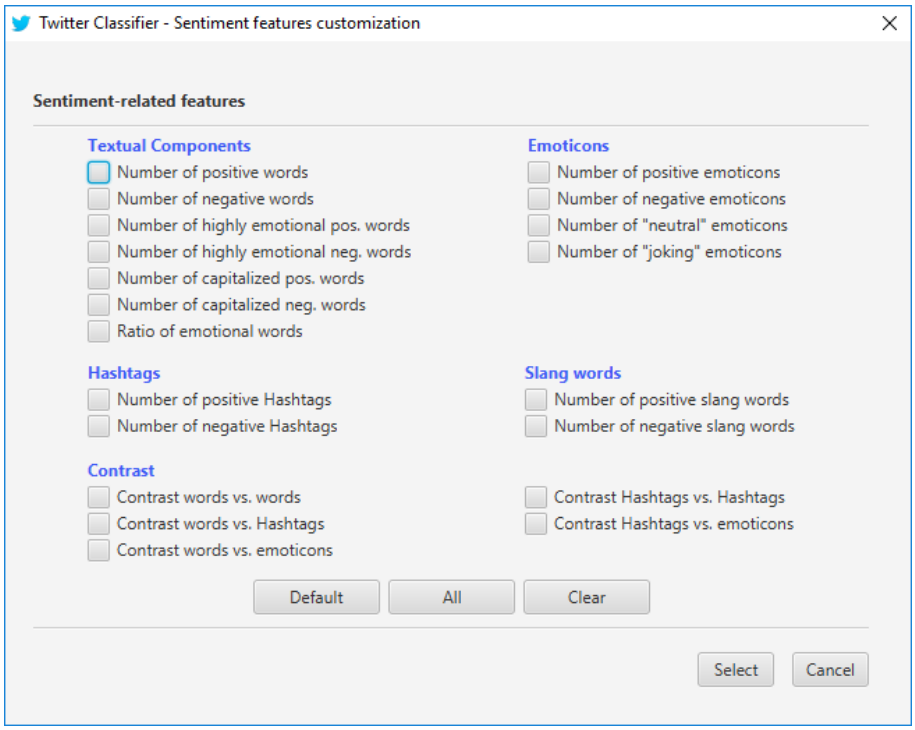

Figure 3.10: The "Sentiment features customization" window

currently building our own. Sentiment features are divided into 5 sub-categories as shown in Fig. 3.10:

- Textual features: these are features that deal with the textual component of the tweet. These include the following features:

- Number of positive words
- Number of negative words
- Number of highly emotional positive words (i.e., words having score returned by SentiStrength greater or equal to 3)
- Number of highly emotional negative words (i.e., words having score returned by SentiStrength less or equal to -3)
- Number of capitalized positive words
- Number of capitalized negative words
- Ratio of emotional words $\rho(t)$ defined as

$$\rho(t) = \frac{PW(t) - NW(t)}{PW(t) + NW(t)} \tag{3.1}$$

where t is the tweet, PW and NW are the total score of positive words and that of negative words as returned by SentiStrength. In case the tweet does not contain any emotional word, ρ is set to 0.

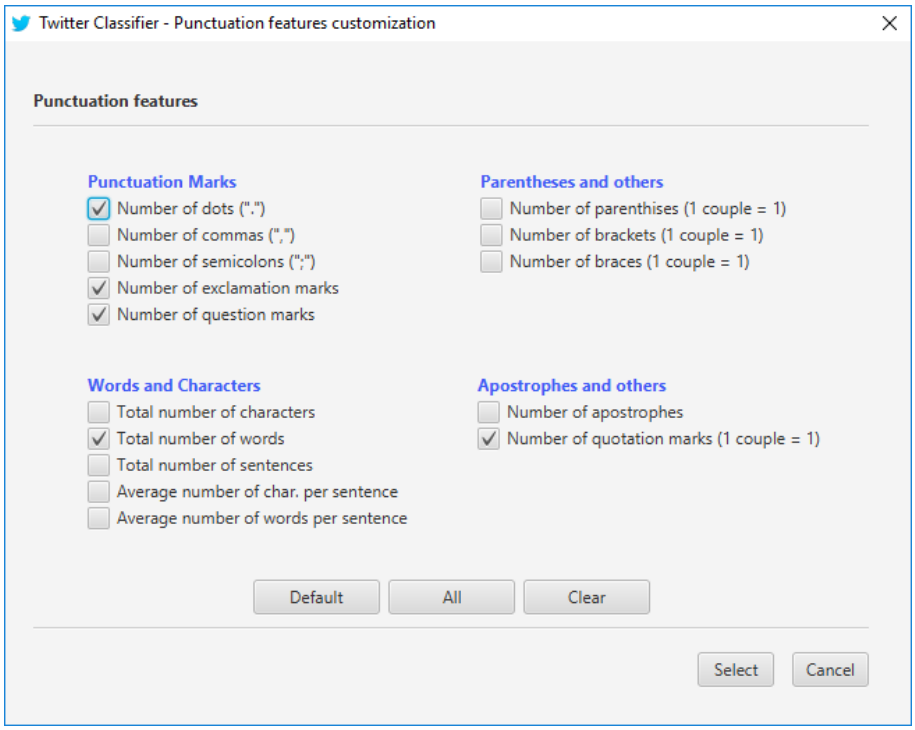

Figure 3.11: The "Punctuation Features Customization" Window

- Emoticons-related features: these include the count of positive, negative, neutral and joking (or ironic) emoticons. Emoticons qualified of neutral are ones who do not show clear emotion such as *"(._.)"* while joking emoticons are ones used sometimes with ironical or sarcastic statements (e.g., *":P"*).

- Hashtags-related features: these include the count of positive and negative hashtags. To decide on a hashtag's polarity, we defined a simple probabilistic model that decomposes the tweet into words, and detects the polarity of the resulting expression.

- Slang words-related features: these include the count of positive and negative slang words. To extract these we refer to a dictionary containing the most common slang words along with their polarities.

- Contrast features: these detect whether there is any contrast between the different components. By contrast we mean the coexistence of a negative component and a positive one within the same tweet, whether the two components have the same nature (e.g., words, emoticons, etc.) or different natures (e.g., words vs emoticons, etc.). In total 5 features are extracted which include the contrast between words, between hashtags, between words and hashtags, between words and emoticons and between hashtags and emoticons.

Punctuation features Punctuation features are ones related to the use of punctuation marks as well as the capitalization of words, etc. as shown in Fig. 3.11. They are divided into 4 subcategories:

- Punctuation marks: these include the number of full stops, commas, semicolons, exclamation marks and question marks.

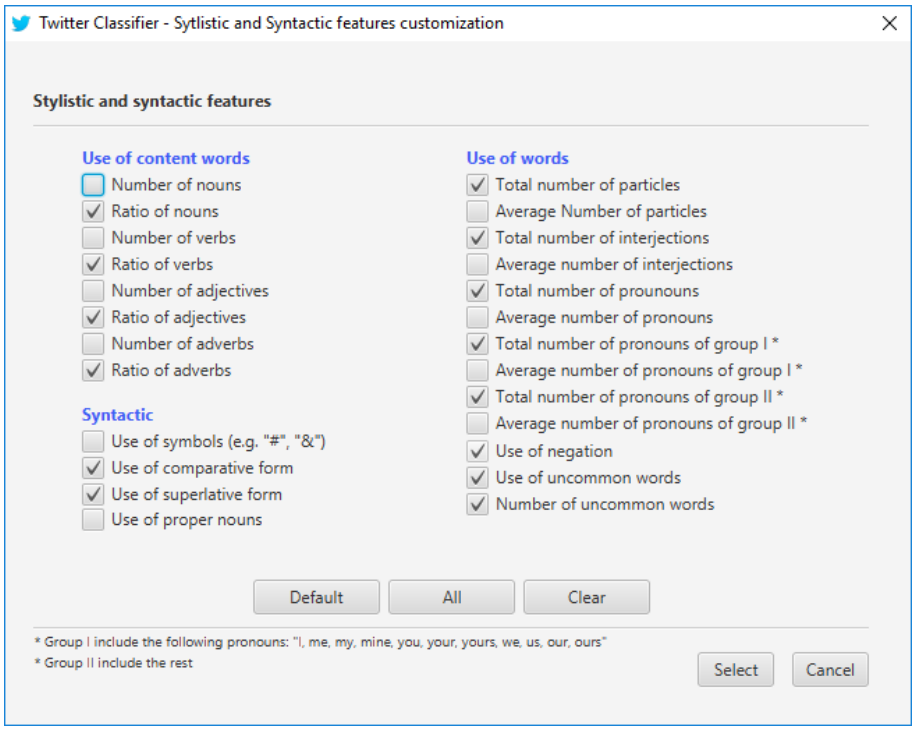

Figure 3.12: The "Stylistic and Semantic Features Customization" Window

- Parentheses and similar symbols: these include the number of parentheses, brackets and braces.

- Words and characters these include the count of words and characters, the average number of words and characters per sentence, etc.

- Apostrophe and quotation marks

Syntactic and stylistic features Syntactic and stylistic features are ones related to the use of words and expressions in the tweet/text. They are divided into 3 sub-categories as shown in Fig. 3.12:

- Use of content words-related features: content words are nouns, verbs, adjectives and adverbs. The features extracted are the count and the ratio of each aside.

- Syntactic features: these are related to the use of some speech forms, proper nouns, and symbols.

- Use of words: these are features related to the use of non-content words such as particles, interjections, pronouns, negation. They also include the use of uncommon words (which might obviously be content words). To judge whether a word is common or not, we referred to a big amount of texts collected online. We calculated the probability of use of the different words and qualified the top 5,000 words as *"common"* while the rest are considered as *"uncommon"*.

Semantic features Semantic features are ones related to the meanings of words in the language as well as the logic behind it. Fig. 3.13 shows the features window. In the current version of the project, very few features are to be extracted. They include the use of opinion words or

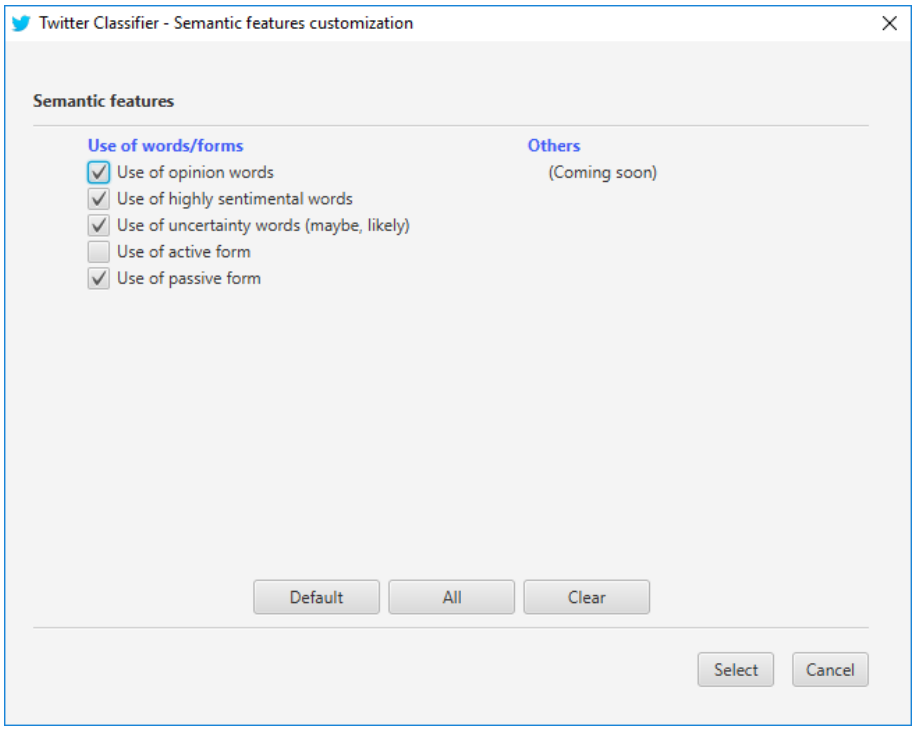

Figure 3.13: The "Semantic Features Customization" Window

expressions, the use of highly sentimental words, the use of uncertainty words and the use of active and passive forms.

Unigram features Unigram features are kind of special features that are extracted with reference to dictionaries built according to the user's defined parameters. Since proposed by Pang et al. [23], unigrams and *n*-grams in general, have been used as basic features for sentiment analysis using machine learning. In the different approaches, unigrams are collected from the training data sets, and either the count or the presence of these unigrams is used as features for the classification. In this work, we make use of WordNet [125] to collect unigrams related to each sentiment class. The user is supposed to come up with a small set of seed words few in number for each class, and used WordNet to collect their synonyms and hyponyms down to a certain depth. The choice of synonyms and hyponyms is based on the fact that these words are highly correlated with the initial seed word, and usually describe the same object, if not a more precise one. While synonyms refer usually to equivalent terms, hypernyms and hyponyms show the relationship between the more general term and its more specific instances.

A hypernym, or a superordinate, is a broader term than a hyponym, whereas a hyponym is a word or an expression which is more specific than its hypernym. For example, for the word *"feeling"*, two of its direct hypernyms are *"perception"* and *"idea"*. The words *"happiness"*, *"anger"* and *"fear"* are some of its hyponyms.

Hypernyms might lose some of the specificities of the initial word, therefore, in our study, we collect only synonyms and hyponyms of the seed words. On the other hand, hyponyms also might lose the original meaning of the word, and collide with some of other classes. Therefore, the depth

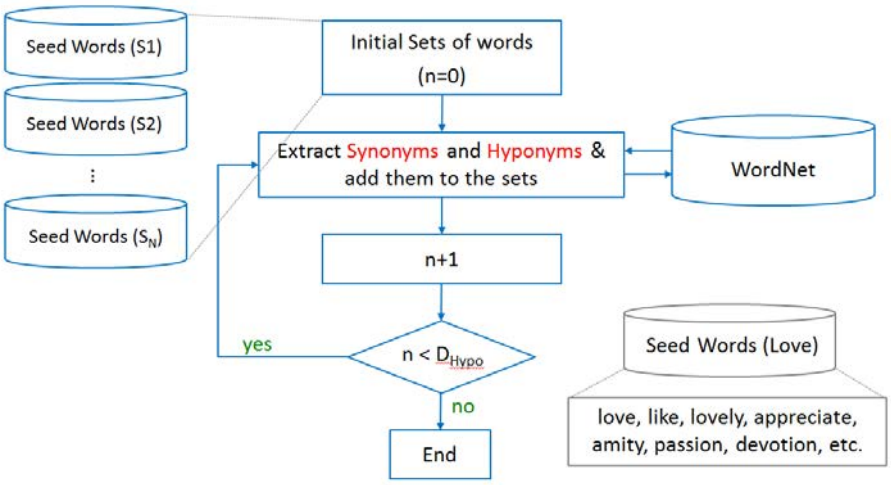

Figure 3.14: Flowchart of the Procedure of Unigram Extraction

down to which we collect the hyponyms is set to a certain value we refer to as $Depth$ (or D_{hypo}, which is a parameter to optimize by the user).

This is explained in Fig. 3.14 which shows how the dictionaries are extracted: we start with a set of seed words for each sentiment class. We then collect the synonyms and hyponyms to get to new sets of words, from which we further extract the synonyms and hyponyms. The same process is repeated over and over D_{hypo} times.

Fig. 3.15 show the different parameters set for unigram features: in SENTA, the extracted words can be used as individual binary features (i.e., a feature for each word that detects whether or not that word appear in the tweet/text or not) or they are all summed for each sentiment class, and the count of words from each set on a given tweet is used as a separate feature. They can also be separated based on their PoS (i.e., nouns, verbs, adjectives and adverbs each aside) so instead of having one group of words per sentiment class, the user can get up to 4. This is because the number of words to be extracted totally has to be set prior to the extraction. The user can also choose to collect only words of just one or two PoS out of the 4. This set of features has been proven to be very efficient in detecting the sentiment of tweets as we will discuss later in this chapter.

The sets of seed words can be defined by pressing *"manage seed words"*. By default, SENTA offers seed words for 12 different sentiment classes so that, if any of them is present, when the user chooses to import default seed words, they are added. The interface showing how to add a seed word is given in Fig. 3.16. The user types the word, chooses its PoS and the class it belongs to.

Top Words Top words, as their name indicate, are the words that occur the most in the training set. Fig. 3.17 shows the parameters related to this set of features: The user can choose the PoS of the top words to be collected, whether he wants each PoS-related words to be extracted separately, the number of Top Words per class or PoS, and again whether the features are binary or numeric.

The two parameters *"Min Ratio"* and *"Min Occurrence"* define the criteria of extraction of top words. For a positive sentiment class "A" (e.g. "Happiness"), the ratio of occurrence of this word

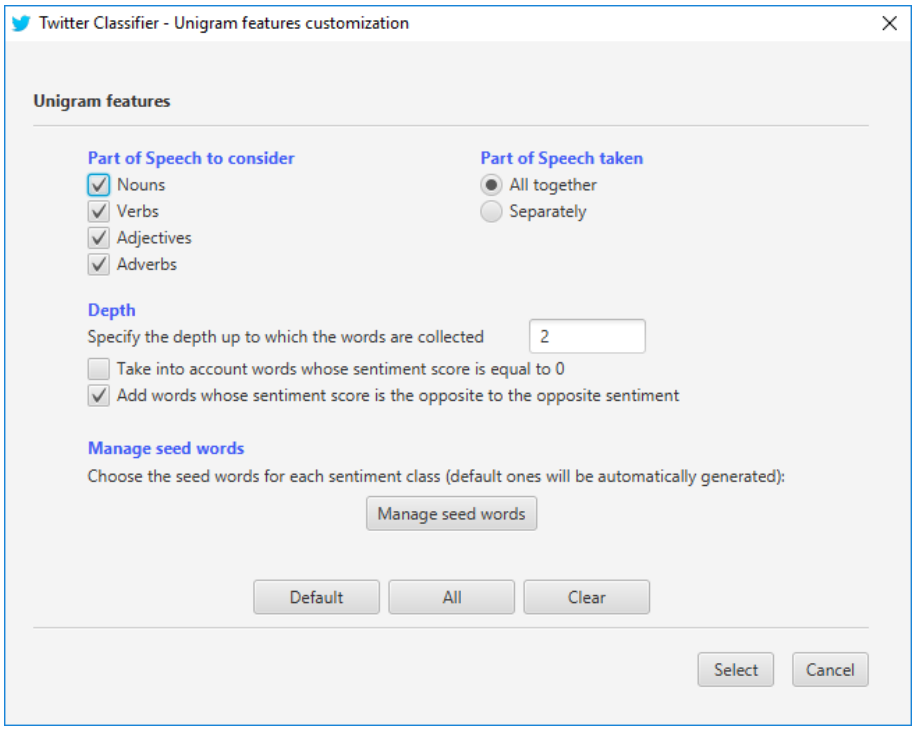

Figure 3.15: The "Unigram Features Customization" Window

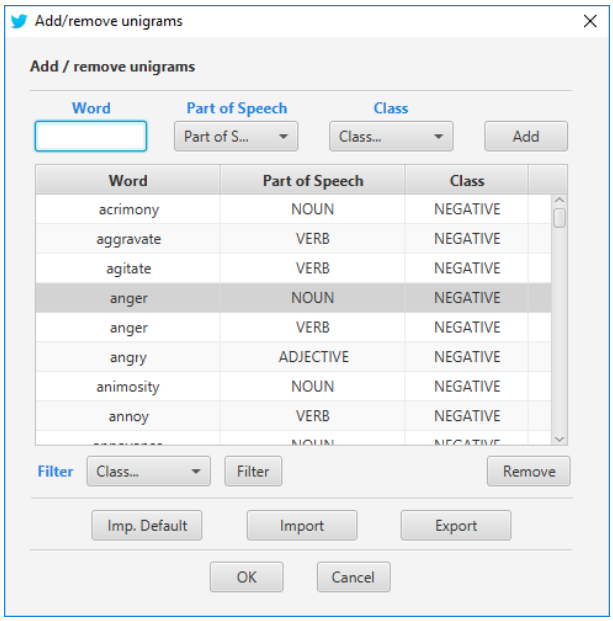

Figure 3.16: The "Seed Words Management" Window

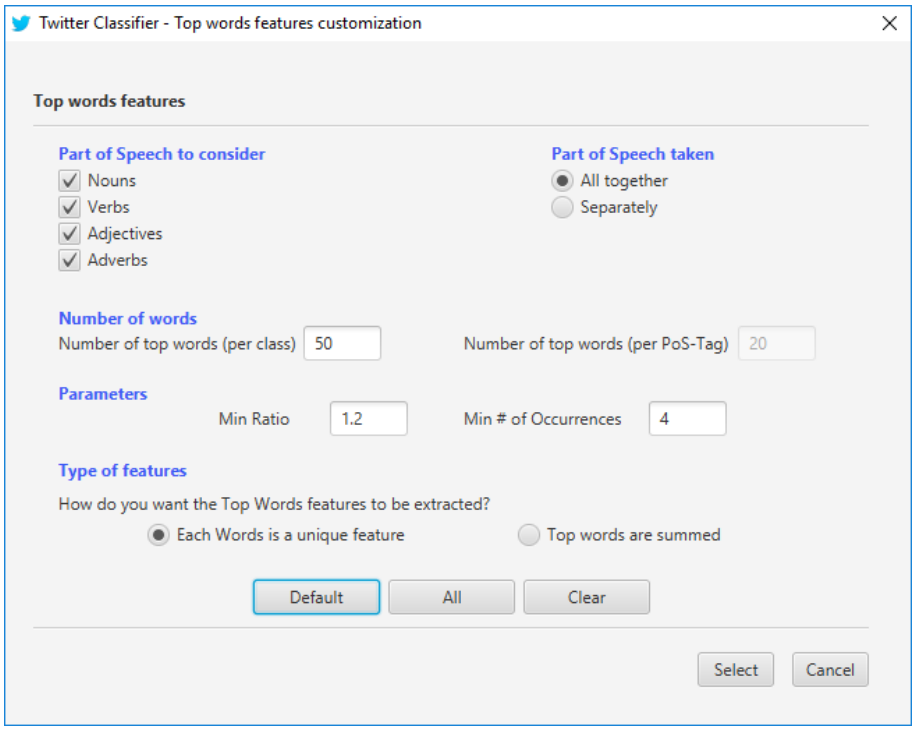

Figure 3.17: The "Top Words Features Customization" Window

on the positive sentiment tweets over that on all the negative sentiment tweets should be higher than *"Min Ratio"*. In addition, it has to occur on the sentiment class "A" more than the value set for the parameters *"Min Occurrence"*. In this work, when we run the multi-class sentiment analysis on our training and test tweets, Top Words have not been used as features, for the reason that they present some redundancy with unigram features, since many of the words on both collide.

Pattern-related features The idea of our pattern-related features has been in the previous chapter (i.e., chapter 2) and in our work [126], in which we proposed an approach that relies on PoS-tags to extract sarcastic patterns. In SENTA we elaborated more this kind of features, and made a more generic approach to extract patterns. Patterns are extracted based on the PoS-tags of words: the different possible PoS-tags (36 in total, along with a 37th one referring to the punctuation) are divided into different groups, and given a sentence S, containing n different words, the words of S are subject to different actions based on their PoS-tag, and according to the rules defined by the user.

Fig. 3.18 shows the different parameters of the Pattern features: initially, the user defines whether he wants his pattern to be used each as a separate feature, or summed based on their length and sentiment class. If the features are separate (i.e., each is a unique feature), only one pattern length is taken into account, otherwise he can choose a minimal and a maximal length for patterns. The user then chooses how many categories he wants his features to be divided into, and specifies the action to do for each category by pressing *"Customize"*. The different actions for the different categories are given in Fig. 3.19: a word can be kept as it is, lemmatized, replaced by a specific expression, or by a user defined expression, etc.

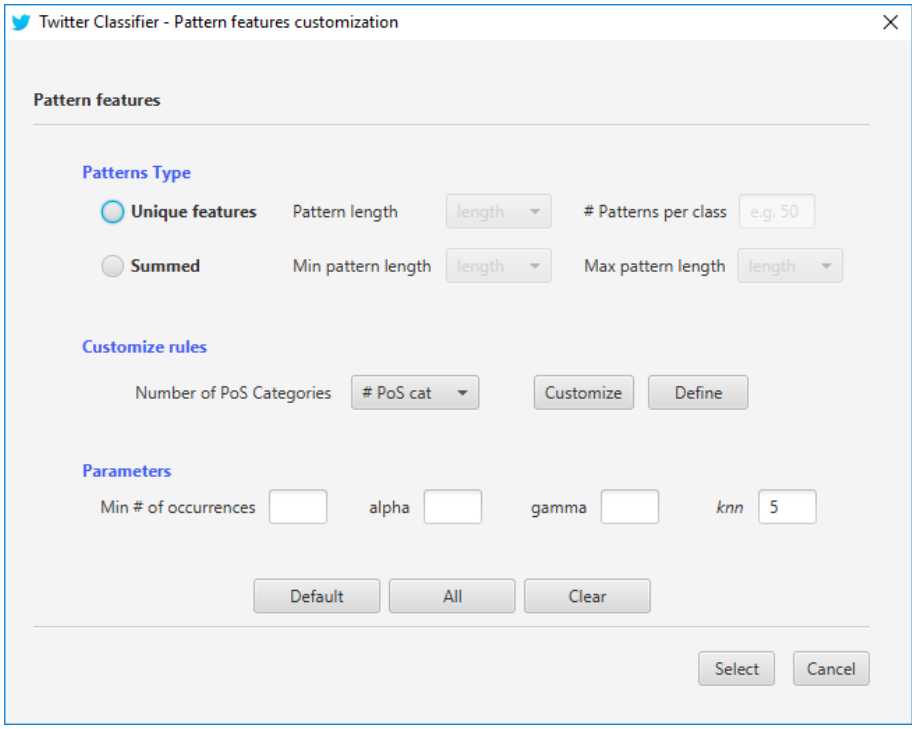

Figure 3.18: The "Pattern-Related Features Customization" Window

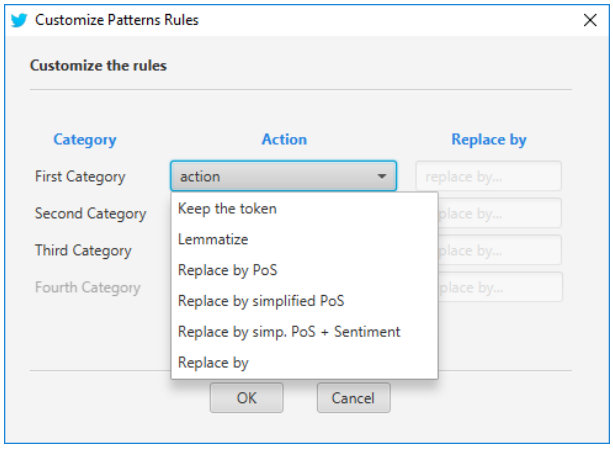

Figure 3.19: The Different Actions for Different PoS-Tags Categories

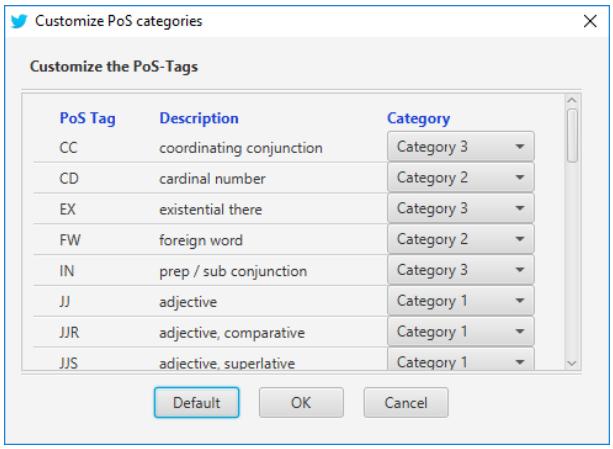

Figure 3.20: The "PoS-Tags Categories Customization" window

The user is next supposed to specify for each PoS tag, which category it belongs to by pressing the button *"Define"* which displays the window shown in Fig. 3.20.

Later on this work, when performing the multi-class classification, we will give a concrete example of how patterns are extracted using SENTA. A pattern should occur on a given sentiment class at least the value of the parameter *"Min # of Occurrences"* times to be considered. Given a full pattern T extracted from a tweet, and a pattern P extracted earlier from the training set, we define the following resemblance function [77]:

$$res(p,t) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order,} \\ \alpha, & \text{if all the words of the pattern appear in the tweet in the correct order but with other words in between,} \\ \gamma \cdot n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appears in the tweet.} \end{cases}$$

The resemblance function defined above is similar to that in chapter 2; however, it has been adjusted by adding the parameter γ.

If the patterns are used as unique features, each feature takes the value of resemblance as defined. Otherwise, the patterns are grouped into different groups based on their sentiment class and length as shown in TABLE 5.2 where $L_1 \cdots L_M$ are the different lengths of the patterns, and $S_1 \cdots S_M$ are the different sentiments (classes).

Given the K patterns $\{p_1, \cdots, p_k\}$ extracted for the sentiment class S_i which resemble the most the tweet in question, and the length L_j p the value of the feature F_{ij} is

$$F_{ij} = \sum_{k=1}^{K} res(p_k, t) \tag{3.2}$$

Table 3.1: Pattern Features

		Pattern length			
		L_1	L_2	$\cdots$	L_M
Sentiment Class	1	F_{11}	F_{12}	$\cdots$	F_{1M}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	S_N	F_{N1}	F_{N2}	$\cdots$	F_{NM}

F_{ij} as defined measures the degree of resemblance of a tweet *t* to patterns of class *i* and length j. Therefore, two more parameters are to be defined by the user which are α and γ.

3.4.5 Extensibility

In its first version, which we introduce here, SENTA extracts some basic features that allow performing tasks such as sentiment analysis, even for multiple classes. However, for more advanced tasks, we believe that it requires more features to be added.

In the second version, more sets of features we qualified are added. These include "advanced semantic features" and "advanced pattern features" that extract deeper features from the texts. However, othor features related to causality, conditionality, differentiation of informative and interrogative form, etc. are to be added.

The different components added to SENTA are detailed in chapter 5.

3.5 Multi-Class Sentiment Analysis - Proposed Approach

3.5.1 Problem Statement

Given a set of tweets, we aim to classify each one of them to one of the following 7 classes: *"love"*, *"happiness"*, *"fun"*, *"neutral"*, *"hate"*, *"sadness"* and *"anger"*. Therefore, from each tweet, we extract different sets of features, refer to a training set and use machine learning algorithms to perform the classification.

We have chosen the aforementioned sentiment classes for different reasons. First of all, given our observation during our work [127], we mainly concluded that we needed a balanced amount of data between negative and positive classes. In addition, while the aforementioned sentiments are the ones present the most in tweets as observed in [128].

3.5.2 Data

For the sake of this work, we manually collected and prepared 2 datasets as follow:

- **Set 1:** this set contains 21 000 tweets which have been manually classified into the 7 classes, each containing 3 000 tweets. This set is used for training. Therefore, in the rest of this work, it will be referred to as the *"training set"*.

Table 3.2: Structure of the Dataset Used

Class	Training set	Test set
Fun	3000	2643
Happiness	3000	2963
Love	3000	1945
Neutral	3000	4989
Sadness	3000	4528
Anger	3000	1558
Hate	3000	1115
Total	**21 000**	**19740**

- **Set 2:** this set contains 19 740 tweets. All tweets are manually checked and classified into the 7 classes. This set will serve as a test set. Therefore, in the rest of this work, it will be referred to as the *"test set"*.

The structure of the dataset used is shown in TABLE 3.2.

3.5.3 Features Extraction

Under different emotional conditions, humans tend to behave differently. This includes the way they talk and express their feelings. Therefore, it might be important to rely, not only on the vocabularies used, but also on the expressions and sentence structures used under the different conditions, to quantify and model these feelings. Therefore, in the rest of this section, we rely on these assumptions to extract different sets (or families) of features.

The features are extracted using SENTA, the tool we introduced in Section 3.4.

Sentiment-based features

As stated above, sentiment-based features are ones based on the sentiment polarity (i.e., *"positive"*/*"negative"*) of the different components of tweets. Out of the different features offered by SENTA, we extract the following ones:

- The number of positive words and that of negative words,
- The number of highly emotional positive words and that of highly emotional negative words,
- The ratio of emotional words,
- The number of positive and negative emoticons,
- The number of positive and negative slang words.

Punctuation-based features

While punctuation do not usually show any sentiments explicitly, except for exclamation marks maybe, we believe that the excessive use of some (e.g., question marks, exclamation marks, etc.) shows the strength of some sentiments. For example, the following two tweets might show different sentiments according to the annotators:

- *"Why didn't you go with him?"*

- *"Why did you tell her???????"*

While in both examples, the twitterers are asking questions, in the first one, the annotators agreed on classifying the tweet as totally neutral, whereas in the second, some of them pointed out that the twitterer is most likely angry or upset. Even though, it is quite hard to tell whether it is the case or not, we agree with the annotator on the fact that the second tweet might be sentimental, regardless of what sentiment is present, while the first one is neutral.

Out of the variety of punctuation features, after our preliminary experiments, we decided to use the following ones:

- The number of full stop marks,
- The number of exclamation mark,
- The number of Question Marks,
- The total number of words,
- Number of quotation marks.

Syntactic and stylistic features

In addition to the aforementioned sets of features, we also extract features related to the use of words. We first extract the ratios of nouns, verbs, adjectives and adverbs in the tweets (out of all the words, including hashtags, symbols, etc.). We also check whether or not the twitterer employed the comparative and/or the superlative forms.

Furthermore, our experiments showed the usefulness of the following features as good indicators of sentiment polarity, as well as the sentiment class for some of them:

- The total number of particles,
- The total number of interjections
- The total number of pronouns, that of pronouns of group I and II separately,
- The use of negation,
- The use, and the total number of uncommon words.

Semantic features

Semantic features are features that focus on the meanings in the language or the logic inside of the sentences. While these features have not all been added, we used few of the existing ones, including:

- The use of opinion words,
- The use of highly sentimental words,

- The use of uncertainty words,
- The use of the passive form of speech.

Unigram features Vs top words features

"Unigram features", as described above, are numeric features that rely on WordNet to be extracted. In brief, a set of seed words for each sentiment class is provided and we use WordNet to enrich them. We then extract N features (where N is the number of sentiments) by counting, for a given tweet, how many words from each set exist in it.

"Top words", on the other hands, are words that are extracted from the training set itself. From all the training tweets of a given sentiment S, we collect the most commonly used words while making sure that the words extracted are ones that show the given sentiment (i.e., that the number of occurrences of any word in the tweets of the sentiment S is higher enough than its occurrences in the tweets of the other sentiments). These words are used later as indicators (features) to detect the sentiment of a given tweet.

However, given the nature of these two sets of features, a huge part of the words will overlap, and create a useless redundancy that we do not need. Therefore, for the sake of this work, we discarded *"Top Words features"*, and focused on what we qualified as *"Unigram Features"*.

We started with 6 sets of words (i.e., for all the sentiments except the sentiment *"Neutral"* containing in total 486 words, with an average number of 81 words for each sentiment. The initial set of words contains an overlapping equal to 0 between words of sentiments of opposite polarities, while we tolerated some overlapping for sentiment of the same polarities (e.g., the word *"enjoy"* is a seed word for both sentiments *"happiness"* and *"fun"*). The words selected can be nouns, verbs, adjectives and adverbs.

Judging from the Fig. 3.21, the overlapping (or duplication) of words in different sentiments including that in sentiments of different polarities increases rapidly. Even though, these words are being removed automatically, the duplication is a crucial indicator of where to stop continuing collecting the words. In this work, we were restricted to a depth equal to 2.

As described above, we use the resulted sets of words to extract 6 features, by counting the occurrences of the words in the tweet to classify, taking into consideration the score of the words.

Pattern-based features

As described in Section 3.4, patterns are used as a complementary set of features to detect what unigrams cannot detect: while in most of the cases, sentimental words are enough to tell the sentiment of a sentence, in other cases, the person employs some specific longer expressions to express his sentiment. For example, the following tweet shows sentiments of happiness without employing any sentimental word showing explicitly happiness:

"You took me to the world I always dreamt of!!! Thank you soooo much!"

Even though the word *"thank"* refers to a positive attitude or sentiment, the tweets contains sentiments of happiness that the twitterer shows and for which she thanks her friend.

To detect such expressions and learn them, we refer to patterns of speech.

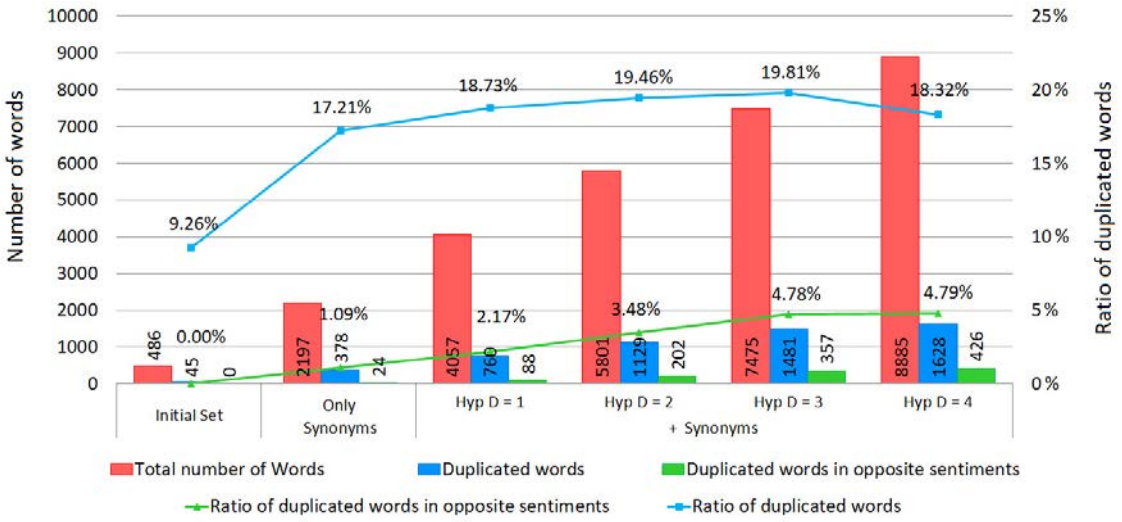

Figure 3.21: Number of Unigrams Collected from WordNet Using the Seed Words Proposed

Table 3.3: Expressions Used to Replace the Words of EI and GFI

PoS-tag	**Expression**
"CD"	[CARDINAL]
"FW"	[FOREIGNWORD]
"UH"	[INTERJECTION]
"LS"	[LISTMARKER]
"NN", "NNS", "NNP", "NNPS",	[NOUN]
"PRP", "PRP$"	[INTERJECTION]
"MD"	[MODAL]
"RB", "RBR", "RBS"	[ADVERB]
"VB", "VBD", "VBG", "VBN", "VBP", "VBZ"	[VERB]
"WDT", "WP", "WP$", "WRB"	[WHDETERMINER]
"SYM"	[SYMBOL]

We basically divide the PoS tags into three categories: a first one, referred to as *EI*, containing words which might have emotional content, a second one, referred to as *"CI"*, containing non emotional words whose content is important and a third one, referred to as *"GFI"*, containing the words whose grammatical function is important. If a word belongs to the first category, it is replaced by the corresponding expression shown in TABLE 3.3 along with its polarity (e.g., the word *"good"* would be replaced by *POS-ADJECTIVE*); if it belongs to the second, it is lemmatized and replaced by its lemma; and if it belongs to the third, it is replaced by the corresponding expression shown in TABLE 3.3.

As mentioned above, the classification into categories is done based on the PoS-tag of the word. The list of part-of-speech tags and their category is given in TABLE 3.4.

We generate the vector of words for each tweet as defined. For example, the following PoS-tagged tweet *"He_PRP is_VBP dummy_JJ ,_, why_WP would_VBD you_PRP think_VBP I_PRP want_VBP to_TO go_VB with_IN him_PRP !!!!_."* gives, among others, the following pattern vector [PRONOUN VERB NEG-ADJECTIVE . *why* VERB PRONOUN VERB PRONOUN POS-VERB

Table 3.4: Part-of-Speech Tag Categories

Class	PoS Tags
CI	"CC", "DT", "EX", "IN", 'MD", "PDT", "POS", "RB", "RBR", "RBS", "RP", "TO", "WDT", "WP", "WP$", "WRB"
GFI	"CD", "FW", "LS", "NNP", "NNPS", "PRP", "PRP$", "SYM", "UH"
EI	"JJ", "JJR", "JJS", "NN", "NNS", "VB", "VBD", "VBG", "VBN", "VBP", "VBZ"

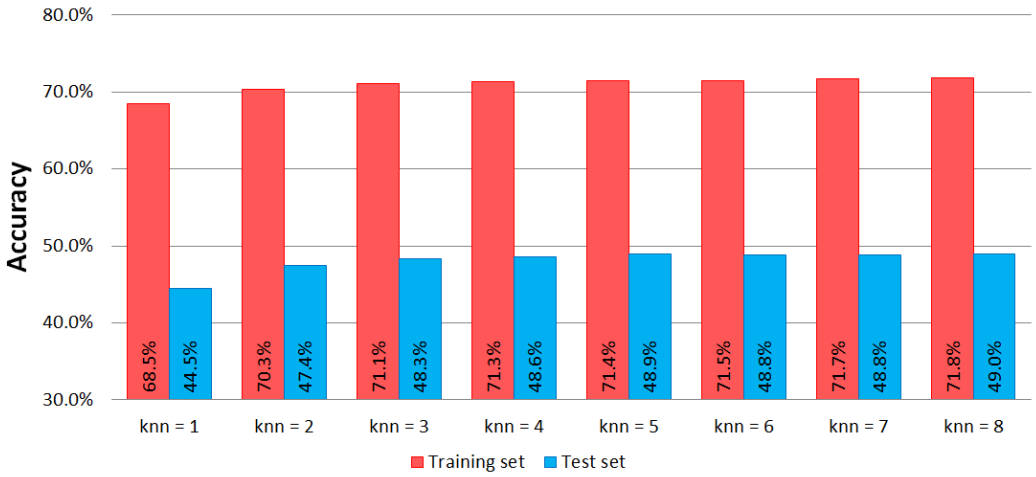

Figure 3.22: Accuracy of Classification Using Pattern-Based Features for Different Value of K

to VERB *with* PRONOUN .] that can be later used to generate smaller patterns following the rules defined (i.e., minimal and maximal lengths of patterns).

In this work, we opted for the use of patterns of different lengths, so that the features created are small in number to make the classification task run faster.

Based on our work [127] and with few adjustments, we set that the most adequate values for N_{occ}, L_{min}, L_{max}, α and γ as follows:

$$\begin{cases} N_{occ} &= 3, \\ L_{min} &= 3, \\ L_{max} &= 10, \\ \alpha &= 0.1, \\ \gamma &= 0.02, \end{cases}$$

On the other hand the parameter K has been introduced in this work since we noticed a high imbalance between the number of patterns for each class. Fig. 3.22 shows the classification accuracy using pattern-based features for different values of K. According to the figure, the best value is 5. Higher values enhance the accuracy during cross-validation, but have no big impact on that of the test set.

Table 3.5: Binary Classification Accuracy, Precision, Recall and and F-Measure

	Accuracy	Precision	Recall	F-Measure
Positive	0.789	0.820	0.789	0.805
Negative	0.835	0.806	0.835	0.820
Overall	**0.813**	**0.813**	**0.813**	**0.813**

In the next section, we evaluate the model we built, and present the results of our experiments in the cases of binary, ternary and multi-class classification.

3.6 Experimental Results

After the extraction of features, we run different test using *"Random Forest"* [109] classifier. We use 4 Key Performance Indicators (KPIs) to evaluate the effectiveness of our approach: Accuracy, Precision, Recall and F-measure:

- **Accuracy** refers to the overall correctness of classification. It measures the ratio of correctly classified instances over the total number of instances.
- **Precision** refers to the fraction of the tweets correctly classified, for a given sentiment, over the total number of tweets classified as belonging to that sentiment.
- **Recall** refers to the fraction of tweets correctly classified, for a given sentiment, over the total number of tweets actually belonging to that sentiment. In other words, for one sentiment, this KPI is nothing different from its accuracy.
- **F-measure** is defined as follows:

$$\textit{F-measure} = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{3.3}$$

3.6.1 Binary Classification

We first run our experiment to detect the sentiment polarity of tweets. For this sake, we remove the tweets belonging to the class *"Neutral"*, and grouped the other classes into two main classes which are *"Positive"* and *"Negative"*. The former class contains tweets from the classes *"Fun"*, *"Happiness"* and *"Love"*, while the latter contains tweets from the classes *"Hate"*, *"Anger"* and *"Sadness"*. TABLE 3.5 shows the results of classification. The accuracy obtained reaches 81.3%. Noticeably, the recall of negative tweets is the highest (i.e., 83.5%), however the precision of positive tweets is the highest (i.e., 82.0%). This means that tweets which are classified as positive are mostly positive. However, tweets which have negative polarity tend to be classified more correctly as shown in the confusion matrix presented in TABLE 3.6.

The classification presents a noticeably low accuracy compared with that of our work [127]. This is because in that work, we exploited the information regarding the detailed sentiment class for unigram features and pattern features. In other words, when we extracted the features from the training and the test set, we counted unigrams belonging to the classes *"Happiness"*, *"Love"*, *"Anger"*, etc. on tweets of the training set and the test set. Furthermore, we extracted patterns

Table 3.6: Binary Classification Confusion Matrix

Class	**Classified as**	
	Positive	Negative
Positive	**5 684**	1 516
Negative	1 245	**6 306**

related to these detailed sentiments and used them to measure the resemblance between the training and the test tweets. While that was fair and acceptable given the fact that we dispose of a training set with the detailed sentiment sub-classes, for a more general case, where a person wants to classify tweets into *"Positive"* and *"Negative"*, such information might not be provided, and so the training set will contain tweets classified only as *"Positive"* and *"Negative"*. Therefore, in this work, we used the training set as a set of tweets having initially only two classes: only two unigram features are extracted, and patterns are also extracted from the training set in only two subsets: positive patterns and negative patterns.

3.6.2 Ternary Classification

Despite its importance, binary classification supposes that the given data are already known to be emotional. However, Twitter contains many tweets which have no emotional polarity such as news tweets, etc. Therefore, in this subsection we add neutral tweets as shown before in the description of our dataset. We then rely on the same set of features to classify the tweets. As described previously, no information regarding the sentiment sub-class is given or exploited here. The results obtained are given in TABLE 3.7, and the confusion matrix of classification is given in TABLE 3.8.

The obtained results show that the introduction of the third class decreases noticeably the accuracy to reach 70.1%. The new class (i.e., *"Neutral"*) presents a low accuracy and a low precision. This can be explained by the fact that the amount of training data (i.e., number of tweets) for this class is lower than that for the other classes. In addition, tweets, regardless of their content tend to be polarized (i.e., either classified as positive or as negative). This is because most of the features used, except for the pattern features, are ones that try to detect any sentimental component in a given tweet, or find any resemblance of the tweet to ones in the training set (which is highly unbalanced in favor of the sentimental classes over the neutral class).

Overall, the results obtained are promising.

Table 3.7: Ternary Classification Accuracy, Precision, Recall and F-Measure

	Accuracy	Precision	Recall	F-Measure
Positive	0.769	0.737	0.769	0.753
Negative	0.743	0.724	0.743	0.733
Neutral	0.537	0.598	0.537	0.566
Overall	**0.701**	**0.697**	**0.701**	**0.699**

Table 3.8: Ternary Classification Confusion Matrix

Class	**Classified as**		
	Positive	Negative	Neutral
Positive	**5806**	924	821
Negative	874	**5348**	978
Neutral	1196	1114	**2679**

3.6.3 Multi-Class Classification

In this subsection, we use the 7 sentiment classes that we described in Section 3.5. The classification results are given in TABLE 3.9, while the confusion matrix is given in TABLE 3.10.

Despite the number of classes, the accuracy obtained is equal to 60.2%, with a precision that reaches 60.8%. More interestingly, some sentiments seem to be easier to detect than others. In particular, tweets belonging to the class *"Love"* and those belonging to the class *"Hate"* were classified with an accuracy equal to 75.2% and 90.9% respectively. This shows that tweets belonging to these classes are easily distinguished from other classes. This might be due to the fact that other classes, such as *"Happiness"* and *"Fun"* for example are very close to each other. Therefore, many tweets of one class are classified as if they belong to the others.

The class *"Neutral"* on the other side, presents the lowest precision. Many tweets, from all the other classes were classified as neutral. While this does not go along with our observations on [127]. We believe that the main difference is that our current training set presents a cleaner reference for training. The training set used in [127] contains a lot of noise, and most of the noisy data are mainly neutral, but are used for the other classes, which resulted in a misclassification of most of the neutral tweets, and made the class *"Neutral"* present a very low recall.

3.6.4 Discussion

Classifying tweets is, to begin with, a difficult task given the very limited size of tweets. The challenges presented in Section 3.2 were tackled by many researchers, however, remain still not completely solved. With reference to this work, we can confirm that classifying tweets into separate sentiment classes is a challenging task: as mentioned above, many tweets present more than one sentiment. Therefore, a more interesting task would be quantifying the sentiments present in

Table 3.9: Multi-Class Classification Accuracy, Precision, Recall and F-Measure

Class	Accuracy	Precision	Recall	F-Measure
Fun	0.407	0.605	0.407	0.487
Happiness	0.543	0.586	0.543	0.564
Love	0.752	0.629	0.752	0.685
Nneutral	0.678	0.523	0.678	0.590
Anger	0.622	0.630	0.622	0.626
Hate	0.909	0.804	0.909	0.854
Sadness	0.521	0.653	0.521	0.580
Avg.	0.602	0.608	0.602	0.597

Table 3.10: Multi-Class Classification Confusion Matrix

Class	Classified as						
	F	Hp	L	N	A	Ht	S
Fun (F)	**1077**	274	195	756	96	34	211
Happiness (Hp)	267	**1610**	309	561	50	20	146
Love (L)	69	167	**1463**	165	16	14	51
Neutral (N)	194	443	178	**3383**	133	51	607
Anger (A)	22	48	31	268	**969**	28	192
Hate (Ht)	4	4	6	9	29	**1014**	49
Sadness (S)	147	200	144	1332	244	100	**2360**

the tweet: a tweet should be attributed more than one sentiment with different scores. The sentiments attributed will represent all the existing sentiments detected in the tweet, whereas the scores will represent the estimated weight of the detected sentiment. We strongly believe that this would allow to have a more accurate description of the sentiments in the tweet, and solves the main issue that we encountered in this work, which is the existence of multiple sentiments in the tweet.

On a related context, even though we have ran several experiments on our dataset, we cannot confirm that the values set for the parameters defined are always good ones. SENTA presents several parameters, for the different sets of features. We tried to optimize each set of parameters, related to the same family of features aside. However, this could be a non-optimal solution given the fact that the machine learning algorithm used (i.e., Random Forest) does not consider the features independently. It rather builds the model with reference to all the features combined. On the other hand, it is unpractical, and almost impossible to try all the combinations of features to derive the most adequate ones, that give the highest accuracy.

Regarding the test set used itself, its manual annotation was done on crowdflower [2]. Several annotators from different backgrounds participated in the annotation. To check the performance of the annotators, we randomly picked 300 tweets, annotated them, and compared the results with those done by the random annotators. Interestingly, the sentiment polarity (whether the tweet is positive, negative or neutral) of 91.3% of the tweets was agreed on. However, when it came to the detection of the sentiment itself, the rate of agreement dropped to 72%. However, for many of the non-agreed on tweets, we understood why the annotators decided to attribute one sentiment over another, and this goes back to the issue we highlighted earlier: the existence of multiple sentiments within the same tweet.

3.7 Conclusion

In this chapter, we have proposed a new approach for sentiment analysis, where a set of tweets is to be classified into 7 different classes. The obtained results show some potential: the accuracy obtained for multi-class sentiment analysis in the data set used was 60.2%. However, we believe that a more optimized training set would present better performances.

Throughout this work, we demonstrated that multi-class sentiment analysis can achieve high accuracy level, but it remains a challenging task. A more interesting task is to quantify sentiments

[2] https://www.crowdflower.com

present in the tweet. Therefore, in a future work, we will use the results obtained for ternary classification (which achieved an accuracy equal to 70.1%) to classify tweets into *"Positive"*, *"Negative"* and *"Neutral"*. The classified sentimental tweets (i.e., which have been classified as *"Positive"* or *"Negative"*) will then be given scores for the corresponding sentiment subclasses. This will be discussed in more details in Chapter 4. In Chapter 5, we will describe our own solution to identify all the existing sentiments.

Chapter 4

Multi-Class Sentiment Analysis: Promises & Limitations

4.1 Introduction

Over the recent years, increasing attention has been paid to the analysis of data collected from social networks and microblogging websites. This is because people tend to discuss all sorts of topics using these services; topics that might include not only their daily affairs and plans, but also some services or products they are using. That being the case, companies and organizations nowadays are trying to analyze posts and discussions of users to extract all possible useful information regarding whether or not they are interested in a given topic, the level of satisfaction of users towards products and services [58, 59], or even their intentions and expectations regarding upcoming elections, sports events, etc. [68]. One type of information that has been a hot topic of research in the last few years surrounds the identification of attitudes or opinions expressed by users in their posts towards a specific topic. This process is called "sentiment analysis".

Twitter, a popular microblogging website, offers for users a service allowing them to post and interact with short messages. It has some unique properties that make it interesting for companies, such as its openness, the length limitation on messages posted, and the wide use of hashtags. While most social networks require a connection between two users before they can access each other's posts, Twitter allows users to follow one another even if no mutual relation has been established, which makes it easy to collect information from Twitter. Furthermore, posts are limited to 140 characters, which means that messages are brief and usually include just one main piece of information. Due to the wide use of hashtags, companies can easily trace "tweets" (i.e., messages posted by Twitter users) that deal with their own products or services.

This makes the process of automatically performing sentiment analysis on tweets an interesting task: not only can tweets dealing with a given topic be collected quite easily (due to the presence of hashtags), but also the information included in a large enough number of tweets usually represents, with a certain level of fidelity, the opinion of a random, but representative, set of people towards the given topic.

However, some challenges remain in automatic analyzing tweets. According to Ghag et al. [115], these challenges include, but are not limited to, opinion object identification, maintaining opinion time and hidden sentiments identification. While most of the work done on sentiment analysis deals with the detection of the sentiment polarity of tweets (i.e., whether they are positive, negative or neutral), hidden sentiment identification refers to the identification within the tweet of actual hidden sentiments such as anger, happiness, disgust and joy.

In the previous chapter, we have proposed an approach to perform multi-class sentiment analysis on Twitter. The target of the proposed approach was indeed to find the precise sentiment in a given piece of text (a tweet in this case). This has proven to be a challenging task.

In the current chapter, we investigate this challenge in more details and present the obstacles that render it difficult to identify the actual sentiment of a given tweet. We perform a multi-class sentiment analysis of tweets and discuss how the number of sentiment classes impact the classification results. We propose a new model to represent sentiments, and use it to show the relationships between the different sentiments and to explain why the task of multi-class sentiment analysis is inherently difficult.

The remainder of this chapter is structured as follows. In Section 4.2 we present our motivation for this work and discuss some previous research dealing with the multi-class sentiment

analysis. In Section 4.3 we describe the data sets we used for this work, and present the procedure of extraction of features from tweets. In Section 4.4 we present our different experiments and the obtained results. In Section 4.5 we introduce our model for representing sentiments and the relation between them, discuss the classification results and analyze the effect of the number of classes on the classification. Finally, Section 4.6 concludes this work.

4.2 Motivations and Related Work

4.2.1 Motivations

The binary classification into positive and negative of posts collected from online web-sites, social networks or microblogging services is an interesting approach that allows companies to estimate the level of satisfaction of users, or their expectations towards an upcoming service. However, determining whether a tweet is positive or negative might not always be sufficient.

Take the following two tweets:

- "Noooooooooo! My iPhone glass cracked :("
- "Damn damn.. no iPhone support for windows XP x64. There are some workarounds, but I can't figure this out."

The difference between these tweets, in terms of sentiment and even interpretations of what the users want, can be easily seen. Both tweets are obviously negative, but in different respects. As a matter of fact, for the company producing the product that is the subject of these tweets, the information that they can extract from each needs to be treated differently. While in the first tweet the user is expressing a sentiment of sadness because of physical damage to the product, in the second tweet the user is expressing anger and frustration due to the product's lack of the support for a particular operating system. The company would probably be best advised to prioritize the problem raised in the second tweet; however, in general, both tweets are important in different ways, and the difference between them needs to be emphasized.

Therefore, the detection of the real sentiment within a tweet is of great importance. Gagh et al [115] nominated "hidden sentiments identification" as one of the most challenging tasks when performing sentiment analysis. They defined it as going beyond the identification of the polarity to the detection of the specific sentiment shown, such as hate, disgust or anger.

While some works have tried to go beyond the binary or ternary classification of tweets, most of these have divided the positive and negative classes into subclasses that focus mainly on the intensity of the sentiment polarity (e.g., "very positive", "positive", "mostly positive" and "very negative", "negative", "mostly negative"); other works have dealt with the task of multi-class classification [120–123], but in a different context as we will describe below.

That said, the current work revolves around two main axes:

- The multi-class classification of tweets; and
- The impact of the number of classes on the classification performance.

4.2.2 Related Work

With the growth of social network and microblogging websites, people began to openly discuss their opinions, thoughts and even daily affairs online. This has attracted researchers to study human behaviors online, collecting and summarizing data posted by people daily. Twitter, for the reasons stated above, has attracted most of this attention. Some of the research on tweets has dealt with the form of the data, the use of slang and how these develop over the time, the use of emoticons and the nature of tweets themselves [57, 81].

However, most of the work has dealt with the actual content of tweets. While the majority have focused on classifying tweets depending on their sentiment polarity (positive or negative), whether the topic of the tweets is a product [58], a service [59] or democratic elections [68], more advanced works have gone deeper into the classification, and focused on assessing the level of sentiment strength (e.g., "very negative", "negative", "mostly negative", "neutral", "mostly positive", "positive" and "very positive"), or even attributing sentiment intensity scores to different texts [82, 118, 119].

Nevertheless, classification into multiple sentiment classes has been the subject of multiple recent works. Lin et al. [120, 121] proposed an approach that classifies documents into reader-emotion categories. They studied the classification of news articles into different sentiment classes representing the emotions they trigger in their readers. Their work mainly differs from other literature in focusing more on what the reader would feel while reading the article rather than what the writer was feeling while writing it. Similarly, Ye et al. [122] studied the problem of emotion detection in news articles from the reader's perspective. Given the limitation of classification into single-labeled classes, they investigated a multi-label classification. Their work falls into the same category as that of Bouazizi et al. [127] who investigated the problem of sentiment quantification, and attributed more than one sentiment class to posts extracted from Twitter. Liang et al. [123] proposed a system that recommends emoticons to users while they are typing their texts, depending on the content of what they are writing.

In the context of multi-class classification, we proposed in the previous chapter a scalable approach that allows the classification of tweets into different sentiment classes. While our approach can be applied to any number of sentiment classes, we restricted our study to seven. The tool we developed is used here to extract features from the tweets, and Weka [105] is used to perform the multi-class classification.

4.3 Multi-Class Classification: Experiment Specifications

In this section, we will show the empirical results of our experiments on two data sets. Despite the fact that these are purely empirical results, we will later use them as a starting point to identify several challenges that make the task of multi-class classification difficult and, in some cases, almost impossible.

Table 4.1: Structure of the Dataset Used

Class	Training set	Test set
Fun	3000	2643
Happiness	3000	2963
Love	3000	1945
Neutral	3000	4989
Sadness	3000	4528
Anger	3000	1558
Hate	3000	1115
Total	**21 000**	**19740**

4.3.1 Problem Statement

Given a set of tweets, we study the possibility of classifying them into different sentiment classes. From each tweet, we extract different sets of features, refer to a manually annotated training set and use machine learning to perform the classification.

Other than the classification itself, which has been detailed in the previous chapter, we study the impact of the number of sentiment classes on the classification performance (i.e., accuracy, precision and recall). We analyze the results of the different experiments and conclude with the limitations that make multi-class classification a difficult task.

4.3.2 Data Sets Used

For our experiments, we used two data sets composed of posts extracted from Twitter that had been manually annotated into 7 different sentiment classes. The 7 different sentiments present 3 pairs of opposite sentiments (i.e. [Love vs Hate], [Happiness vs Sadness] and [Fun vs Anger]) in addition to the sentiment class [Neutral].

The structure of the data sets is given in Table 4.1.

We used the data sets either entirely or in part depending on the requirements of each experiment, so will explicitly mention the parts of the data set used in each case.

4.3.3 Features Extraction

To extract the desired features from the different tweets, we used SENTA. While SENTA offers the possibility to extract a multitude of features, we did not use all of them in this work: in this sub-section, we briefly introduce the features we did use. The detailed significance of each feature is described in the previous chapter.

Sentiment features Sentiment features rely on the sentiment polarities of different components of the tweet. Following are the sentiment features we extracted:

- The number of both positive and negative words,
- The number of both highly emotional positive and highly emotional negative words,

- The ratio of emotional words,
- The number of both positive and negative emoticons,
- The number of both positive and negative slang words.

Punctuation features With the exception of exclamation marks, punctuation does not usually reveal any sentiments explicitly; nonetheless, the excessive use of some forms of punctuation (question marks, exclamation marks, etc.) is a good indicator of the presence of a strong sentiment. Therefore, the following features are extracted:

- The number of full stops,
- The number of exclamation marks,
- The number of question marks,
- The total number of words, and
- The number of quotation marks.

Syntactic and stylistic features These are features related to the use of words and expressions in the tweet. The following features are extracted:

- The number of particles,
- The number of interjections,
- The number of pronouns,
- The use of negation, and
- The number and use of uncommon words.

Semantic features Semantic features are features that focus on the meanings in language or the logic inside of sentences. The following semantic features are extracted:

- The use of opinion words,
- The use of highly sentimental words,
- The use of words expressing uncertainty,
- The use of the passive form of speech.

Unigram features These are features collected with reference to a prebuilt dictionary containing words that are highly correlated with the different sentiment classes. In each tweet, we check whether any of the words in the dictionary are present; if so, the feature corresponding to the sentiment of that word is incremented by 1. In other words, these features count the existence of words related to each sentiment in the tweet. Therefore, 6 features are extracted (for the 6 sentiments other than Neutral). The prebuilt dictionary is the same as that used in [129].

Pattern features Patterns are used as a complementary set of features to detect what unigrams cannot detect. In most of the cases, sentimental words are sufficient indication of the sentiment present in a sentence, whereas in other cases a person can employ some specific longer expressions to express a sentiment. Therefore, the main contribution of pattern features is to detect these longer expressions. Pattern features are extracted from the training set. They are exclusive to each sentiment polarity (i.e., if a pattern exists in two sentiments of opposite polarities, it is excluded from the lists of patterns of both sentiments). A resemblance function has also been defined to measure how close a given tweet is a pattern. As mentioned above, the procedure of the extraction of pattern features, as well as the other sets of features, is detailed in the previous chapter and in [129]. The selection of features as well as the optimization of the parameters related to them is therefore outside of the scope of this chapter.

However, we will discuss pattern and unigram features in more details in a later section when we introduce our model for representing the sentiment space.

4.3.4 Experiment Specifications

As mentioned above, our data sets contain tweets fitting into 7 sentiment classes. The sentiments taken into account are divided into 3 pairs of opposite sentiments and an additional single sentiment: [Fun vs Anger], [Love vs Hate], [Happiness vs Sadness] and [Neutral]. For convenience, in what follows, each sentiment class will be referred to by its name or by its abbreviation:

- Fun (F),
- Anger (A),
- Happiness (Hp),
- Sadness (S),
- Love (L),
- Hate (H), and
- Neutral (N).

We used the Random Forest classifier [109] in our experiments, and applied 4 Key Performance Indicators (KPIs) for evaluating the classification: Accuracy, Precision, Recall and F-measure:

- **Accuracy** refers to the overall correctness of classification, measuring the ratio of correctly classified instances over the total number of instances.
- **Precision** refers to the fraction of the tweets correctly classified, for a given sentiment, over the total number of tweets classified as belonging to that sentiment.
- **Recall** refers to the fraction of tweets correctly classified, for a given sentiment, over the total number of tweets actually belonging to that sentiment. In other words, for a single sentiment, this KPI is equivalent to its Accuracy.

Table 4.2: Accuracy, Precision, Recall and F-Measure of the Binary Classification

Class	Accuracy	Prec.	Recall	F-Measure
Fun	80.1%	88.4%	80.1%	84.0%
Anger	82.2%	70.9%	82.2%	76.1%
Fun vs Anger	**80.9%**	**81.9%**	**80.9%**	**81.1%**
Happiness	81.9%	74.3%	81.9%	77.9%
Sadness	81.5%	87.3%	81.5%	84.3%
Happiness vs Sadness	**81.6%**	**82.2%**	**81.6%**	**81.8%**
Love	93.8%	98.9%	93.8%	96.3%
Hate	98.1%	90.1%	98.1%	93.9%
Love Vs Hate	**95.4%**	**95.7%**	**95.4%**	**95.4%**

- **F-measure** is defined as follows:

$$\textit{F-measure} = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{4.1}$$

4.4 Experimental Results

To evaluate the impact of the number of classes on the classification performance, we measure the KPIs mentioned above for different numbers of sentiments.

4.4.1 Two Sentiment Classes

In our first experiment, we run the binary classification of the different pairs of sentiments, each pair apart. To recall, the sentiments are chosen so that they fit into several pairs of approximately opposite sentiments. The term *approximately* is used here to highlight the fact that, even though we treat them as pairs of opposite sentiments, this assumption is not very accurate: this is discussed in details below.

That being said, in this first round of experiments, we divide our data set into sub-sets, each contains only the tweets of a pair of sentiments. Additionally, the term "vs" used in the following in the format [A vs B], where A and B are two sentiments, means that the sentiment A is checked against the sentiment B. In other words, the classifier is trying to classify the tweets into on of the two classes A and B". The classification Accuracy, Precision, Recall and F-measure of the binary classification of pairs of sentiments are given in Table 4.2.

The binary classification of the different pairs of sentiments presents good Accuracy, Precision and Recall. All the classification tasks achieved an Accuracy higher than 80%, with the pair [Love vs Hate] having the highest (95.4%). The average Accuracy of classification is 86.0%.

4.4.2 Three Sentiment Classes

After adding the class Neutral as a third class to the same sets we used in the previous sub-section, the Accuracy of classification dropped remarkably, as shown in Table 4.3.

Table 4.3: Accuracy, Precision, Recall and F-Measure of the Ternary Classification

Class	Accuracy	Prec.	Recall	F-Measure
Fun (F)	50.0%	63.2%	50.0%	55.8%
Neutral (N)	74.5%	73.6%	74.5%	74.1%
Anger (A)	70.9%	54.0%	70.9%	61.3%
(F) vs (N) vs (A)	**66.9%**	**67.3%**	**66.9%**	**66.7%**
Happiness (Hp)	68.2%	64.0%	68.2%	66.0%
Neutral (N)	69.3%	62.5%	69.3%	65.8%
Sadness (S)	59.2%	70.7%	59.2%	64.4%
(Hp) vs (N) vs (S)	**65.4%**	**65.8%**	**65.4%**	**65.3%**
Love (L)	82.0%	75.4%	82.0%	78.6%
Neutral (N)	84.8%	92.2%	84.8%	88.4%
Hate (Ht)	93.0%	77.2%	93.0%	84.3%
(L) vs (N) vs (Ht)	**85.3%**	**86.1%**	**85.3%**	**85.5%**

Table 4.4: Accuracy, Precision, Recall and F-Measure of the 4-Class Classification

Classes	Accuracy	Prec.	Recall	F-Measure
(F) - (A) - (Hp) - (S)	60.4%	60.7%	60.4%	60.2%
(F) - (A) - (L) - (Ht)	74.9%	75.9%	74.9%	74.5%
(Hp) - (S) - (L) - (Ht)	74.5%	75.2%	74.5%	74.7%

While the pair [Love vs Hate] maintained a high Accuracy, Precision and Recall levels, the two other pairs were highly impacted by the introduction of the third class. In particular, the class Fun showed a decrease of Accuracy and Precision from 80.1% and 88.4% to 50.0% and 63.2%, respectively. This decrease will be addressed later, but, in brief, we suspect this to be due to the low number of sentimental words collected for unigram features for this sentiment, and its proximity to the class Neutral. The overall average Accuracy with Neutral added is 72.5%.

4.4.3 Four Sentiment Classes

For this set of experiments, we discarded the class Neutral and tried the different possible combinations of pairs of sentiments. For convenience, we kept only the overall classification performance for each experiment. The results are given in Table 4.4.

Again, the overall Accuracy, Precision, Recall and F-measure are lower than those of the ternary classification. While the pair [Love vs Hate] achieves the highest Accuracy, the classes Happiness and Fun present low Accuracy and Recall. These two classes were confused with each other, the reason for which can easily be seen from the nature of the two classes themselves: they are quite similar to each other, with most of the sentimental words used to express happiness also used to express fun and enjoyment. The overall average Accuracy is 69.9%.

Table 4.5: Accuracy, Precision, Recall and F-Measure of the 5-Class Classification

Classes	Accuracy	Prec.	Recall	F-Measure
(F)-(A)-(Hp)-(S)-(N)	54.4%	55.4%	54.4%	54.1%
(F)-(A)-(L)-(Ht)-(N)	66.9%	66.9%	66.9%	66.3%
(Hp)-(S)-(L)-(Ht)-(N)	64.1%	64.6%	64.1%	63.8%

Table 4.6: Accuracy, Precision, Recall and F-Measure for the 6-Class Classification of tweets of 6 Classes

Class	Accuracy	Precision	Recall	F-Measure
Fun	39.1%	56.8%	39.1%	46.3%
Anger	59.3%	52.4%	59.3%	55.6%
Happ.	57.6%	54.6%	57.6%	56.0%
Sadness	63.9%	68.6%	63.9%	66.1%
Love	71.1%	55.5%	71.1%	62.3%
Hate	86.8%	73.2%	86.8%	79.4%
Overall	**60.4%**	**60.5%**	**60.4%**	**60.0%**

4.4.4 Five Sentiment Classes

Keeping the same combinations we used in the 4-class classification, we added the class Neutral and re-ran the classification. The results are given in Table 4.5.

The same observations made in the previous sub-section are present again: the sentiment Fun was rather confused with the classes Happiness and Neutral. The introduction of the new class decreased the overall average Accuracy to 61.8%.

4.4.5 Six Sentiment Classes

For this experiment, we used the entire data set, except fir the tweets of the class Neutral. The performance of the classification is given in Table 4.6.

The class Fun still presents the lowest Accuracy and Recall, with most of its tweets misclassified. The tweets of the class Happiness present the second lowest Accuracy and Recall. The pair of sentiments [Love vs Hate] presents the highest Accuracy and Recall due to the fact that these sentiments are easily distinguishable from each other, and also from the rest of the sentiments.

The overall average Accuracy is 60.4%, which presents no major difference from that of the classification into 5 sentiments.

4.4.6 Seven Sentiment Classes

Finally, we ran the classification using the entire data set. The performance of classification into sentiment classes is given in Table 4.7.

The same trend seems to hold, with the overall Accuracy of 60.2% slightly lower compared to that of the previous experiment. Again, the classes Love and Hate present the highest Accuracy.

Table 4.7: Classification Accuracy, Precision, Recall and F-Measure for the Classification of tweets of 7 Classes

Class	Accuracy	Precision	Recall	F-Measure
Fun	40.7%	60.5%	40.7%	48.7%
Anger	62.2%	63.0%	62.2%	62.6%
Happ.	54.3%	58.6%	54.3%	56.4%
Sadness	52.1%	65.3%	52.1%	58.0%
Love	75.2%	62.9%	75.2%	68.5%
Hate	90.9%	80.4%	90.9%	85.4%
Neutral	67.8%	52.3%	67.8%	59.0%
Overall	**60.2%**	**60.8%**	**60.2%**	**59.7%**

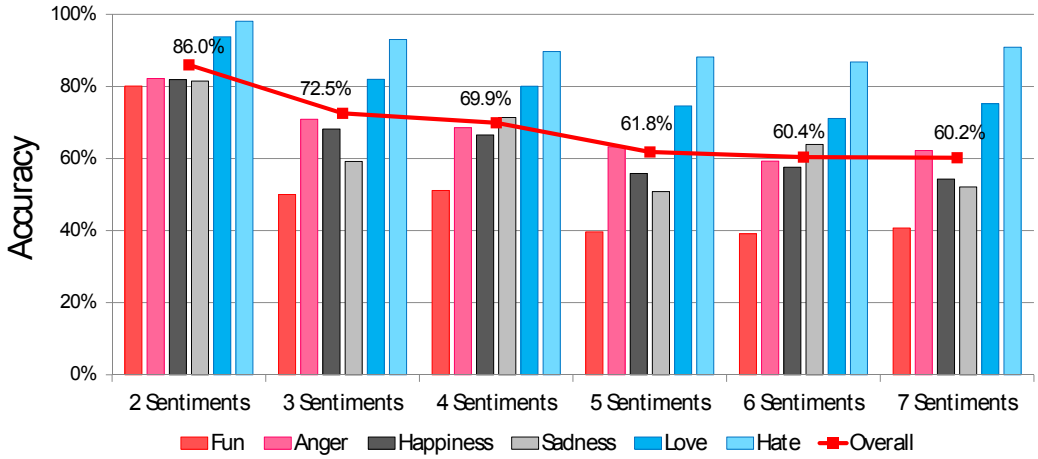

Figure 4.1: Overall classification Accuracy and Individual Sentiment Classification Accuracy for Different Number of Sentiment Classes

4.5 Analysis and Discussion of the Results

4.5.1 Observations

Because it is the most important indicator of good classification, we focus mainly on the level of Accuracy. For each different number of sentiment classes, the level of accuracy for the different sentiments is shown, alongside the overall Accuracy, in Fig. 4.1.

Obviously, classification Accuracy decreases with an increase in the number of sentiments. However, the decrease rate slows. Starting from 5 sentiment classes, Accuracy starts to be almost unchanging. While this is true for the current dataset, we cannot generalize this behavior, nor determine whether it will maintain the same rate if we continue to add more sentiment classes. We suggest that the addition of an extra pair of sentiments (e.g., [Enthusiasm vs Boredom]) would help to clarify this point.

On a side note, the slight improvement in Accuracy of some sentiment classes (e.g., Fun and Anger) in the 7-class classification over that in the 6-class classification does not mean that adding a seventh class makes it easier to detect these sentiments; rather it is mainly due to how the

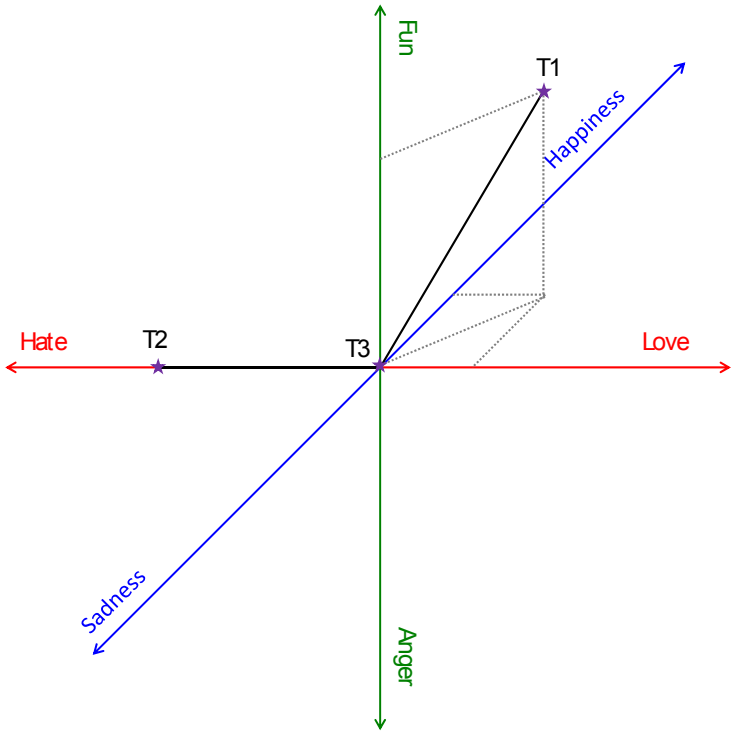

Figure 4.2: First Representation of the Sentiment Space

classifier works. In other words, the classifier's rules are built so that the overall Accuracy is the highest. This can make the rules defined for 6 sentiment classes different from those of 7 sentiment classes, which results in this slight enhancement of some sentiments over others. Despite this, we believe that the overall trend still reflects the behavior of classification Accuracy as a function of the number of sentiments.

In addition, the pair of sentiment classes [Love vs Hate] seem to be the least prone to have their Accuracies decrease regardless of the number of classes, whereas sentiments such as Fun and Happiness seem to be easily confused with each other and with other sentiments, such that many of these tweets are misclassified.

4.5.2 Analysis

Sentiment Space Representation

At a first glance, we could imagine sentiments as defined in this work as pairs of opposite sentiments, as we initially intended. Accordingly, we could define a space with $n/2$ different dimensions, where each dimension has two ends representing the opposite sentiments. Fig. 4.2 shows this possible representation of the sentiments for 3 pairs of sentiments (the seventh is the sentiment Neutral). Obviously, the farther a point from the origin, the stronger the sentiment is. A short text (such as a tweet), in this space, could be represented as a point, or a vector starting from the origin whose projection on each of the dimensions shows how strong it is. In the same figure, the point $T1$ represents a text showing the sentiments [Happiness, Love, Fun], while the point $T2$ represents a text showing only the sentiment Hate, and the point $T3$ represents a Neutral text.

However, in practice, and based on our observations on the data set, this representation has several flaws. One flaw is that it suggests that the dimensions are orthogonal. This is not always

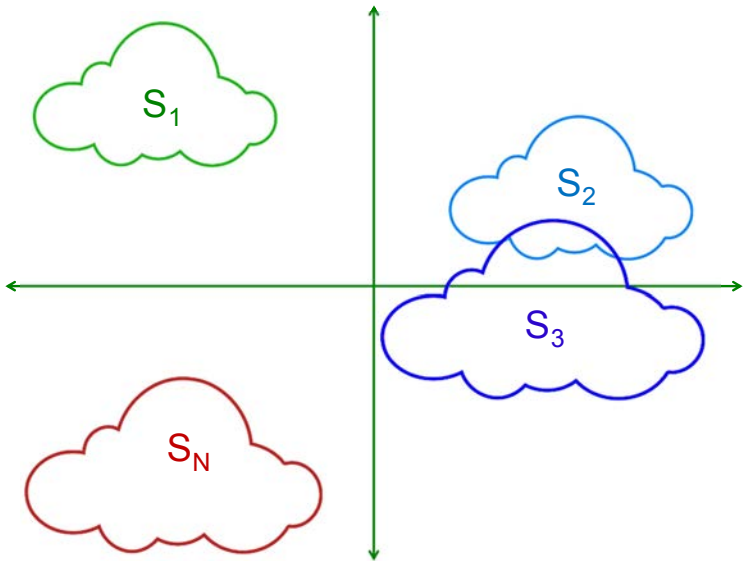

Figure 4.3: Second Representation of the Sentiment Space

true, because some sentiments are highly correlated and are not sufficiently independent from each other to be considered orthogonal, as we discuss below. Also, the class Neutral in this representation is restricted to an infinitesimal region near the origin.

A more reasonable and practical way to represent the sentiments in a given space is to have each sentiment represented by a cloud centered on a specific point. This is more natural as it suggests the texts are by default neutral, unless they are in or near the given region of a particular cloud (which represents a sentiment). In addition, the dimensions in this space could represent any information, and does not need to be sentiment related. In Fig. 4.3, we show an example of this representation in a 2-dimensional space. Some sentiments are obviously close to each other such as sentiments S_2 and S_3, and therefore share a common area in the space.

However, in such a representation, it is not clear how a given text could be presented in such a space. In addition, the cloud representation does not give an accurate description of where the sentiment is at its strongest. For these reasons, the representation is slightly modified in the current work as follows: a cloud is denser near the center and fades away as we get farther from it. In other words, a text located at the edges of the cloud shows less of the sentiment.

More importantly, this representation could allow us to define what we call the distance between two sentiments. Unlike the case of multi-dimensional representation, sentiments here can be correlated, and it is possible to define metrics to measure the distance between any two sentiments, for example the distance between the centers of the two corresponding clouds. In this work, we will refer to a cloud corresponding to a given sentiment S_i as Ω_i.

Given two different sentiments, S_i and S_j, they each could share some resemblance, through similar patterns or expressions, or a set of words that can be used for either of them. The word "fun" in the expression "@user I'm having soo much fun here!" for example shows sentiments of Fun and Happiness.

Distance Between Two Sentiments

A simple way to define the distance between two sentiments S_i and S_j is as follows: suppose there is a set of words, expressions or patterns that are commonly used to show each of the two. We will refer to the number of words, expressions or patterns that are used to express S_i as N_i, and those that are used to express S_j as N_j. The two sentiments share n words, expressions or patterns to express them (e.g., the word "upset" could be used to show both Anger and Sadness). The distance between the two sentiments could be expressed as follows:

$$D(S_i, S_j) = 1 - 2 \cdot \frac{n}{N_i + N_j}. \tag{4.2}$$

The distance is maximal (i.e., equal to 1) when the two sentiments share nothing in common, and is minimal (i.e., equal to 0) when they are identical. This representation is efficient but does not faithfully reflect how we defined the sentiment clouds, as there is no way to tell whether or not a point is close to the center of the cloud.

Thankfully, in the particular case of words (i.e., unigrams), we could derive an even more precise and meaningful expression for the distance. To recall, unigrams are simple words that are extracted in the context of unigram Features using SENTA. SENTA extracts unigrams as follows:

1. For each sentiment, the user defines a small set of words that he judges as highly correlated with the given sentiment;

2. SENTA refers to WordNet to extract the hyponyms of the words defined by the user and adds them to the list;

3. SENTA extracts the hyponyms of the new words and adds them to the list, keeping a single copy of each word; then

4. SENTA keeps repeating Step 3 several times according to the parameters set by the user.

The final list of words for a given sentiment will have the following format:

$$U(S_i) = \{w_1, w_2, \cdots, w_{n_i}\}. \tag{4.3}$$

However, the words that have been added manually by the user are more trustworthy and more likely to be highly correlated with the sentiment than the ones that are extracted later on. This is because hyponyms lose part of the meaning of their hypernyms as explained in [127].

In the following, we will suppose that we keep track of the depth at which each unigram is found for the first time. So words that have been introduced by the user are considered to have been found at depth 0, whereas words that are hyponyms of the words introduced by the user are considered to have been found at depth 1, and so on.

In this context, the unigrams of a given sentiment could be seen as a cloud with several layers as shown in Fig. 4.4, where unigrams closer to the center of the cloud are ones extracted at an earlier stage (i.e., having a lower depth value). At the very center of the cloud are the words that are used to name the sentiment along with their direct derivations (e.g., for the class Happiness, these are "happiness", "happy", etc.).

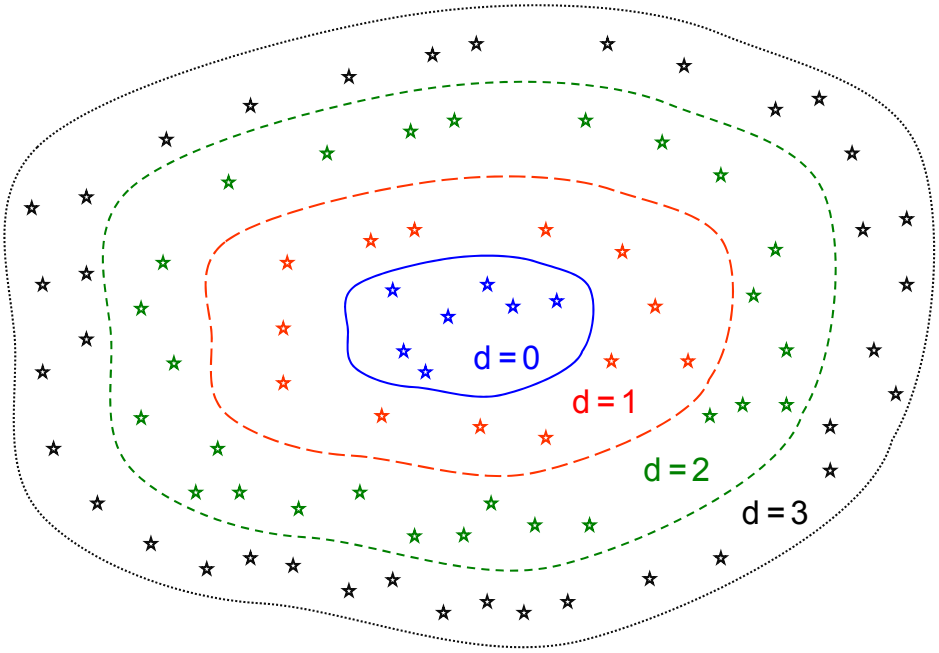

Figure 4.4: The Multiple Layers of a Single Cloud of a Given Sentiment

Following the same logic, we could also represent two sentiments in the same space as two clouds sharing some of their unigrams, as shown in Fig. 4.5.

With that being said, given the sentiment S_i, we will refer to the maximum depth selected by the user as d_{max}, a given depth as a or b, and $N_{(i,d)}$ will equal the number of new words added to the sentiment S_i at the depth d. The seed words are those that have a depth equal to 0.

Therefore, returning to the definition of the distance between two sentiments, we express it as follows:

$$D(S_i, S_j) = \sum_{a=0}^{d_{max}} \sum_{b=0}^{d_{max}} \delta_{(a,b)} \cdot \left(1 - 2 \cdot \frac{n_{(a,b)}}{(N_{(i,a)} + N_{(j,b)})}\right) \tag{4.4}$$

where $n_{(a,b)}$ is the number of common unigrams of the sentiments S_i at the layer a and S_j at the layer b, and $\delta_{(a,b)}$ is a coefficient highlighting the weight of the common unigrams between two different layers (a and b) of the two clouds. Obviously δ is symmetric (i.e., $\delta_{(a,b)} = \delta_{(b,a)}$), and all of the coefficients $\delta_{(a,b)}$ should sum up to 1.

Correlation Between Different Sentiments

Now that the distances between the clouds are defined, we define the question **(Q1)**: "Is it possible to identify which sentiments are more likely to co-occur or to be highly correlated?". The short answer for this question is *"yes"*. However, below we realistically measure the distances between sentiments in our data set, and identify which sentiments are likely to co-occur within a tweet.

Another interesting output of the current representation of sentiments is that, given an expression (or a unigram in this case), we can also tell how far it is from each cloud and what sentiment it conveys. While we have limited our study in this chapter to unigrams, it is always possible to

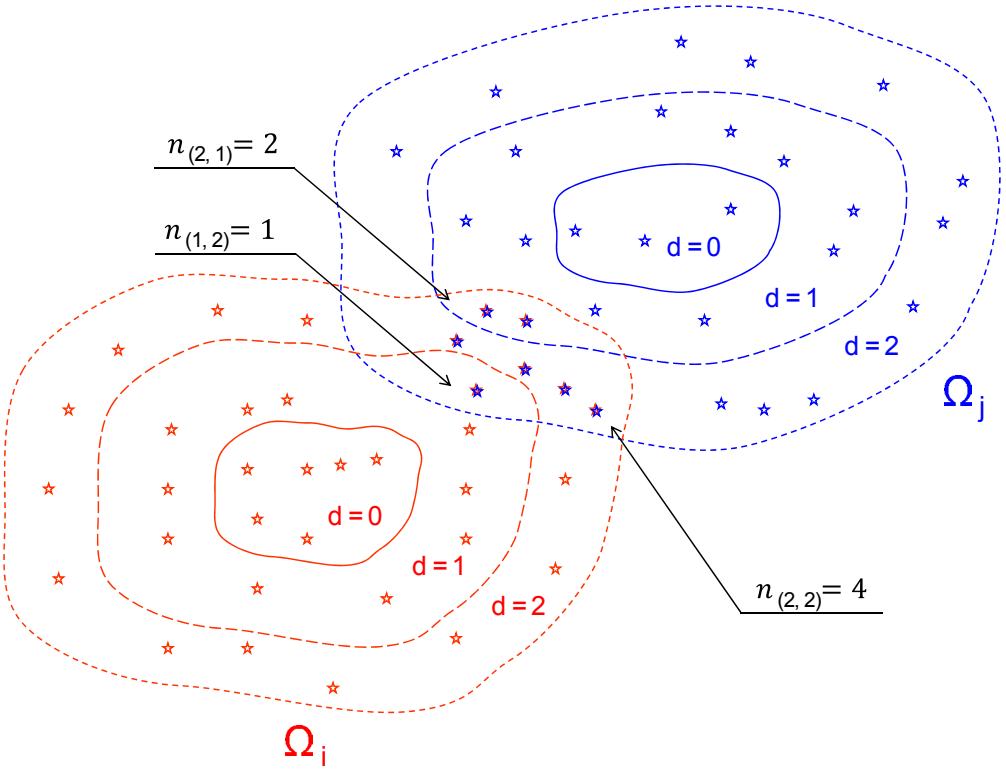

Figure 4.5: The Intersection Between Two Clouds with Several Layers each

extend it to longer n-grams, patterns or even full sentences. This leads us to our next question **(Q2)**: "Given a sentence (i.e. a tweet in our case), is it possible to attribute different scores to show the distance the sentence has from the sentiment?", which can be reformulated into **(Q2')**: "Is it possible to attribute different scores showing the strength of each of the sentiments within the sentence?". This can be simply seen as representing the sentence by a point in the space introduced above, where the closer that point is to a cloud, the stronger the sentiment corresponding to the cloud is in the sentence. In other words, the score can be any increasing function of the inverse of the distance.

In the current work, we briefly introduce the concept of quantification, which we explain in more detail elsewhere. By quantification, we refer to the attribution of sentiment scores to a given text, where each score represents how strongly the sentiment is present in the text. The scores are rarely equal to 0, so we define a certain threshold T_L below which a sentiment score is considered too low, and the corresponding sentiment is thereby considered non-existent or negligible. That being said, in the current work, given a tweet T, and a set of N sentiments $S_1, S_2, \cdots, S_N$, we extract 2 different sentiment scores for each of these sentiments using the two sets of features qualified as unigram features and pattern features, as explained in [129] and which we will refer to as "unigram score" (s^u) and "pattern score" (s^p), respectively.

In the case of unigram scores s^u, they are generated simply by counting the number of unigrams generated by SENTA for each sentiment present in the tweet.

As for pattern scores, these are computed slightly differently: SENTA, as explained above, allows for extracting writing patterns from the training set (or eventually any manually annotated set, which we will be referring to as the "pattern set") that are unique to each sentiment. These

patterns could have different lengths. Given a tweet T and a pattern p extracted from the pattern set for a sentiment S_i and whose length is equal to L_j (i.e., the j^{th} length), we have used the following resemblance function defined in the previous chapter:

$$res(p,T) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order,} \\ \alpha, & \text{if all the words of the pattern appear in the tweet in the correct order but with other words in between,} \\ \gamma \cdot n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appears in the tweet.} \end{cases}$$

Patterns of different lengths and for different sentiments are saved into different lists. We then have defined a certain number of features we qualified as "pattern features", each in the following format:

$$F_{ij} = \sum_{k=1}^{knn} res(p_k, T) \tag{4.5}$$

where p_k are patterns that most resemble the tweet T, and knn is a parameter referring to the number of patterns to be considered. These features are used to attribute a pattern score: suppose that we have set the minimal pattern length to L_{min} and the maximal pattern length to L_{max}. We will refer to $M = L_{max} - L_{min}$ as the number of lengths. The pattern score s_p will be defined as follows:

$$s_p = \sum_{j=1}^{M} \left(\beta_j \cdot \sum_{k=1}^{knn} res(p_k, T) \right) \tag{4.6}$$

where β_j is a weight given to each length. Obviously, the longer the pattern is, the more important its weight should be.

Using both the unigram scores and the pattern scores, we can attribute scores showing the strength of the different sentiments within a tweet. However, this step falls outside of the scope of the current chapter, in which our main goal is to model sentiments in way that makes it possible for a given text to have multiple sentiments, and to measure the distance between the text and a given sentiment, as well as the distance between different sentiments.

In the current work, we have used both unigram scores and pattern scores to define the distance between the different sentiments. We will use equations (4.2) and (4.4) to measure the distances between sentiments using pattern scores and unigram scores, respectively.

In particular, regarding equation (4.4), it is important to mention that we have restricted our extraction of unigrams to a maximum depth $d_{max} = 3$. Without loss of generality, we define and will be using the values of the different combinations of a and b shown in Table 4.8:

The distance measures between the different sentiment classes will be referred to as D_U and D_P for unigrams and patterns, respectively. For our data set, these distances are displayed in Table 4.9 and Table 4.10.

Table 4.8: Values of $\delta(\mathbf{a}, \mathbf{b})$ for different depths

$(\mathbf{a}, \mathbf{b})$	$\delta(\mathbf{a}, \mathbf{b})$
(0,0)	$1/2^{(*)}$
(0,1), (1,0)	1/8
(1,3), (2,2), (3,1)	1/24
(1,4), (2,3), (3,2), (4,1)	1/64
(2,4), (3,3), (4,2)	1/96
(3,4), (4,3), (4,4)	1/128

Table 4.9: Distance Between the Different Sentiments as measured with D_U

	(F)	(Hp)	(L)	(N)	(A)	(S)	(Ht)
(F)	0	0.61	0.85	-	1	1	1
(Hp)	0.61	0	0.79	-	1	1	1
(L)	0.85	0.79	0	-	1	1	1
(N)	-	-	-	0	-	-	-
(A)	1	1	1	-	0	0.83	0.71
(S)	1	1	1	-	0.83	0	0.84
(Ht)	1	1	1	-	0.71	0.84	0

Table 4.10: Distance Between the Different Sentiments as measured with D_P

	(F)	(Hp)	(L)	(N)	(A)	(S)	(Ht)
(F)	0	0.95	0.94	0.98	1	1	1
(Hp)	0.95	0	0.95	0.99	1	1	1
(L)	0.94	0.95	0	0.99	1	1	1
(N)	0.98	0.99	0.99	0	0.99	0.99	0.99
(A)	1	1	1	0.99	0	0.96	0.97
(S)	1	1	1	0.99	0.96	0	0.96
(Ht)	1	1	1	0.99	0.97	0.96	0

As expected, and under both metrics, the class Fun has the smallest distance to the class Happiness. Especially when using the metric D_U, these two classes have by far the smallest distance. This means that these two sentiments have a lot in common, and therefore can be easily confused. In addition, using the metric D_P with reference to the class Neutral, the class Fun has a relatively small distance compared with all other sentiments.

It is also noticeable that, overall, the positive sentiments have a smaller distance from on another, compared to that of the negative ones. This translates into a lower Accuracy and Precision for positive sentiments than negative ones.

4.5.3 Discussion

From our observations and analysis, we can confirm that the task of multi-class sentiment analysis presents many challenges. To begin with, the presence of multiple classes, in general, makes it harder for a given classifier to define the borders between different classes. Moreover, in the case of text sentiment analysis, different sentiments have much in common, and the actual border between two sentiments, examplified by Happiness and Fun, is somewhat unclear. In other words, it is sometimes difficult even for humans to detect the difference. In addition, the more classes there are, the less patterns can be extracted for an individual class. Nevertheless, some sentiments can coexistent, and a certain sentence can contain more than one sentiment. Given the following tweet: *"Man, I'm having sooo much fun here. Glad my whole family came with me. It's just amazing!"*, the author explicitly presents enjoyment and happiness. This makes it hard to attribute the tweet to one sentiment class.

This leads to an important conclusion: even though many texts can be classified into one of multiple sentiment classes, it might be a more interesting task to detect all of the sentiments that exist in a tweet, and to attribute a certain score to each sentiment class, reflecting its weight.

4.5.4 Multi-Class Classification: Challenges

To recapitulate, here we list the main challenges that make multi-class sentiment analysis difficult. We illustrate with tweets from our data set that have been misclassified and explain the reasons for the misclassification.

Presence of Negation Handling negation has always been an issue when it comes to sentiment analysis. Not only is it hard to tell whether the presence of negation is a polarity switcher or not, but also, in the case of multi-class classification, switching polarity does not automatically indicate that the sentiment of the tweet is the opposite of that negated. This can be seen in the following tweet: "Well guess what?? I'm not really happy with what he said anyway!". The word "happy" is a word that is used usually to express sentiments of Happiness. On the other hand, as stated in the previous subsections, Happiness and Sadness are supposedly a pair of opposite sentiments. However, the negation in this tweet did not show the sentiment of Sadness which has been reported by the classifier, but rather the sentiment of Anger.

Context Dependency Tweets are often intended as replies to other tweets, making them highly context dependent. We read the tweet *"I remember someone saying it's gonna be fun.."* as a

Neutral tweet, but some of the annotators labelled the tweet as showing sentiments of Anger. This is because they assumed the user is showing dissatisfaction towards an event that was supposed to be funny, but in actual fact was not. However, while this assumption can be made by a human, machines are not able to imagine such scenarios and extract the actual sentiment out of it.

Polysemy Several words in English, as with other languages, have multiple meanings depending on their context. These meanings could be similar or totally unrelated. However, for multi-class sentiment analysis, even the similar meanings could indicate different emotions. An example is the word "mad", the meanings of which include angry as well as crazy. Furthermore, craziness often points to something being good or funny. *"Mad"* can also be used as an adverb meaning "very", as can be seen in the following tweet: "It was mad fun man!". This tweet was classified as showing sentiments of Anger, despite the presence of two sentimental words. However, the tweet could have easily been detected as belonging to the class Fun if the PoS-Tagger could identify the word "mad" as an adverb.

Presence of Multiple Sentiments Even though tweets are short in length and limited to a certain number of characters (i.e., 140 characters per tweet), they can be poly-sentimental in the sense of containing more than just one sentiment. As a matter of fact, a large number of the tweets we have in our data set present multiple sentiments, as illustrated with these tweets:

- "I'll miss you sooo much! I can't believe you have to leave.. love you!!" This tweet shows sentiments of Sadness and Love.

- "Damn it.. This guy behind me just ruined the movie for me. I hate people talking in the cinema. Idiots!!" This tweet shows sentiments of Anger and Hate.

That being the case, it is quite difficult to identify all existing sentiments present in a few words, let alone detect which one is predominant. Several tweets that have been misclassified present multiple sentiments, and the classifier had difficulty determining the predominant one.

Closeness between different sentiments This has been discussed in the previous sub-section. Sentiments such as Happiness and Fun or Anger and Hate are largely similar, and tweets of one of each pair could easily be misclassified as being of the other. Along with context dependency, this is probably the major cause of misclassification.

Absence of Sentiment Indicators As stated above, tweets are short in length, and sometimes it is hard to extract useful information from them, or even find a common pattern that makes similar sentences show the same emotion. This has led, in the case of 7-class classification, to the misclassification of many tweets as Neutral (i.e., a low Precision of the class Neutral), as well as the misclassification of tweets with sentiments of the same polarity or even of different polarities. For example, the tweet *"Dead sure it was. invite me again anytime soon!"* was annotated as being of the class Happiness but classified as being of the class Sadness.

4.6 Conclusions

In this chapter, we studied the task of multi-class sentiment analysis. We evaluated the evolution of various KPIs as the number of sentiment classes increased. We analyzed the difficulties of, and the different challenges involved with, multi-class classification, and proposed some metrics to measure the distance between sentiments (i.e., how similar they are to one another). We concluded that, even though the task of multi-class analysis is important, it might be more interesting to perform a sentiment detection task through which all of the sentiments present within a text are extracted. This will be the focus of the next chapter, in which we describe our approach to perform this task that we refer to as sentiment quantification.

Chapter 5

Sentiment Quantification

5.1 Introduction

Sentiment analysis has been deeply studied in the literature: several approaches were proposed to perform this task on data collected from Twitter [37, 130–132] as well as other sources of online data [133, 134]. In a previous work [113], we have proposed an approach that performs this task on data collected from Twitter for several topics, where tweets were classified into positive or negative.

In chapter 3, we have dealt with a more challenging task, which we refer to as the "multi-class sentiment analysis", where tweets were classified into one of 7 different sentiment classes. However, as we discussed in chapter 4, this task presents several challenges. A major challenge we have deeply discussed is the fact that tweet simply might contain more than one sentiment. That being the case, in the current work, we aim to deal with this problem and solve it. We propose an approach that tries to actually detect all the sentiments existing in a given tweet and attribute different scores to these sentiments showing their weight, or how relevant they are in the tweet. We refer to this task as "quantification".

The contributions of this work are the following:

1. we introduce the task of sentiment quantification as described above and as we will describe in more detail more later in this work,
2. we propose an approach that relies on writing patterns along with other sets of features to perform a ternary sentiment classification of tweets (i.e., the classification into "positive", "negative" and "neutral"),
3. upon classification, the writing patterns are used again to attribute scores for each sentiment in every tweet. These scores are used to filter the sentiments we judge as being conveyed in the tweet (within the process we refer to as quantification),
4. we added the required quantification components to our previously introduced tool SENTA, to make it easy to run the approach.

The remainder of this chapter will be structured as follows: in section 5.2, we discuss the limitations of the multi-class sentiment analysis and present our motivations for this work. In section 5.3, we present some of the work related to the subject we discuss in this chapter. In section 5.4, we describe the modules and components that we have added to SENTA. In section 5.5 we describe in details our proposed approach for sentiment quantification and in section 5.6 we show the results of our experiments using the approach on a data set made out of tweets, we analyze the obtained results, and discuss the potentials and limits of the approach. Finally, section 5.7 concludes this work and proposes possible directions for future work.

5.2 Motivations

5.2.1 Multi-Class Classification: Potential and Limits

In chapter 3 and 4, we have explored the task of multi-class sentiment analysis in Twitter: for given tweet, instead of telling whether it is positive, negative or neutral, our aim was to actually identify the most dominant sentiment in it, that being "Happiness", "Love", "Sadness", etc.

Such a task is interesting given that it allows companies, for example, to distinguish between comments regarding their products that are dissatisfaction-driven and those which relate to physical damage or other. This can be seen in the following two tweets that show 2 different sentiments, despite being both negative:

- *"C'mon Valve!! get a solution for these bastard cheaters?? They are ruining the game and soon enough there won't be anyone playing CSGO!"*
- *"I bought it yesterday, and now it's discounted. Just why Valve why? :("*

Even though both tweets could interest the company in question, the first tweet could be judged as more important and a useful feedback of a frustrated and angry user, whereas the second is, somehow, showing a sentiment of sadness for the bad luck the user had.

The tweets in question are not unique, nor few in number. A negative tweet could have several interpretations, depending on the actual sentiment shown. The same can be said about positive tweets.

This highlights the importance of the multi-class classification, and shows why it is indeed needed. However, as we will see in more details in the next sections, tweets tend to show more than one sentiment in a single tweet. In the data set we have used in this work, we have asked human annotator to attribute one sentiment or more to every tweet, and the results show that more than 55% of the tweets actually contain more than one sentiment. That is not surprising though: in the previous chapter, we have studied the performances of the multi-class classification, and concluded that this is indeed a common thing: a sentimental tweet (i.e., a tweet that is not neutral) shows usually more than one sentiment. Nevertheless, some sentiments are highly correlated. As a matter of fact, tweets showing hate tend to show anger and frustration as well.

5.2.2 Why Quantification?

The presence of several sentiments within a tweet, as shown above, makes the task of multi-class classification a bit obsolete given that, out of all the sentiments presents, only one is identified. That being the case, the identification of all the existing sentiments is a very challenging task [115, 129]. Not only does it suggest that the different sentiments co-exist within the tweet, but also these might have different weights and manifestations. This leads to a more challenging task: is it possible to identify these sentiments and attribute different scores to them, each showing the weight of the corresponding sentiment?

In this work, we refer to the task of identification of these sentiments and the attribution of scores to them as "quantification".

5.2.3 SENTA: Requirement for an Update

SENTA has previously been introduced for the purpose of multi-class classification: it helps extract several sets of features and export them in several formats, allowing the user to use later on any program or tool to perform the classification. However, to makes it easy for a user to experiment with his data, it would be more interesting to allows him to run the classification using SENTA.

Nonetheless, as part of the quantification process, tweets are initially classified into 3 classes: positive, negative and neutral (ternary classification). Performing the classification somewhere else separately, and re-introducing the results is very inconvenient and impractical. Therefore arises the need for adding a classifier component to the tool so that the classification is performed internally.

Nonetheless, for the sake of quantification, other sets of features need to be introduced, notably what we will refer to as "Advanced Pattern Features". These features are very important for quantification, however, they can also be used for classification.

5.3 Related Work

Twitter, being one of the biggest web destinations and a very active microblogging service, has attracted an important part of the attention of researchers [24]. This is due partially to the several properties of Twitter that we introduced in Section 5.1. It is also due to the abundance of Twitter-collected data and the ease of manual annotation of tweets to experiment with.

Twitter analysis has covered several of its properties, and was not restricted to its content. Some of the works studied the relations between users and the identification of hidden communities [7, 135] the and the influence they might have on each other [136]. Tweets have also proven to be able to influence false memory [137] and spread fake information [138], making it interesting to understand how this platform (i.e., Twitter) orients the public opinion and influences it [114]. In this context, Achananuparp et al. [139] studied the user behavior with regards to the information propagation through microblogging websites, taking Twitter as an example. They used retweets as indicators of *originating* and *promoting* behaviors. They proposed several models to measure these two behaviors and demonstrated their applicability.

In a related context, Twitter has been studied as a potential teaching and learning tool [140] [141]. In [140], the authors conducted experiments to explore the teaching practice of Twitter as an active, informal learning tool, while in [141], the authors focused on the impact of Twitter, whether it is positive or negative, on informal learning, class dynamics, motivations and academic and psychological development of students.

However, sentiment analysis in social media in general, and Twitter in particular, has been among the hottest topics of research in the recent years: while sentiment anlysis has been a subject of research for decades and goes back to the 90s of the previous century (and even way back to the early years of the 20^{th} century) [24], the rise of internet, followed by the exponential growth of online content and the spread of social media usage made the topic of a high interest to companies and organizations [24]. This is because, nowadays, the end-user generated amount of data is very rich and covers several aspects of the users' lives as well as their opinions towards various topics and subjects. Performing sentiment analysis on such data is of great use to companies, for example, that want to know the opinion of average consumers [58, 59]. This is because data collected from online shops or dedicate movie review websites tend to be polarized, and people who are very satisfied or dissatisfied are more likely to share their experiences on these websites.

That being the case, we find in the literature several works that have dealt with the topic of sentiment analysis in Twitter. These works revolve mostly around the use of machine learning and

a pre-labeled data set to learn how to classify tweets. They started with simple approaches that re-applied the existing works that have been proposed previously for other types of texts, and soon after evolved into a more sophisticated ones that use features that are very specific to Twitter such as the use of slang words [82] or emoticons [81].

A particular task in sentiment analysis, referred to as aspect-based sentiment analysis, has also attracted the attention of researchers. Aspect-based sentiment analysis refers to the classification of sentiments for the different aspects present in a given piece of text. Zainuddin et al. [142] proposed a hybrid sentiment classification approach in which they use Twitter attributes as features to improve Twitter-aspect-based sentiment analysis. They ran their approach on several existing data sets to validate the efficiency of their proposed approach. Similarly, Bhoi and Joshi [143] proposed to use various classification approaches involving conventional machine learning and deep learning techniques to perform aspect-based sentiment analysis.

Multi-class sentiment analysis on Twitter has attracted part of the attention as well, but has not matured yet and the state-of-the-art works are good, but require deeper study. Multi-class classification refers to the identification of the exact sentiment(s) present in a given piece of text rather than just determining its overall polarity (whether it is positive, negative or neutral). To begin with, most of these works have dealt with this task in a different way from that we are dealing with. In fact, multi-class classification has conventionally referred to the attribution of one of several sentiment strengths to a text or a tweet. A typical classification task was to attribute one of the following sentiment classes to tweets: {“very negative”, “negative”, ‘neutral”, “positive” and “very positive”}, or simply attribute a score ranging from -1 to 1, showing at the same time the polarity and the strength of the sentiment [118, 119]. Nonetheless, with the wide adoption of Deep Learning as a cutting edge technology, this task has been dealt with as well in works such as that of Yu and Chang [144] and that of Araque et al. [145].

However, there have been several approaches which dealt with multi-class classification the way we do in this work: detect one (or more) sentiment(s) for a given text or tweet. For instance, Lin et al. [120, 121] proposed an approach in which they extracted features they qualified as “similarity features” and which they used to classify tweets into reader-emotion categories. A similar task has been tackled by Ye et al. [122] who proposed an approach that tries to identify the sentiments of readers of news articles. Nevertheless, Liang et al. [123] proposed a system that recommends emoticons (which eventually show emotions) for users while they are typing a text message. These emoticons are obviously generated by analyzing the sentiment in the text being typed. In a more recent work, Krawczyk et al. [146], has tackled the problem of multi-class sentiment analysis in imbalanced data collected from Twitter. They proposed an approach that relies on binarization scheme and pairwise dimensionality reduction to reduce the task into an easier one: they generate pairwise dichotomies, then for each pair of classes they reduced the feature dimensions and used several classifiers to perform the binary classification.

In a related, yet a bit far context, the term “quantification” has been used in the context of sentiment analysis the literature to refer to the estimation of the relative frequency of the different classes that the instances of a given data set are to be classified into. In other words, in most of the cases, the party who is performing sentiment analysis, cares more about the the percentage of data showing each sentiment (mainly in the case of binary or ternary classification). Therefore, it

might be interesting to find ways to identify these percentages instead of actually finding the class labels of the individual tweets. This idea has been developed and several approaches were made to solve this problem [147–150], even for a poor initial classification accuracy of the individual tweets [117]. It is important to understand that the current task we are dealing with in this work is completely different. It actually aims to identify the actual labels of the individual tweets. It is fair to assume it is closer to the context of the multi-class classification.

5.4 SENTA - Integrating the Quantification Components

5.4.1 Tools

To recall, SENTA was built using Java 8 and JavaFX, a platform used to make desktop applications.

We have also used Apache OpenNLP[1] Application Programming Interface (API) to perform the different Natural Language Processing (NLP) tasks such as the tokenization, Part-of-Speech (PoS) tagging, lemmatization, etc.

In the current work, we have referred to Weka[2] API [105], to make use of the different classifiers built-in. While Weka has a Graphical User Interface (GUI), we have built our own for the different classifiers that we have implemented so far.

5.4.2 Convention

As we previously stated in [129], the term "user" will be used to refer to the user of SENTA, whereas, if needed, the term Twitterer will be used to refer to a Twitter user. Nevertheless, in this section, the term "interface", will be used to refer to the graphical user interface of SENTA.

Furthermore, the interfaces and components of SENTA, which have been previously introduced in [129] will not be detailed here.

5.4.3 Graphical User Interfaces

Advanced Features Customization

The sets of features we have introduced previously were enough for tasks such as the multi-class classification. However, for quantification, our experiments have shown the limits of these in the detection of all the existing sentiments within a tweet. To begin with, only few sets actually take into account the different sentiments (i.e., unigram features, top words and pattern features). Other features, such as punctuation features, do not refer to the sentiments in the tweets, nor do they have any direct correlation with a given sentiment.

That being the case, we believe that adding more features is required to perform the task of quantification: we refer to these as "Advanced Features". Mainly 2 sets of features have been fully integrated so far, as shown in Fig. 5.1. These are:

- Advanced Unigram Features

[1] https://opennlp.apache.org
[2] https://www.cs.waikato.ac.nz/ml/weka/

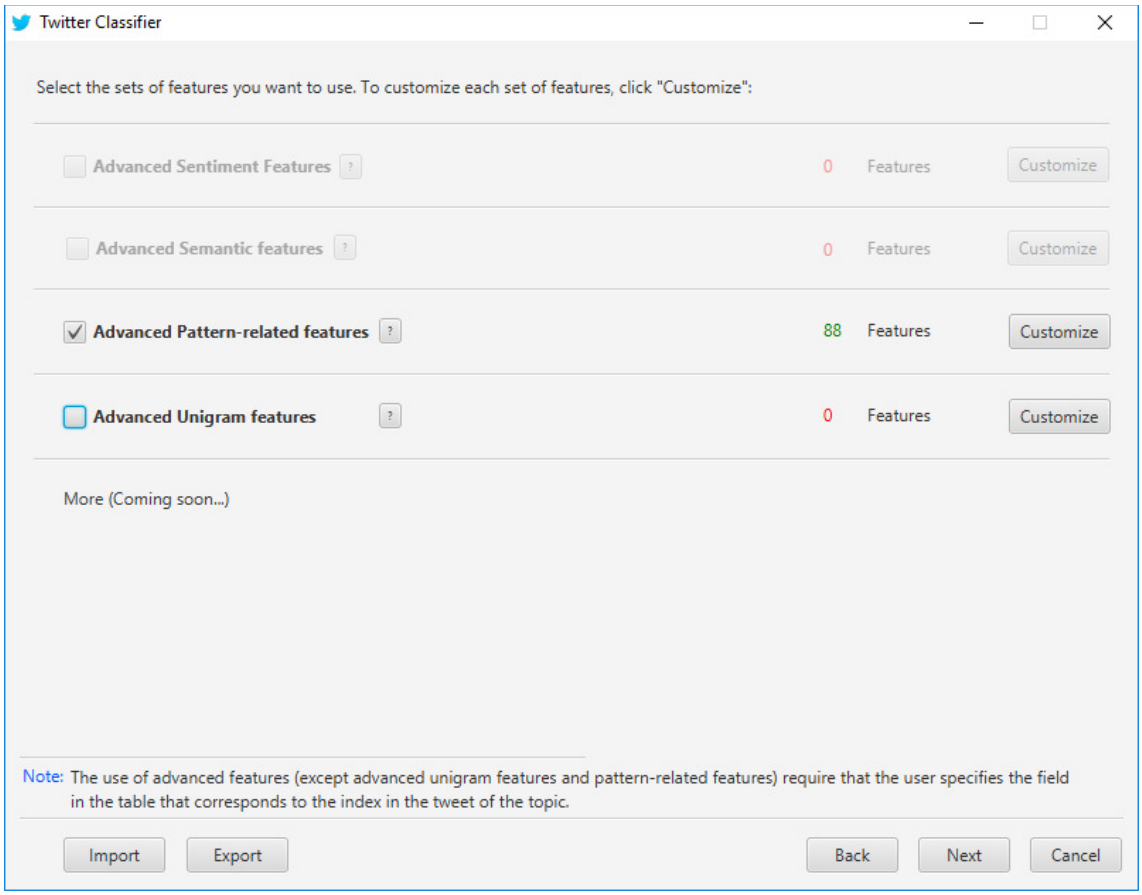

Figure 5.1: Advanced Features – Main Window

- Advanced Pattern Features

In the rest of this subsection, we describe these two sets of features, what they refer to and how they are extracted.

Advanced Pattern Features Advanced pattern features are similar to the old pattern features [129]. They are extracted from a given set (that could be the training set), and are used in two different ways (either each pattern is a unique feature, or several patterns can be scored and summed up together as we will explain later on). We rely on both Part-of-Speech tags and sentiment scores of words to extract the different advanced patterns. First of all, a word can be sentimental or not: if a word has the PoS of a verb, an adverb, a noun or an adjective, it is qualified as sentimental given that only these words (as well as some interjections) could convey sentiments; a word having any of the remaining PoS is qualified as non-sentimental. In addition, the same way we previously extracted words correlated with a given sentiments [129] (Unigram features) with the help of WordNet [125], we use the same approach to extract words correlated with each sentiment that we use in our data set. Obviously, these can only be verbs, adverbs, nouns or adjectives.

Unlike basic patterns, which are extracted for a given tweet regardless of its sentiment, advanced patterns are extracted differently for different sentiments. An advanced pattern is created as follows:

- For training tweets (tweets of known sentiments): given a tweet having sentiments $\{s_1, \cdots, s_N\}$,

Table 5.1: List of Simplified Part-of-Speech Tags

PoS-tag	**Expression**
“CC”	COORDCONJUNCTION
“CD”	CARDINAL
“DT”	DETERMINER
“EX”	EXISTTHERE
“FW”	FOREIGHWORD
“IN”	PREPOSITION
“LS”	LISTMARKER
“MD”	MODAL
“JJ”, “JJR”, “JJS”	ADJECTIVE
“NN”, “NNS”, “NNP”, “NNPS”,	NOUN
“PDT”	PREDETERMINER
“POS”	POSSESSIVEEND
“PRP”, “PRP$”	PRONOUNS
“RB”, “RBR”, “RBS”	ADVERB
“VB”, “VBD”, “VBG”, “VBN”, “VBP”	VERB
“RP”	PARTICLE
“TO”	TO
“UH”	INTERJECTION
“WDT”, “WP”, “WP$”, “WRB”	WHDETERMINER
“.”	.

for the sentiment s_i, the corresponding pattern will be extracted as follows: for each token, if it is a sentimental word, we verify whether it conveys the sentiment s_i. If it does, it is replaced in the pattern by its simplified PoS-Tag as shown in TABLE 5.1 along with the sentiment. Otherwise, if it is sentimental but does not convey s_i or if it is not sentimental, it is simply replaced by the corresponding simplified PoS-Tag as shown in TABLE 5.1.
- For test tweets (tweets whose sentiments are unknown): for all the sentiments that are being studied, we do the same: for each sentiment s_i, we extract a separate pattern using the same approach.

To concretize, given the following tweet:

“I liked it sooo much. Thanks a lot!”

if we suppose this is a tweet of known sentiments that has been annotated by human annotators into two sentiments “Happiness” and “Love”: this generates the following two full patterns:
- **Happiness:** [PRONOUN HAPPINESS_VERB PRONOUN INTERJECTION ADVERB . HAPPINESS_NOUN PARTICLE ADJECTIVE]
- **Love:** [PRONOUN LOVE_VERB PRONOUN INTERJECTION ADVERB . NOUN PARTICLE ADJECTIVE]
given that the word “like” shows both happiness and love, while “thank” shows only happiness.

If this tweet is of unknown sentiments, and whose sentiments need to be detected, in addition to the aforementioned patterns, we need to extract all the possible patterns for all the possible sentiments including:
- **Sadness:** [PRONOUN VERB PRONOUN INTERJECTION ADVERB . NOUN PARTICLE

ADJECTIVE]
- **Neutral:** [PRONOUN VERB PRONOUN INTERJECTION ADVERB . NOUN PARTICLE ADJECTIVE]
- etc.

Patterns are defined as ordered sequence of words with very specific length(s). They are extracted from the known data set. For a given tweet and a given sentiment, it is possible to extract several patterns. If a pattern happens to occur in a tweet of negative sentiments and a tweet of positive ones, it is discarded. Additionally, a pattern needs to occur several times in tweets of a given sentiment to make sure it really characterizes that sentiment. Patterns can be either unique features or summed up.

In the case where patterns are used as unique features they must have all the same length, and each pattern extracted from the known data set will be used to generate a single feature as follows: For a tweet T, and a reference pattern P extracted earlier from the known data set. We first extract the full patterns from the tweet and use the following resemblance function [77] to measure how much T resembles P:

$$res(p,T) = \begin{cases} 1, & \text{if the tweet vector contains the pattern as it is, in the same order,} \\ \alpha, & \text{if all the words of the pattern appear in the tweet in the correct order but with other words in between,} \\ \gamma \cdot n/N, & \text{if } n \text{ words out of the } N \text{ words of the pattern appear in the tweet in the correct order,} \\ 0, & \text{if no word of the pattern appears in the tweet.} \end{cases}$$

The result of resemblance is attributed to the corresponding feature, for the tweet T.

Obviously, this adds few parameters, that the user can adjust to maximize the results of detection of sentiments: he needs to choose the length of a pattern, the values for α and γ, as well as the minimum number of occurrences of the pattern.

In the case where patterns can have multiple lengths, they are taken such as their length satisfies the following:

$$L_{Min} \leq Len(pattern) \leq L_{Max} \tag{5.1}$$

where L_{Min} and L_{Max} refer to the minimal and the maximal allowed length for a pattern, while $Len(pattern)$ is the length of the pattern. In addition to the aforementioned parameters, one last parameter, which we refer to as knn, is to be optimized. Given all the patterns extracted for the sentiment class s_i and the length L_j, one feature is extracted. The value of this feature, which we refer to as F_{ij}, is calculated as follows [126]:

$$F_{ij} = \sum_{k=1}^{knn} res(p_k, T) \tag{5.2}$$

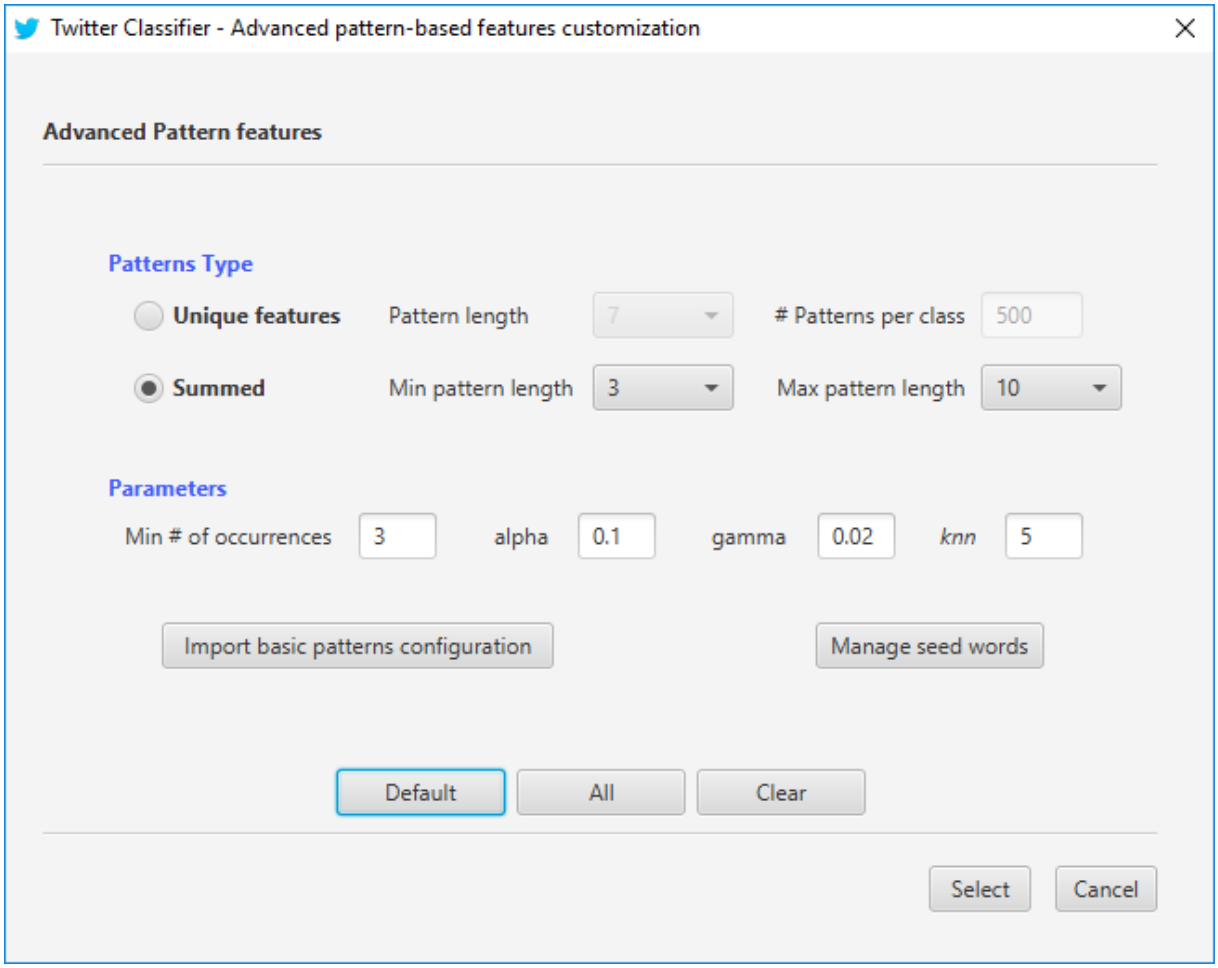

Figure 5.2: Advanced Pattern Features – Customization Window

where the different patterns p_k here are ones that have the highest resemblance to the tweet T. F_{ij} as defined measures the degree of resemblance of a tweet T to patterns of the sentiment class s_i and length j.

The different parameters related to advanced patterns can be optimized via the window shown in Fig. 5.2.

As stated previously, this set of features can be used for both classification and quantification. However, in the case of quantification, the user can only use patterns of multiple lengths (later on, we explain the reason).

Advanced Unigram Features Advanced unigram features are unigrams that the user specifies manually, and that will be checked against a given tweet. If a unigram exists in that tweet, the corresponding feature will be attributed the value *"True"*, otherwise, it will be attributed the value *"False"*.

Fig. 5.3 shows the window through which the user configures the advanced unigram features. The user needs to save the unigrams he wants to check in a file (one unigram per line). He can then select the file location by pressing *"Select"*. Optionally, the user can choose whether to compare the lemmas of the words of the tweets to those of the list he provides, or the actual words. For

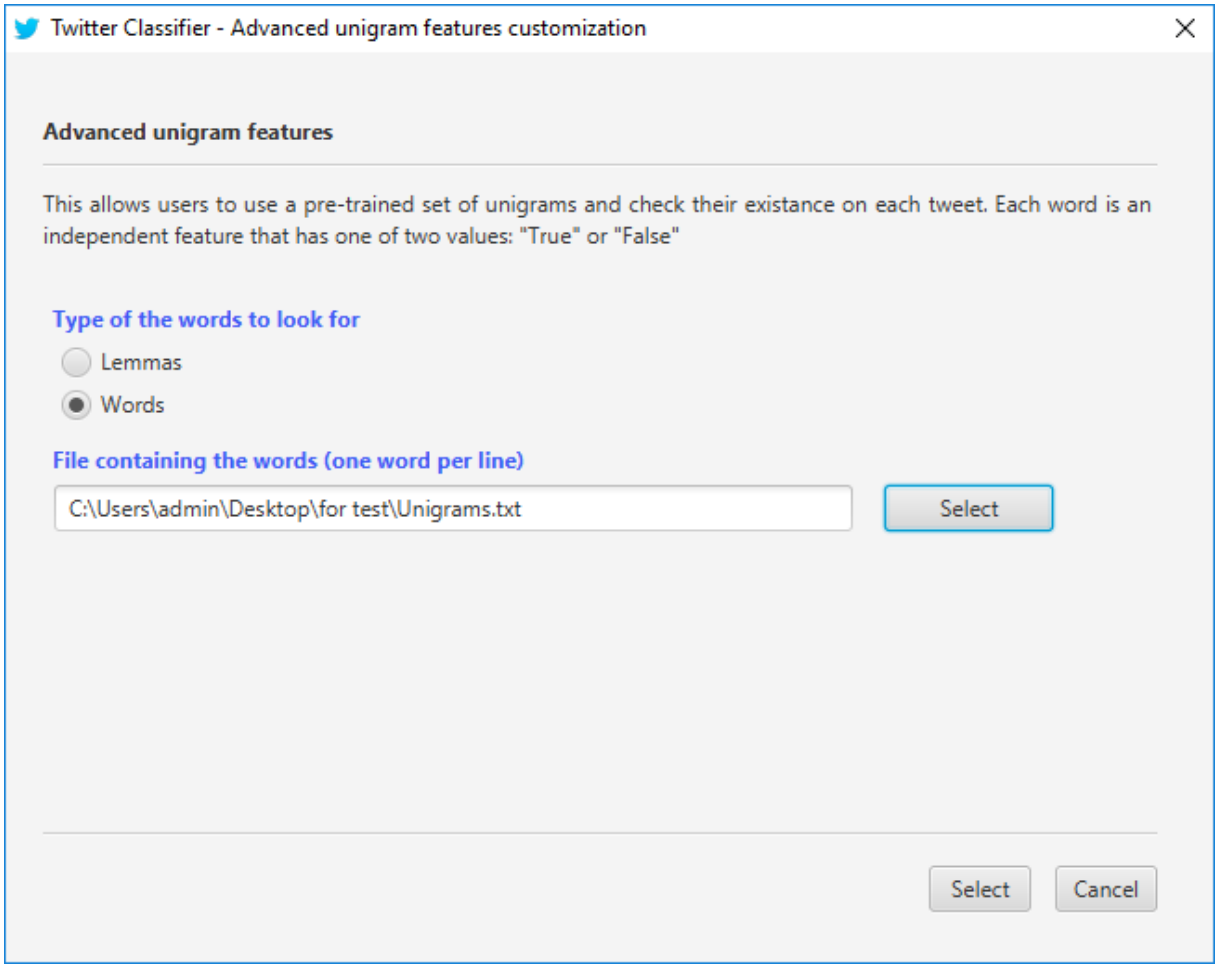

Figure 5.3: Advanced Unigram Features – Customization Window

example, if the the list of words contain the word ”love” and the tweet contains a word such as “loving”: if the users chooses to check for words, the corresponding feature for the word ”love” will be attributed “False”, whereas if he chooses to compare lemmas, the feature will be attributed the value “True”.

Advanced unigram features are supposed to be used in case the basic unigram features or the top words are not enough. It does not include useful information for the quantification though, so it will not be used in the current work.

Classification Window

In addition to the new sets of features we have described above, we have implemented several classifier interfaces, using Weka API. In the current version, we have added several classifiers. These include, but are not limited to:

- Naive Bayes classifier,
- Random Forest classifier [109],
- Iterative Dichotomiser 3 (J48) classifier [151]

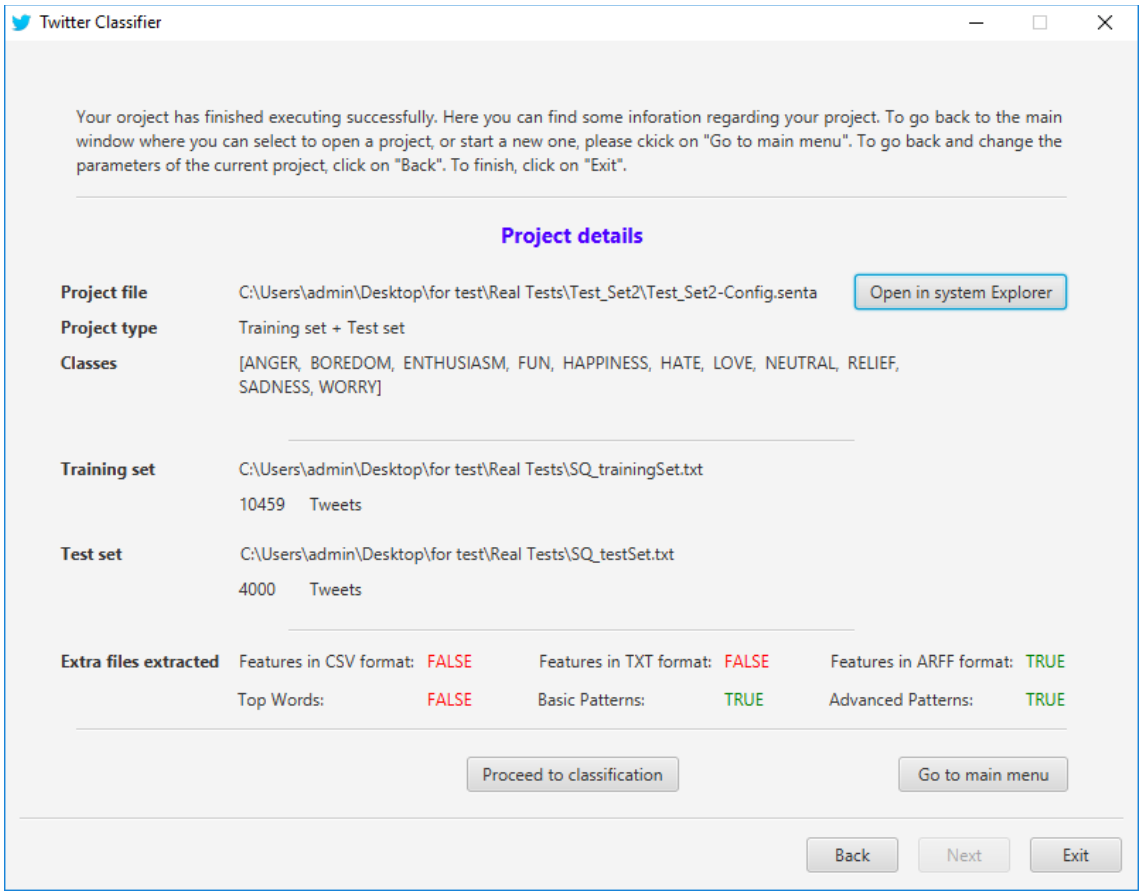

Figure 5.4: The Main Window Showing the Summery of the Project

Once SENTA has finished extracting the different features selected by the user, in the interface shown in Fig. 5.4 the user can press the button *"Proceed to classification"*. Since we are using Weka API, proceeding to classification requires the files with *"*.arff"* extension (i.e., weka file format) for both the training and the test set to be generated. So in case the user has not selected to generate these files, they will be automatically generated.

Upon proceeding, the interface shown in Fir. 5.5, will be displayed. The user chooses the classifier he wants to use, sets the different parameters of the classifier and selects the operation he wants to perform (e.g. training set cross validation, experimenting with the test set, etc.). In Fig. 5.6 we show an example of parameters optimization window (that of Random Forest classifier). The default parameters offered by Weka are used as default parameters here.

The classification results will be saved every time and the user can go back to check them by selecting the corresponding iteration from the table, and clicking *"Display"*. However, only the results are saved, and not the classification model. Additionally, SENTA stores only the results of classification of the individual tweets only for the last classification operation (Later on, for quantification, these results are the ones that are used).

Quantification Window

Once the classification is done (on the test set or the validation set), the user can proceed to the quantification. Basically, if the user has chosen to perform a quantification task, regardless

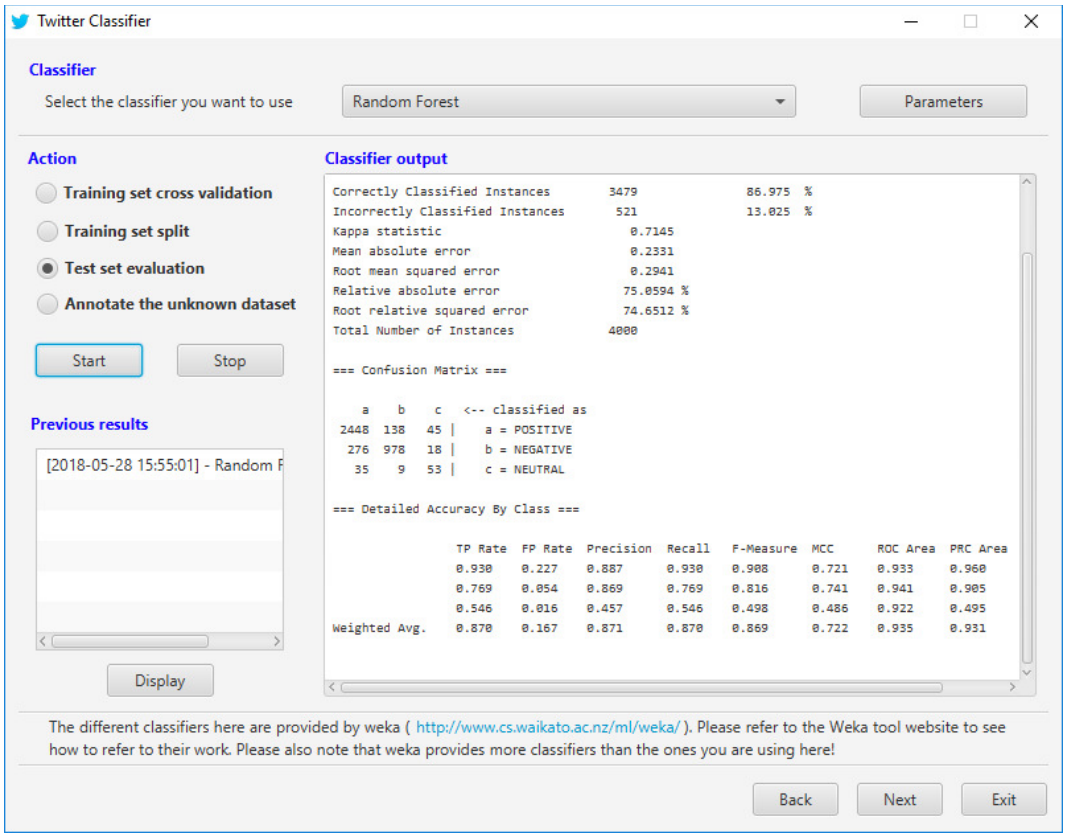

Figure 5.5: Classifiers Main Window

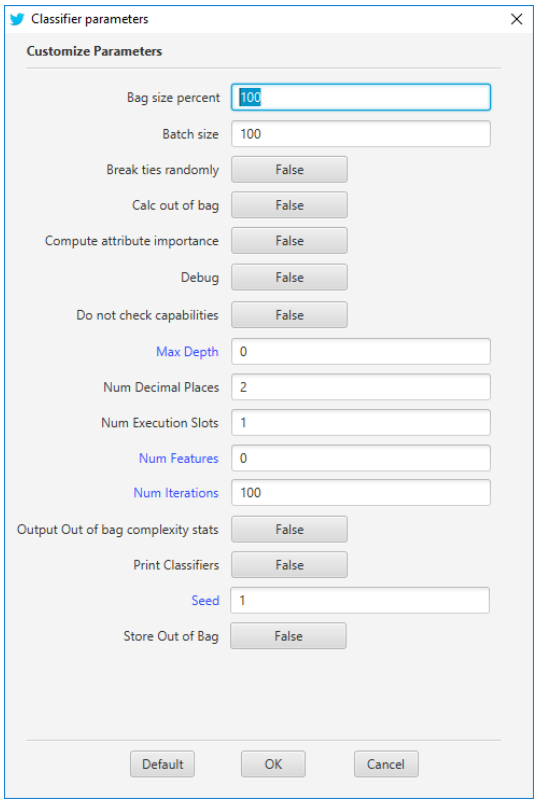

Figure 5.6: Classifier Parameters Optimization Window

Table 5.2: Pattern Features

		Pattern length			
		L_1	L_2	$\cdots$	L_M
Sentiment Class	1	F_{11}	F_{12}	$\cdots$	F_{1M}
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	S_N	F_{N1}	F_{N2}	$\cdots$	F_{NM}

of the number of sentiment classes that he initially selected and that the tweets might contain, the classification task will classify tweets into one of 3 classes: positive, negative or neutral. The sentiment classes the user has specified will be used in quantification. This assumes that a tweet contains exclusively positive, negative or neutral sentiments (i.e., a tweet cannot have two sentiments of different polarities at once). Despite the fact that this assumption is not always satisfied (e.g., in our data set less than 3% of the tweet did actually have sentiments of different polarities), it is needed in order for the ternary classification to make sense. Technically, SENTA is implemented in a way that, in case a tweet contains sentiments of different polarities, the polarity of the first sentiment present in the list of sentiments of that tweet is taken into account.

The quantification task will use the results of the classification, and the values of the following sets of features:

- Unigram features,
- Basic pattern features,
- Advanced Unigram features.

To recall, unigram features work as follows: we dispose of several lists of words, each we judged highly correlated with a certain sentiment. We count, in every tweet how many words from each list appear in it.

For a given tweet, suppose that the corresponding features have the following format $[U_1, U_2, \cdots, U_N]$ where U_i is the i^{th} feature corresponding to the i^{th} sentiment. These values are then normalized by dividing all of them by the maximum value (obviously if they are all equal to 0, they are kept as they are). We refer to the resulting scores as $S_i^U(T)$, where $i \in \{1, \cdots, N\}$.

We do the same for the different patterns (basic and advanced patterns work the same way): Given that the user has set the parameters for L_{Min} and L_{Max} for the minimal and maximal pattern lengths respectively, and the parameters α and γ, the features will have the format shown in TABLE 5.2 as detailed in [129].

Given that these features are extracted, we need to derive two scores (one using basic pattern features and one using advanced pattern features) for each sentiment for a given tweet T. The scores will have the following format:

$$S_i^P(T) = \sum_{j=1}^{M} \Big(\beta_j \cdot F_{i,j}\Big) = \sum_{j=1}^{M} \Big(\beta_j \cdot \sum_{k=1}^{knn} res(p_k, T)\Big) \tag{5.3}$$

where $S_i^P(T)$ is the score generated using patterns, of the sentiment i for the tweet T, M is the number of pattern lengths (to recall, the lengths are $\{L_1, \cdots, L_M\}$) and β_j is a weight given to patterns of length L_j (regardless of their class). Currently, we set the values for β_j as follows:

$$\beta_j = \frac{L_j - 1}{L_j + 1}, \tag{5.4}$$

where L_j is the length of the pattern. Again these scores are normalized by dividing them by the highest score for T. The resemblance function $res(p_k, t)$ is the one that we have defined in Section 4.3.1.

We refer to the Basic Pattern Score and Advanced Pattern scores of the i^{th} sentiment in the tweet T as $S_i^{BP}(T)$ and $S_i^{AP}(T)$ respectively.

Finally, the user gets to choose a coefficient that highlights the importance of each of the given scores (i.e., $S_i^U(T)$, $S_i^{BP}(T)$ and $S_i^{AP}(T)$), to detect the sentiments existing in the tweet. In other words, given the following total score:

$$S_i(T) = \tau \cdot S_i^U(T) + \mu \cdot S_i^{BP}(T) + \nu \cdot S_i^{AP}(T) \tag{5.5}$$

the user can adjust the values of τ, μ and ν to adjust the importance of the 3 sub-scores. In addition, $\tau + \mu + \nu = 1$.

The different scores $S_i(T)$ are normalized as well. Sentiments that have a score higher than a certain threshold are ones judged as detected. The threshold is also a parameter to optimize.

In Fig. 5.7, we show the interface through which the user can set these parameters. The user can also choose to let SENTA automatically optimize these parameters for him. The function to optimize is the $F1\text{-}Score$, which we will introduce and explain later in this work.

5.4.4 Future Extension

In the current version of SENTA, we have introduced few new sets of features. However, 2 of them are still under experimenting and require some tuning to be useable. They will be used exclusively for classification purposes and will not contribute to the quantification. The next version will include these sets of features.

In addition, we have implemented few classifiers. These are ones that we have found best fitting in the context of multi-class sentiment analysis (mainly Random Forest). However, a user might need to compare several machine learning algorithms, or perform a task different from the one SENTA was designed for (e.g., sarcasm detection or hate speech detection) which will require a different classifier such as Support Vector Machine (SVM) or others. These classifiers will be added as well in the next version of SENTA.

Finally, it might be interesting for a user to save the classification model built using his training set, or import one that he has already built externally using Weka. Such features need to be added as well.

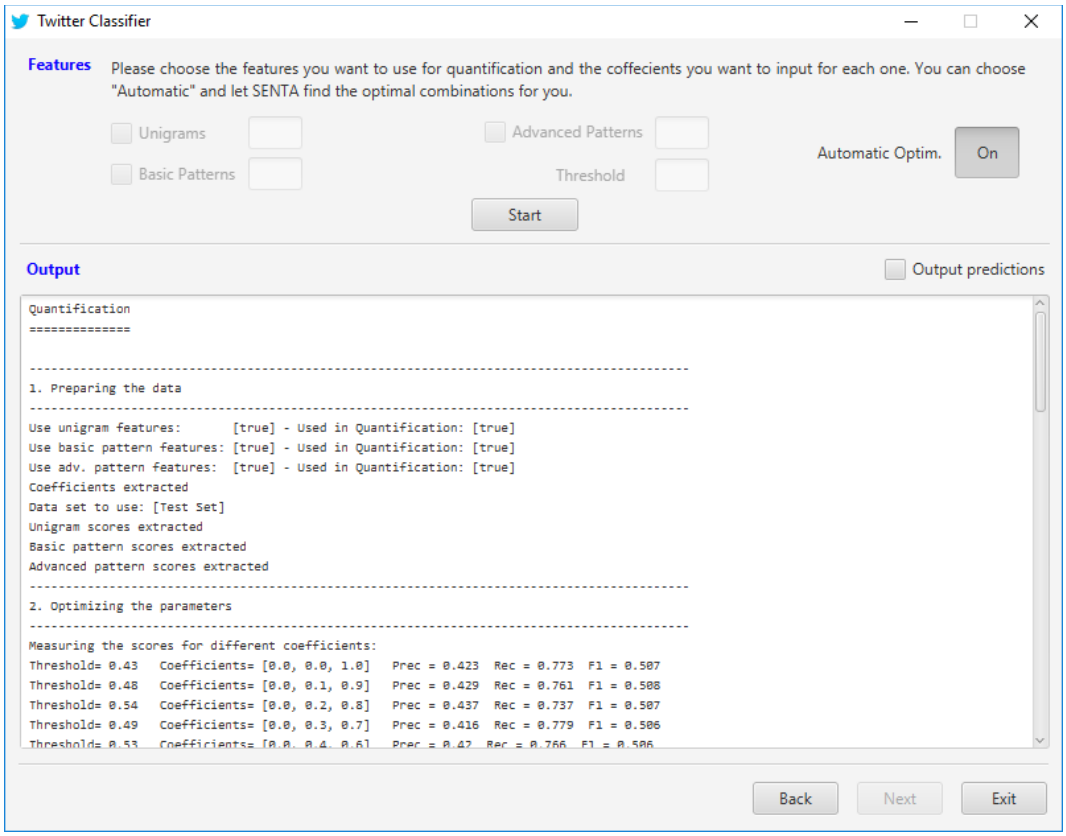

Figure 5.7: Quantifier Main Window

5.5 Sentiment Quantification - Proposed Approach

5.5.1 Problem Statement

Although the multi-class classification of tweets has its advantages and makes sense in the context of detecting the actual sentiment of a given tweet, it has its limitations as we explained in Section 5.2. Among these limitations, we highlighted the particular issue of not being able to identify all the existing sentiments within the tweet if it contains more than one. In other words, if a tweet presents more than one sentiment, the classification task will attribute a single sentiment label.

This makes it more reasonable to try to detect all the existing sentiments. As a matter of fact, in the training set we are using in this work for example, over 59% of the tweets contain more than 2 sentiments (the details of the structure of the data sets used will be given in the next subsection). That being the case, the task we tackle here is as follows: given a tweet, we first try to detect its sentiment polarity (i.e., whether it is positive, negative or neutral). We then try to identify all the existing sentiments by attributing a score for each sentiment. The sentiments are then ranked according to the attributed scores, and the ones that have the highest scores are judged as conveyed in the tweet. In other words, a tweet will be classified into one of the 3 classes described, and then into a further granularity level, but allowing it to have multiple classes.

5.5.2 Data

For the sake of this work, we have prepared a data set made of tweets collected using Twitter API. These tweets were manually annotated by several annotators using the services of CrowdFlower[3]. We asked the annotators to attribute 1 or more sentiments (out of 11) to each tweet, and encouraged them to choose more than one. However we have not made this requirement mandatory.

Two annotators annotated each tweet. The outputs of their judgement are merged. Tweets with inconsistent judgement are discarded from our data set. By the expression "tweets with inconsistent judgement", we mean ones that the annotators did not agree on a single sentiment shown in them. We have also discarded tweets with sentiments of opposite polarities (i.e., tweets which have at least one positive sentiment and at least on negative sentiment).

As stated above, when running the task, we have asked the annotators to attribute one or more sentiment(s) for each tweet, from the following sentiment classes:

- **Positive sentiments:** Enthusiasm, Fun, Happiness, Love and Relief,
- **Negative sentiments:** Anger, Boredom, Hate, Sadness and Worry,
- **Neutral sentiment:** Neutral.

This data set has then been divided into 5 data sets, as follows:

- **A pattern extraction set:** as we described in [129] and in Section 5.4, we need to collect what we qualified as patterns that we will use later to attribute pattern scores (which we refer to as "Pattern Features" and "Advanced Pattern Features" and which we use later to perform both the classification and the quantification). In [129], we extracted these patterns from the training set itself. However, we believe that this would make the classification favors these features over the others, because they fit in very well for the training set. Therefore, in the current work, we use an independent data set (thus the name "Pattern Extraction Set") to avoid such problem. This set is used only for the extraction of patterns of each sentiment class, and will be discarded afterwards.
- **A training set:** This set is used to train our model for classification.
- **A test set:** This set is used to run our experiments. The classification and quantification results obtained in this work are ones that were run on this set.
- **A validation set:** Throughout our experiments, we have optimized several parameters that we defined for SENTA. To make sure that these parameters are good, we validate them using a separate data set. This set will be referred to, in the rest of this work, as the "Validation Set".

As stated above, it is important to notice that several tweets were judged by the annotators as containing sentiments of opposite polarities (i.e., containing at least a positive sentiment and a negative one). These tweets were discarded as well, since they do not fit in the problem we stated in the previous subsection.

[3] https://www.crowdflower.com/

Table 5.3: Number of Tweets Having each Sentiment in the Different Data Sets

	Pattern set	Training set	Test set	Validation set
Fun	2854	2182	892	925
Enthusiasm	4010	3099	1327	1320
Happiness	4499	3631	1458	1471
Love	2557	2019	780	775
Relief	679	545	216	247
Neutral	4136	1591	395	401
Anger	1820	1080	450	417
Boredom	962	553	189	201
Hate	967	645	277	258
Sadness	3425	2040	827	780
Worry	2578	1522	590	572

Table 5.4: Distribution of Sentiments in the Different Data Sets

	Pattern	Training	Test	Validation
1 Sentiment	7937	4287	1478	1463
2 Sentiments	6568	4726	1949	1985
3 Sentiments	866	620	267	274
4 Sentiments	1204	827	306	278
Total # tweets	16575	10460	4000	4000

The structure of the data sets is given in TABLES 5.3 and 5.4: In the first table we describe the number of tweets having each sentiment in each of the data set. And in the second, we describe the number of sentiments per tweet in each of the data sets.

Fig. 5.8, shows a diagram of the proposed approach procedure: Initially, from the data set we have qualified as “Pattern Set”, basic and advanced patterns are extracted following the rules we have described previously. These two sets of features are then used along with the other sets of features as described in [129] to train a classification model on the training set. The model is optimized for the test set. After classification, the quantification process is run on the test set. The values of the parameters that have given the best results of classification and quantification on the test set were then verified on a totally independent set, which we refer to as the validation set, to verify whether they are overfitting the test set or they do present good (probably sub-optimal) performances on other sets.

5.5.3 Features Extraction

From the tweets, we extract different sets of features, that we use to perform the classification and later on the quantification. SENTA offers the option to extract the features we need for this work.

Basic Features

Here, we refer to our previous work [129] and extract the same features, with the same parameters. To recall, the features extracted are the following:

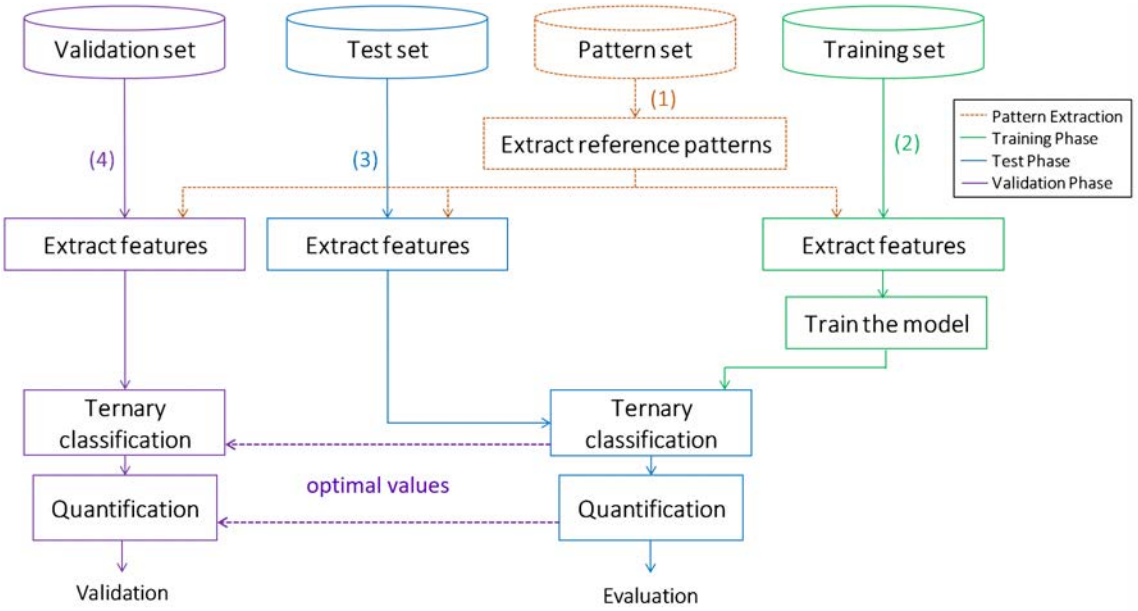

Figure 5.8: Flowchart of the Proposed Approach

1. **Sentiment Features:** these are features that help detect the sentiment polarity of the different components of the tweet (e.g., words, emoticons, hashtags, etc.).

2. **Punctuation Features:** these are features related to the use of punctuation in the tweet.

3. **Syntactic and Stylistic Features:** these are features related to the use of words and expressions in a tweet.

4. **Semantic Features:** these are features related to the meaning of words, the relations between them and the logic behind them.

5. **Unigram Features:** these are features extracted with references to word lists, where each list presents the words that are highly correlated with a given sentiment.

6. **Basic Pattern Features:** these are features that try to identify the common patterns or expressions used in different contexts to show certain emotions. They are extracted with reference to the data set with manually labeled data.

Advanced Features

In the current work, we will restraint to the use of one set of advanced features, which we qualified as “Advanced Pattern Features”. Advanced pattern features resemble to basic pattern features, but are more specific to the different sentiments ad we explained previously in Section 5.4.

We use all the features together to perform the classification and the quantification. However, unlike [129], we have not referred to the training set to extract the patterns that we use for the classification, but rather to a separate data set that we qualified as “pattern extraction set”.

5.6 Experimental Results

In this section, we present the results obtained for ternary classification and quantification, on both the test set and the validation set. As we explained earlier, the classification parameters and model as well as the quantification parameters will be optimized for the test set. The validation set is used to check the validity of these parameters and model on a new data set that has not been involved in the optimization.

5.6.1 Key Performance Indicators

After the extraction of features, we run different tests using the *"Random Forest"* [109] classifier. We use 4 Key Performance Indicators (KPIs) to evaluate the classification and quantification results: True Positives Rate, Precision, Recall and F1-score:

- **True Positives Rate (TPR or Recall)** measures the rate of tweets correctly classified as part of a given class over the total number of tweets of that class:

$$TPR = Rec = \frac{TP}{TP + FN} \tag{5.6}$$

- **False Positive Rate (FPR)** measures the rate of tweets falsely classified as part of a given class over the total number of tweets that are not part of that class:

$$FPR = \frac{FP}{FP + TN} \tag{5.7}$$

- **Precision (Prec)** measures the rate of tweets correctly classified as being part of a class, over the total number of tweets classified as belonging to that class:

$$Prec = \frac{TP}{TP + FP} \tag{5.8}$$

- F_1 **score** is a combination of both precision and recall defined as follows:

$$F_1 \textit{ score} = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec} = \frac{2TP}{2TP + FP + FN}. \tag{5.9}$$

In the context of classification the terms TP, FP, TN and FN are measured for all the tweets at once and are defined, for a given class C, as follows:

- **TP (True Positive)** refers to the fraction of tweets belonging to C and identified as belonging to C,

- **FP (False Positive)** refers to the fraction of tweets not belonging to C and identified as belonging to C,

- **TN (True Negative)** refers to the fraction of tweets not belonging to C and identified as not belonging to C,

Table 5.5: Sentiments confusion Matrix for a Given Tweet

	True sentiments	
Predicted sentiments	TP	FP
	FN	TN

Table 5.6: Ternary Classification Performances on the Test Set

Class	TP Rate	FP Rate	Precision	Recall	F1-Score
Positive	0.902	0.349	0.778	0.902	0.836
Negative	0.683	0.086	0.794	0.683	0.734
Neutral	0.319	0.023	0.602	0.319	0.417
Overall	0.774	0.232	0.766	0.774	0.762

- **FN (False Negative)** refers to the fraction of tweets belonging to C and identified as not belonging to C.

In the context of quantification, we measure the values of these terms is different. Given the quantification results of the single tweet shown in TABLE 5.5, where:

- **TP (True Positive)** refers to the sentiments that are identified correctly by our code as being shown in the tweet,
- **FP (False Positive)** refers to the sentiment that were judged as being shown in the tweet, when in reality, according to the annotators, they are not,
- **FN (False Negative)** refers to the sentiments that are present, according to the annotators, in the tweet, but our code could not identify them,
- **TN (True negative)** refers to the sentiments that are not present in the tweet, and were not judged as present in the tweet.

In this sense, the overall KPIs measured for the entire test set (and validation set) are the average of the values of these KPIs measured at tweet level.

5.6.2 Ternary Classification Results

Ternary Classification on the Test Set

We first run the classification on the test set. The classification results returned by the classifier Random Forest are the best, compared with other classifiers. This goes along with our previous observations in [126, 127, 129]. The results of classification are given in TABLE 5.6.

The results show that, in the current data set, the positive tweets are easier to detect than the negative or the neutral ones. The classification TPR of positive tweets reaches 90.2%, whereas that of negative tweets is 68.3% and that of neutral ones is only 31.9%. As we have explained in [129], in such data sets, tweets tend to be polarized (classified either as positive or negative, but rarely neutral) for several reasons including the nature of features themselves which are engineered to detect the presence of sentimental components, as well as the unbalanced amount of training data in favor of the non-neutral tweets.

Table 5.7: Ternary Classification Performances on the Validation Set

Class	TP Rate	FP Rate	Precision	Recall	F1-Score
Positive	0.897	0.368	0.764	0.897	0.825
Negative	0.680	0.090	0.786	0.680	0.729
Neutral	0.285	0.020	0.617	0.285	0.390
Overall	0.763	0.241	0.756	0.763	0.749

The overall accuracy is equal to 77.4%, with a precision level equal to 76.6%, a recall equal to 77.4% and an F1-score equal to 76.2%. These results are promising, even though they are lower than those obtained in [129].

Ternary Classification on the Validation Set

Given the same classifier parameters we have used in the previous classification task, we run the classification on the validation set. The results of classification are given in TABLE 5.7.

As we can observe, the classification results do not differ much from those on the test set. While we notice a slight decrease in the overall accuracy by about 1.1%, the results are pretty much close. The overall accuracy on the validation set is equal to 76.3% with a precision equal to 75.6%, a recall equal to 76.3% and an F1-score equal to 74.9%.

Moreover, the classification performances per class are also very similar: the classification TPR and recall of the positive tweets is the highest marking values equal to 89.7% both. Neutral tweets are also the hardest to identify with a TPR equal to 28.5%, but with a high precision level proving again that the reason of misclassification of these tweets is actually the tendency to polarize tweets. However, once identified as neutral, a tweet is most likely to be neutral (precision equal to 61.7%).

However, the important results we can conclude is that the classification performances are independent from the test set, and that we can proceed to the quantification part with no overfitting issue for the classification part.

5.6.3 Quantification Results

Given a tweet that was annotated by human annotators into m sentiments. The tweet is attributed n sentiments using our method.

While the different KPIs are being measured, we only focus on optimizing the F_1 score given that it is the most significant KPI. In other words, for a high precision, a high threshold can be used, which will result in a low recall given that the process of minimizing the False Positives tends to favor the detection of a single sentiment. The same goes the other way around: for a high recall, a very low threshold can be used, which will result in a low precision, given that the process of minimizing the False Negatives tends to favor the detection of almost all sentiments, so that no True Positive escapes.

Running the quantification on the test set gave us the results shown in TABLE 5.8. The results shown are the top ones for different values of the tuple [τ, μ, ν]. For convenience and ease of display, we discarded the combinations that gave lower values.

Table 5.8: Quantification Results on the Test Set

τ	μ	ν	**Precision**	**Recall**	**F1-score**
0	0.2	0.8	0.403	0.653	0.459
0	0	1	0.388	0.682	0.459
0	0.3	0.7	0.387	0.683	0.459
0	0.1	0.9	0.388	0.681	0.458
0	0.4	0.6	0.388	0.674	0.457
0.1	0.3	0.6	0.387	0.68	0.457
0.1	0.2	0.7	0.401	0.651	0.456
0.1	0.1	0.8	0.399	0.658	0.456
0.1	0	0.9	0.395	0.665	0.456
0.1	0.4	0.5	0.373	0.703	0.455
0.2	0	0.8	0.39	0.671	0.454
0.2	0.3	0.5	0.382	0.682	0.454
0	0.5	0.5	0.367	0.712	0.454
0.2	0.2	0.6	0.404	0.642	0.453
0.2	0.1	0.7	0.399	0.652	0.453
0.3	0	0.7	0.379	0.688	0.452
0.3	0.3	0.4	0.365	0.714	0.452
0.1	0.5	0.4	0.359	0.726	0.452
0.2	0.4	0.4	0.356	0.733	0.452
0.3	0.2	0.5	0.359	0.729	0.451

The values obtained reach a maximal F_1 score equal to 45.9% when [τ, μ, ν] = [0, 0.2, 0.8]. More interestingly, all these top values are obtained for a value of τ equal to 0, or very small. This translates into the fact that unigram scores do not contribute much to the detection of sentiments. In fact, this feature returns a score equal to 0 for many tweet, meaning that they actually do not contain words referring to any sentiments at all.

In TABLE 5.9, we show the results of quantification using the same tuples [τ, μ, ν] (in the same order). The best result obtained in the test set corresponds to a sub-optimal, yet very good, results on the validation set. The best F1-score is obtained when [τ, μ, ν] = [0, 0.3, 0.7] (i.e., F1-score equal to 47.7%). However, the tuple [0, 0.2, 0.8] presents very good results reaching 44.6%.

As stated previously, if we opt for the optimization of the recall, we observe that for all the tweets that were correctly classified, the quantification results in attributing a threshold for sentiment equal to 0 leading to attributing all the sentiment of the polarity to the tweets. In other words, given a positive tweet for example, optimizing the Recall results in attributing all the positive sentiments to the tweet, to make sure the correct sentiments are detected. In a similar way, the optimization of the precision results in very strict selection, leading to the attribution of a single sentiment per tweet. Therefore, we opted for the optimization of F1-score, which makes a lot of sense. The corresponding values of Recall and Precision are not the optimal, but are more meaningful.

Since the contribution of the Unigram score is minimal, we collected the different combination that have τ set to 0. The F1-score of these combinations on the test set and the validation set are given in Fig. 5.9.

Table 5.9: Quantification Results on the Validation Set

τ	μ	ν	**Precision**	**Recall**	**F1-score**
0	0.2	0.8	0.384	0.652	0.446
0	0	1	0.373	0.6765	0.445
0	0.3	0.7	0.369	0.683	0.447
0	0.1	0.9	0.373	0.674	0.446
0	0.4	0.6	0.365	0.683	0.445
0.1	0.3	0.6	0.374	0.666	0.444
0.1	0.2	0.7	0.371	0.677	0.445
0.1	0.1	0.8	0.369	0.680	0.444
0.1	0	0.9	0.366	0.685	0.443
0.1	0.4	0.5	0.365	0.680	0.443
0.2	0	0.8	0.359	0.699	0.442
0.2	0.3	0.5	0.360	0.691	0.442
0	0.5	0.5	0.353	0.707	0.442
0.2	0.2	0.6	0.363	0.689	0.442
0.2	0.1	0.7	0.371	0.673	0.442
0.3	0	0.7	0.356	0.703	0.441
0.3	0.3	0.4	0.352	0.702	0.439
0.1	0.5	0.4	0.349	0.711	0.439
0.2	0.4	0.4	0.366	0.668	0.440
0.3	0.2	0.5	0.355	0.701	0.440

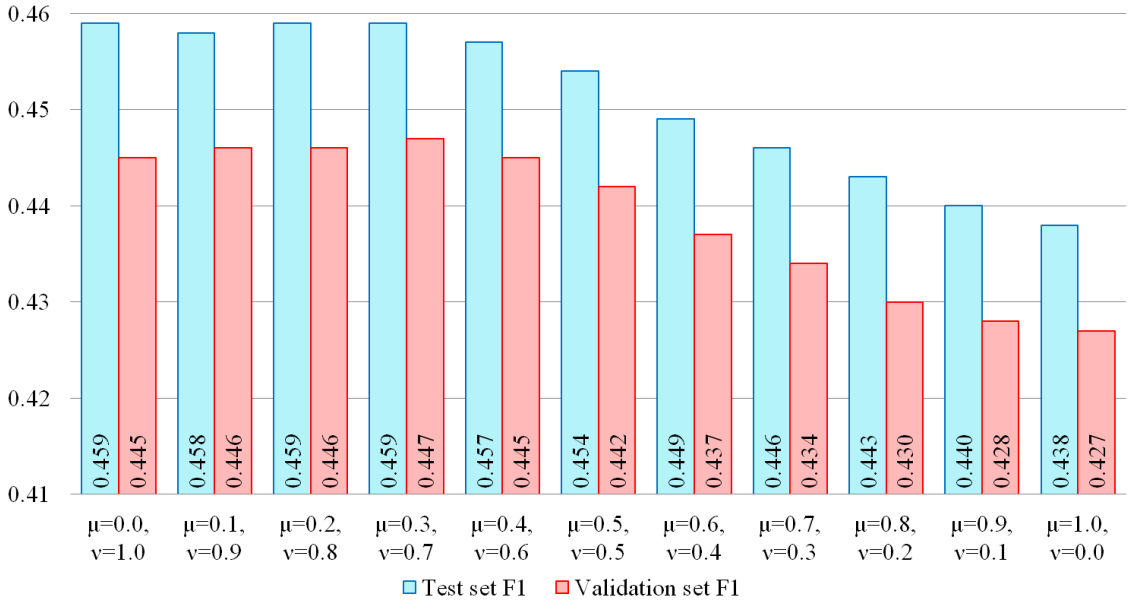

Figure 5.9: F1-Score for Different Values of μ and ν on the Test Set and the Validation Set

Table 5.10: Comparison Between the Proposed Approach and the Baseline One

Approach	Precision	Recall	F1-Score
Proposed Approach	0.403	0.653	0.459
Baseline	0.270	0.563	0.365

The figure shows a very similar behavior on both the test set and the validation set. It also highlights the fact that the advanced patterns, which are part of the contribution of this chapter, are more valuable in terms of detection of sentiments and quantification in general. As a matter of fact, even if we discard the basic pattern scores (i.e., set μ to 0), the results obtained are very close to the best ones obtained for $[\mu, \nu] = [0.2, 0.8]$.

5.6.4 Comparison with a Baseline Approach

To the best of our knowledge, the task we have defined in this work is new, and no previous work we encountered dealt with it. Therefore, to evaluate our approach, we define a baseline and compare the performances of our approach to its performances.

The baseline approach is defined as follows: Given a tweet T, we run the binary classification on each sentiment to guess whether or not that sentiment is present on the tweet or not. We use all the sets of features, except advanced pattern features (which are part of the contribution of this work).

This baseline has given very poor results, so it has been adjusted so that it makes use of the output of the ternary classification. Instead of running the classification on all the sentiments, we use the output of the ternary classification to restrict the number of sentiments to be verified. For example, if a tweet is judged as positive, the binary classification of only the five positive sentiments is run.

A comparison between the performances of the proposed approach and the baseline one on the test set is given in TABLE 5.10.

5.6.5 Discussion

In this work, we have introduced a task different from the conventional sentiment analysis one, and even from the multi-class classification task introduced in [129]. Throughout this work we have tried to identify all the existing sentiments within tweets, by attributing different scores to each sentiment in a tweet, and selecting ones with the highest scores. We referred to this task as quantification.

The results of quantification observed were promising. However, we believe that the not-exceptionally good results can be enhanced in more than one way. Several factors have led to a low results of classification and/or quantification of many tweets.

To begin with, the quantification task is a challenging task, that is highly subject to the annotators' opinion. This is actually a property that is valid for sentiment analysis in general, even for simple tasks such as the binary classification, where texts are to be classified into positive or negative. However, the finer the granularity level of classification is, the harder the task gets, and the more discrepancy between annotators there is. As a matter of fact, we have studied the data

set we used in [129] and we found a ratio of agreement between annotators on a sample of 300 tweets to be 67.3% on the 7-class classification, an agreement that jumps to 82.7% for ternary classification. Therefore, we expect to have even more disagreement (i.e., lower agreement level) on a data set that needs to be attributed one(s) from 11 sentiment classes.

Nevertheless, it is important to mention that the values of the two parameters α and γ set for the basic and advanced patterns were optimized for the classification. This means that they might not be the optimal values for the quantification. In fact, setting these two values to 0.1 and 0.02 respectively decrease greatly the value of sparse and incomplete resemblance of patterns to the tweet. This leads us to believe that different values for these features might mean different results for the quantification. This dilemma is set in favor of the classification, given that a misclassified tweet has an F1-score equal to 0 anyway.

On a related context, we have noticed that the accuracy of classification of the neutral tweets on both the test and validation sets was very low. It was way lower than that observed in [129]. Again, that is due to the low amount of training data for these tweets, among others. A neutral tweet that is misclassified has an F1-score equal to 0. This leads to a total decrease in the overall F1-score.

Over and above that, we believe that more training data instances, and more importantly a training set that is balanced among all the sentiment classes could improve noticeably the results. As we can see in TABLE 5.3 that we described in Section 5.5.2, the tweets are very unbalanced among the different sentiment classes. This is because it is hard to collect a balanced set a priori, especially with the fact that it is totally up to the annotators to decide on which sentiments exist in a tweet, and more importantly how many. In fact, we started indeed with a data set extracted from a bigger one that was automatically annotated into positive and negative (using a previously trained model). The data set we uploaded for manual annotation was indeed balanced between the two sentiments.

Another critic that we address is the fact that we assumed that a tweet could contain exclusively positive or negative sentiments (neutral tweets are by definition ones that show no sentiment), which is a hard assumption that is not always true. In fact, as we explained in Section 5.5.2, several tweets were annotated as having sentiments of opposite polarities, which we have discarded for the sake of this work. As observed on our initial data set (before discarding any tweet), some sentiments tend to co-occur more than others. Namely, the sentiments “Love” and “Worry” co-occurred in many tweets where the tweeter is worried about something precious to him, or someone he cares about. In a similar way, in some tweets, the tweeters have shown both sentiments of “Boredom” and “Relief”, to express how bored they are of some event and how relieved they are it was over. As mentioned in Section 5.5.2, for the sake of this work, these tweets have been discarded. We consider only tweets with sentiments of a single polarity. This is because we rely on the results of classification to choose the set of sentiments from which we guess the actual sentiments of the tweet. This limits the potential of the proposed approach, and needs to be addressed in a future work.

5.7 Conclusion

In this chapter we have introduced the task of sentiment quantification in Twitter: for a given tweet, we tried to identify in a first step its sentiment polarity (whether it is positive, negative or neutral), and in a second step we tried to identify all the sentiments conveyed within it. We added several components to our previously introduced tool SENAT, to make the quantification task feasible and automated. Our proposed approach has proven to be good in detecting sentiments hidden in tweets with an average F1-score equal to 45.9% for 11 different sentiment classes.

We have also discussed the different potential misclassification reasons, and presented some solutions to enhance the performances of the proposed approach, which we will be dealing with as part of our future work. In our future work, we will also address the case of tweets with sentiments belonging to different polarities (i.e., tweets which have at the same time positive sentiments and negative ones), and try find possible ways to identify these sentiments.

Chapter 6

Conclusions and Future Work

The objective of this research has been to address some of the challenges in sentiment analysis. Throughout this work, we have tried to improve the performance of detection of sarcasm in social media, and perform reliable fine-grained sentiment analysis. The approaches proposed can be used in real world applications. As a matter of fact, we have demonstrated how to identify sarcasm and use it to enhance the performance of sentiment analysis. Nevertheless, we have built a tool we called SENTA which helps, through easy-to-use graphical user interface run the approaches we have proposed for multi-class sentiment analysis and sentiment quantification.

6.1 Contributions

In Chapter 2, we introduced our method to detect sarcasm in Twitter. The proposed method makes use of the different components of the tweet. It relies on Part-of-Speech-tags to extract patterns characterizing the level of sarcasm of tweets. The approach has shown good results, though might have even better results if we use a bigger training set since the patterns we extracted from the current one might not cover all possible sarcastic patterns. We also proposed a more efficient way to enrich our set with more sarcastic patterns using an initial training set of 6000 Tweets, and the hashtag *"#sarcasm"*. The overall accuracy obtained reached 83.1% with a precision of detection of sarcastic statements equal to 91.1% We have also demonstrated how to use this information (i.e., identifying whether a tweet is sarcastic or not) to enhance the performance of an existence sentiment analysis method.

In Chapter 3, we have proposed a new approach for sentiment analysis, where a set of tweets is to be classified into 7 different classes. The obtained results show some potential: the accuracy obtained for multi-class sentiment analysis in the data set used was 60.2%. However, we believe that a more optimized training set would present better performances. We demonstrated that multi-class sentiment analysis can achieve high accuracy level, but it remains a challenging task. Alongside, we have introduced SENTA, a tool we have built to demonstrate the efficiency of the proposed method, and to help perform sentiment analysis with no programming skills required, through an user-friendly interface.

In Chapter 4 we studied more deeply the task of multi-class sentiment analysis. We evaluated the evolution of various KPIs as the number of sentiment classes increased. We analyzed the difficulties of, and the different challenges involved with, multi-class classification, and proposed some metrics to measure the distance between sentiments (i.e., how similar they are to one another). We concluded that, even though the task of multi-class analysis is important, it might be more interesting to perform a sentiment detection task through which all of the sentiments present within a text are extracted.

In Chapter 5, we addressed the issues mentioned in Chapter 4. We have introduced the task of sentiment quantification in Twitter: for a given tweet, we tried to identify in a first step its sentiment polarity (whether it is positive, negative or neutral), and in a second step we tried to identify all the sentiments conveyed within it. We added several components to our previously introduced tool SENAT, to make the quantification task feasible and automated. Our proposed approach has proven to be good in detecting sentiments hidden in tweets with an average F1-score equal to 45.9% for 11 different sentiment classes. We have also discussed the different

potential misclassification reasons, and presented some solutions to enhance the performances of the proposed approach, which we will be dealing with as part of our future work.

6.2 Future Work

Our humble effort made to enhance sentiment analysis systems through the identification of sarcastic statements led us to believe that sophisticated forms of speech are indeed possible to identify using patterns. This is because, for most internet users, such sophisticated forms of speech are hard to come up with. Therefore, users are less creative and original, and tend to copy more creative ones. Writing patterns seem to have great potential in the field of NLP, and one possible direction for future work would be to explore this potential and see how far it leads in text classification tasks. Nevertheless, Natural Language Processing (NLP) has benefited from the advances in the field of Deep Learning (DL). In the recent years, several works have been proposed to perform several NLP tasks using DL techniques. However, we believe that finding language patterns can be done using such Neural Networks. The state-of-the-art works nowadays rely on what is referred to as Language Models (LM) to perform these tasks. We believe these LM can be further enhanced if it learns to recognize, not only attention models and relation between words, but also writing patterns.

Appendix A

List of Author's Publications and Awards

A.1 Journals

1. M. Bouazizi and T. Ohtsuki, "Multi-class sentiment analysis on twitter: Classification performance and challenges," in *Big Data Mining and Analytics*, vol. 2, no. 3, pp. 181–194, September 2019.

2. M. Bouazizi and T. Ohtsuki, "Multi-Class Sentiment Analysis in Twitter: What if Classification is Not the Answer," in *IEEE Access*, vol. 6, pp. 64486–64502, 2018.

3. H. Watanabe, M. Bouazizi and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," in *IEEE Access*, vol. 6, pp. 13825–13835, 2018.

4. M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," in *IEEE Access*, vol. 5, pp. 20617–20639, 2017.

5. M. Bouazizi and T. Otsuki Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," in *IEEE Access*, vol. 4, pp. 5477–5488, 2016.

A.2 Full Articles on International Conferences Proceedings

1. M. Bouazizi and T. Ohtsuki, "Sentiment Analysis in Twitter: From Classification to Quantification of Sentiments within Tweets," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington DC, December 2016.

2. M. Bouazizi and T. Ohtsuki, "Sentiment analysis: From binary to multi-class classification: A pattern-based approach for multi-class sentiment analysis in Twitter," in *Proc. IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, May 2016.

3. M. Bouazizi and T. Ohtsuki, "Sarcasm Detection in Twitter: "All Your Products Are Incredibly Amazing!!!" - Are They Really?," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, San Diego, CA, December 2015.

4. M. Bouazizi and T. Ohtsuki, "Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis," *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 1594–1597, Paris, August 2015.

A.3 Articles on Domestic Conference Proceedings

1. M. Bouazizi and T. Ohtsuki, "Multi-Class Sentiment Analysis in Twitter Using Machine Learning and Deep Learning", *IEICE General Conf*, Tokyo, March 2019.
2. M. Bouazizi and T. Ohtsuki, "A Deep Learning-Based Approach for Multi-Class Sentiment Analysis in Twitter", *IEICE General Conf*, Tokyo, March 2018.
3. M. Bouazizi and T. Ohtsuki, "Multi-Class Sentiment Analysis in Twitter," *IEICE Society Conf.*, Tokyo, September 2017.
4. M. Bouazizi and T. Ohtsuki, "Author Profiling in Twitter," *IEICE Society Conf.*, Japan, September 2017.
5. M. Bouazizi and T. Ohtsuki, "Sentiment Analysis in Twitter for Multiple Topics", *IEICE General Conf.*, Kyoto, March 2015.

A.4 Technical Reports

1. M. Bouazizi and T. Ohtsuki, "Sentiment Analysis in Twitter for Multiple Topics - How to Detect the Polarity of Tweets Regardless of Their Topic", *IEICE ASN*, January 2015.
2. M. Bouazizi and T. Ohtsuki, "Sarcasm Detection – How to Identify Sarcastic Statements in Twitter", *IEICE ASN*, July 2015.
3. M. Bouazizi and T. Ohtsuki, "Multi-Class Sentiment Analysis in Twitter – A Pattern-Based Approach", *IEICE ASN*, January 2016.
4. M. Bouazizi and T. Ohtsuki, "[Encouragement Talk] A Noval-Approach for Multi-Class Sentiment Analysis in Twitter", *IEICE ASN*, May 2016.
5. M. Bouazizi and T. Ohtsuki, "Multi-Class Sentiment Analysis in Twitter: from Classification to Quantification of Sentiments", *IEICE ASN*, July 2016.

A.5 Awards

1. The 32nd Telecommunications Dissemination Foundation Award, March 2016.
2. IEICE ASN Best paper Award, January 2016.

References

[1] M.S. O'Hern and L.R. Kahle, "The empowered customer: User-generated content and the future of marketing," *Global Economics and Management Review*, vol. 18, no. 1, pp. 22–30, Elsevier, 2013.

[2] A.Z. Bahtar and M. Muda, "The impact of User–Generated Content (UGC) on product reviews towards online purchasing–A conceptual framework," *Procedia Economics and Finance*, vol. 37, pp. 337–342, Elsevier, 2016.

[3] M.J. Mallen, S.X. Day, and M.A. Green, "Online versus face-to-face conversation: An examination of relational and discourse variables," *Psychotherapy: Theory, Research, Practice, Training*, vol. 40, no. 1–2, pp. 155. 2003.

[4] P.S. Lee, L. Leung, V. Lo, C. Xiong and T. Wu, "Internet communication versus face-to-face interaction in quality of life," *Social Indicators Research*, vol. 100. no. 3, pp. 375–389, Sprinter, 2010.

[5] M. Qiu and D. McDougall, "Foster strengths and circumvent weaknesses: Advantages and disadvantages of online versus face-to-face subgroup discourse," *Computers & Education*, vol. 67, pp. 1–11, 2013.

[6] R. Grieve, M. Indian, K. Witteveen, G.A. Tolan and J. Marrington, "Face-to-face or Facebook: Can social connectedness be derived online?," *Computers in human behavior*, vol. 29, no. 3, pp. 604–609, Elsevier, 2013.

[7] A. Java, X. Song, T. Finin and B. Tseng, "Why we twitter: understanding microblogging usage and communities," in *Proc. 9th WebKDD and 1st SNA-KDD workshop on Web mining and social network analysis,* pp. 56–65, Aug. 2007.

[8] M. Cha, H. Haddadi, F. Benevenuto, and P. K. Gummadi, "Measuring User Influence in Twitter: The Million Follower Fallacy," in *ICWSM*, vol. 10, No. 30, pp. 10–17, 2010.

[9] M. Trusov, A. Bodapati and R. E. Bucklin, "Determining Influential Users in Internet Social Networks," in *Journal of Marketing Research*, vol. 47, no. 4, pp. 643–658, American Marketing Association, 2010.

[10] J. Messias, L. Schmidt, R. Oliveira and F. Benevenuto, "You followed my Bot! Transforming Robots into Influential Users in Twitter," in *First Monday*, vol. 18, no. 7, 2013.

[11] C. G. Akcora, M. A. Bayir, M. Demirbas, and H. Ferhatosmanoglu, "Identifying breakpoints in public opinion," in *Proc. First Workshop on Social Media Analytics*, pp. 62–66, July 2010.

[12] R. Page, "The Linguistics of Self-Branding and Micro-Celebrity in Twitter: The Role of Hashtags," in *Discourse & communication*, vol. 6, no. 2, pp. 181–201, Sage Publications Sage UK: London, England, 2012.

[13] W. Xiao, L. Wen-zhong and D. Jian-gang, "The development research of the emotional contagion theory," in *Proc. IEEE Int. Conf. Software Engineering and Service Sciences*, Beijing, 2010, pp. 628–632.

[14] A. D. I. Kramer, J. E. Guillory and J. T. Hancock, "Experimental evidence of massive-scale emotional contagion through social networks," in *Proc. National Academy of Sciences*, vol. 111, no. 24, pp. 8788–8790, 2014.

[15] M. Gamon, "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis," in *Proc. ACL 20th Int. Conf. Computational Linguistics*, pp. 841, 2004.

[16] T. Rao and S. Srivastava, "Analyzing stock market movements using twitter sentiment analysis," in *Proc. IEEE/ACM ASONAM 2012*, pp. 119–123, 2012.

[17] M. Makrehchi, S. Shah and W. Liao, "Stock Prediction Using Event-Based Sentiment Analysis," in *Proc. 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 1, pp. 337–342, Washington, DC, 2013.

[18] M. Ghiassi, J. Skinner and D. Zimbra, "witter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," *Expert Systems with applications*, vol. 40, no. 16, pp. 6266–6282, Elsevier, 2013.

[19] D. Zimbra, M. Ghiassi and S. Lee, "Brand-related Twitter sentiment analysis using feature engineering and the dynamic architecture for artificial neural networks," in *Proc. 49th Hawaii Int. Conf. System Sciences (HICSS)*, pp. 1930–1938, 2016.

[20] X. Zhou, X. Tao, J. Yong and Z. Yang, "Sentiment analysis on tweets for social events," in *Proc. IEEE 17th Int. Conf. Computer Supported Cooperative Work in Design (CSCWD)*, pp. 557-562, Whistler, BC, 2013.

[21] A. Tumasjan, T. O. Sprenger, P. G. Sandner and I. M. Welpe, Isabell M, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. Fourth Int. AAAI Conf. weblogs and social media*, pp. 1–6, 2010.

[22] A. Bermingham and A. Smeaton, "On using Twitter to monitor political sentiment and predict election results," in *Proc. Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pp. 2–10, 2011.

[23] B. Pang, L. Lillian, and V. Shivakumar, "Thumbs up?: Sentiment classification using machine learning techniques," in *Proc. ACL-02 Conf. Empirical Methods in Natural Language Process.*, vol. 10, pp.79–86, July 2002.

[24] M. V. Mäntylä, D. Graziotin and M. Kuutila, "The evolution of sentiment analysis–A review of research topics, venues, and top cited papers," *Computer Science Review*, vo. 27, pp. 16–32, Elsevier, 2018.

[25] J.A. Richmond, "Spies in ancient Greece," *Greece and Rome (Second Series)*, vol. 45, no. 01, pp. 1–18, 1998.

[26] D.D. Droba, "Methods used for measuring public opinion," *American Journal of Sociology*, vol. 37, no. 3, pp. 410–423, University of Chicago Press, 1931.

[27] R. Stagner, "The cross-out technique as a method in public opinion analysis," *The Journal of Social Psychology*, vol. 11, no. 1, pp. 79–90, Taylor & Francis, 1940.

[28] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," *Standford University, CS229*, vol. 15, pp. 1–5, 2012.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," in *Proc. IEEE* vol., 86, no. 11, pp.2278–2324, Nov. 1998.

[30] G. E. Hinton, , S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets." *Neural computation*, vol. 18, no.7, pp. 1527–1554, 2006.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp. 1097–1105, 2012.

[32] T.T.Thet, J.C. Na, and C.S. Khoo, "Aspect-based sentiment analysis of movie reviews on discussion boards," *Journal of information science*, vol. 36, no. 6, pp.823–848, 2010.

[33] M. Wöllmer, F. Weninger, T. Knaup, B. Schuller, C. Sun, K. Sagae and L.P. Morency, "Youtube movie reviews: Sentiment analysis in an audio-visual context," *IEEE Intelligent Systems*, vol. 28, no. 3, pp.46–53, 2013.

[34] S. Kiritchenko, X. Zhu, C. Cherry and S. Mohammad, "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews," in *Proc.Int. Workshop Semantic Evaluation (SemEval 2014)*, pp. 437–442, 2014.

[35] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 5, 2015.

[36] A. Pak, and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," *LREc*, Vol. 10, No. 2010, pp. 1320–1326, May 2010.

[37] A. Agarwal, B. Xie, I. Vovsha, O. Rambow and R. Passonneau, "Sentiment analysis of twitter data," in *Proc. Workshop Language in Social Media (LSM 2011)*, pp. 30–38, 2011.

[38] T. Wilson, J. Wiebe and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis", in emphProc. HLT/EMNLP-05, pp. 347–354, 2005.

[39] A. Ortigosa, J.M. Martín and R.M and Carro, "Sentiment analysis in Facebook and its application to e-learning", *Computers in Human Behavior*, vol.31 , pp. 527–541, 2014.

[40] O. Popescu, C. Strapparava, "Time corpora: epochs, opinions and changes", *Knowledge-Based Systems*, vol. 69, pp. 3–13, 2014.

[41] H. Kanayama, T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proc. Conf. Empirical Methods in NLP*, pp. 355–363, Association for Computational Linguistics, July 2006

[42] S. Wang, D. Li, X. Song, Y. Wei, H. Li, "A feature selection method based on improved Fisher's discriminant ratio for text sentiment classification," *Expert Systems with Applications*, vol. 38, pp. 8696–8702, 2011.

[43] C. Banea, R. Mihalcea, J. Wiebe, "Sense-level subjectivity in a multilingual setting," *Computer Speech & Language*, vol. 28, pp. 7–19, 2014.

[44] F. Bravo-Marquez, M. Mendoza, and B. Poblete, "Meta-level sentiment models for big social data analysis," *Knowledge-Based Systems*, vol. 69, pp. 86–99, 2014.

[45] M.D. Molina-González, E. Martínez-Cámara, M.T. Martín-Valdivia and J.M. Perea-Ortega, "Semantic orientation for polarity classification in Spanish reviews," *Expert Systems with Applications*, vol. 40, no. 29, pp. 7250–7257, Elsevier, 2013.

[46] K. Hiroshi, N. Tetsuya, W. Hideo, "Deeper sentiment analysis using machine translation technology," in *Proc. ACL Int. Conf. Computational Linguistics (COLING)*, pp. 494, 2004.

[47] H. Wang, P. Yin, L. Zheng, James N.K. Liu, "Sentiment classification of online reviews: using sentence-based language model," *Journal of Experimental & Theoretical Artificial Intelligence* vol. 26, no. 1, pp. 13–31, Taylor & Francis, 2014.

[48] M.T. Martín-Valdivia, E. Martínez-Cámara, J.M. Perea-Ortega, and L.A. Ureña-López, "Sentiment polarity detection in Spanish reviews combining supervised and unsupervised approaches," *Expert Systems with Applications*, vol. 40, no. 10, pp. 3934–3942, Elsevier, 2013.

[49] A. Heydari, M. A. Tavakoli, N. Salim, Z. Heydari, "Detection of review spam: a survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, Elsevier, 2015.

[50] M. Ott, C. Cardie, J.T. Hancock, "Negative deceptive opinion spam," in *Proc. Conf. North American Chapter ACL: Human Language Technologies*, pp. 497–501, June 2013.

[51] M. Ott, Y. Choi, C. Cardie, J.T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in *Proc. Annual Meeting ACL: Human Language Technologies*, vol. 1, pp. 309–319, 2011.

[52] S. Banerjee and Alton Y.K. Chua, "Applauses in hotel reviews: Genuine or deceptive?," in *IEEE Science and Information Conf. (SAI)*, pp. 938–942, 2014.

[53] Y. Liu, J. Jin, P. Ji, J.A. Harding, R.Y.K. Fung, "Identifying helpful online reviews: a product designer's perspective," *Computer-Aided Design*, vol. 45, no. 2, pp. 180-194, Elsevier, 2013.

[54] S. Krishnamoorthy, "Linguistic features for review helpfulness prediction," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3751–3759, Elsevier, 2015.

[55] N. Purnawirawan, P. De Pelsmacker, N. Dens, "Balance and sequence in online reviews: how perceived usefulness affects attitudes and intentions," *Journal of interactive marketing*, vol. 26, no. 4, pp. 244–255, Elsevier, 2012.

[56] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive Language Detection in Online User Content," in *Proc. WWW'16*, pp. 145–153, Apr. 2016.

[57] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in Twitter to improve information filtering," in *Proc. 33rd Int. ACM SIGIR Conf. Research and development in information retrieval,* pp. 841–842, July 2010.

[58] M. A. Cabanlit and K. J. Espinosa, "Optimizing N-gram based text feature selection in sentiment analysis for commercial products in Twitter through polarity lexicons," in *Proc. 5th Int. Conf. Inform., Intelligence, Syst. and Applicat.,* pp. 94–97, July 2014.

[59] U. R. Hodeghatta, "Sentiment analysis of Hollywood movies on Twitter," in *Proc. IEEE/ACM ASONAM,* pp. 1401–1404, Aug. 2013.

[60] O. Almatrafi, S. Parack and B. Chavan, "Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014," in *Proc. ACM Int. Conf. Ubiquitous Information Management and Communication*, pp. 41, 2015.

[61] D. Paul, F. Li, M.K. Teja, X. Yu and R. Frost, "Compass: Spatio-Temporal Sentiment Analysis of US Election What Twitter Says!," in *Proc. ACM SIGKDD*, pp. 1585–1594, 2017.

[62] Y. Choi, E. Breck, C. Cardie, "Joint extraction of entities and relations for opinion recognition," in *ACL Proc. Conf. Empirical Methods in NLP*, pp. 431–439, 2006.

[63] M. Wiegand and D. Klakow, "Convolution kernels for opinion holder extraction," *ACL Human language technologies: Annual Conf. North American Chapter of the Association for Computational Linguistics*, pp. 795–803, 2010.

[64] A. Katiyar and C. Cardie, "Investigating LSTMs for joint extraction of opinion entities and relations," in *Proc. Annual Meeting ACL*, vol. 1, pp.919–929, 2016.

[65] W. Che, Y. Zhao, H. Guo, Z. Su and T. Liu, "Sentence Compression for Aspect-Based Sentiment Analysis," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2111-2124, Dec. 2015.

[66] V. K. Singh, R. Piryani, A. Uddin and P. Waila, "Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification," *Int. Mutli-Conf. Automation, Computing, Communication, Control and Compressed Sensing*, pp. 712-717, Kottayam, 2013.

[67] T.H. Nguyen and K. Shirai, "Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods in NLP*, pp. 2509–2514, 2015.

[68] J. M. Soler, F. Cuartero, and M. Roblizo, "Twitter as a tool for predicting elections results," in *Proc. IEEE/ACM ASONAM*, pp. 1194–1200, Aug. 2012.

[69] S. Homoceanu, M. Loster, C. Lofi, and W-T. Balke, "Will I like it? Providing product overviews based on opinion excerpts," in *Proc. IEEE CEC,* pp. 26–33, Sept. 2011.

[70] C. Liebrecht, F. Kunneman, and A. Van Den Bosh, "The perfect solution for detecting sarcasm in tweets #not," in *Proc. WASSA 2013,* pp. 29–37, June 2013.

[71] D. Maynard, and M. Greenwood, "Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis," in *Proc. 9th Int. Conf. Language Resources Evaluation,* pp. 4238–4243, May 2014.

[72] R. L. Brown, "The pragmatics of verbal irony," *Language use and the uses of language,* pp. 111–127, 1980.

[73] S. Attardo, "Irony as relevant inappropriateness," Irony in language and thought, pp. 135–174, June 2007.

[74] R. W. Gibbs and J. O?Brien., "Psychological aspects of irony understanding," Journal of Pragmatics, pp. 523–530, Dec. 1991.

[75] H. Grice, "Further notes on logic and conversation," Pragmatics: syntax and semantics, pp. 113–127, Academic Press, 1978.

[76] O. Tsur, D. Davidov, and A. Rappoport. "ICWSM ? A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," in *Proc. AAAI Conf. Weblogs and Social Media,* pp 162-?169, May 2010.

[77] D. Davidov, O. Tsur, and A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," In *Proc.14th Conf. on Computational Natural Language Learning*, pp. 107–116, July 2010.

[78] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on Twitter: A behavioral modeling approach," in *Proc. 18th ACM Int. Conf. Web Search Data Mining,* pp. 79–106, Feb. 2015.

[79] A. Joshi, P. Bhattacharyya, and M.J. Carman, "Automatic sarcasm detection: A survey," *arXiv,* Feb. 2016.

[80] M. W. Berry, "Survey of text mining: Clustering, classification, and retrieval", 2004.

[81] M. Boia, B. Faltings, C.-C. Musat and P. Pu, "A :) is worth a thousand words: How people attach sentiment to emoticons and words in tweets," in *Proc. Int. Conf. Social Computing*, pp. 345–350, Sept. 2013.

[82] K. Manuel, K. V. Indukuri and P. R. Krishna, "Analyzing internet slang for sentiment mining," in *Proc. 2nd Vaagdevi Int. Conf. Inform. Technology for Real World Problems*, pp. 9–11 Dec. 2010.

[83] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Ann. Meeting on Assoc. for Computational Linguistics*, vol. 2, pp. 49–54, June 2014.

[84] F. Jr. Sting, *The meaning of irony.* New York, State University of NY, 1994.

[85] S. G. Shamay-Tsoory, R. Tomer, and J. Aharon-Peretz, "The neuroanatomical basis of understanding sarcasm and its relationship to social cognition," *Neuropsychology,* vol. 19, no. 3, pp. 288–300, May 2005.

[86] C. Burfoot and T. Baldwin., "Automatic satire detection: Are you having a laugh?" in *Proc. ACL-IJCNLP 2009,* pp. 161–164, Aug. 2009.

[87] J. D. Campbell and A. N. Katz, "Are there necessary conditions for inducing a sense of sarcastic irony?," *Discourse Processes,* pp. 459-?480, Aug. 2012.

[88] D. Wilson, "The pragmatics of verbal irony: Echo or pretence?," *Lingua,* Vol. 116, no. 10, pp. 1722–1743, Oct. 2006.

[89] S. L. Ivanko and P. M. Pexman, "Context incongruity and irony processing," *Discourse Processes,* vol. 35, no. 3, pp. 241–279, 2003.

[90] R. Giora, "On irony and negation," *Discourse Processes,* vol. 19, no. 2, pp. 239?264, 1995.

[91] J. Tepperman, D. Traum, and S. S. Narayanan, "Yeah right: Sarcasm recognition for spoken dialogue systems," in *Proc. InterSpeech,* pp. 1838–184, Sept. 2006.

[92] T. Veale and Y. Hao, "Detecting ironic intent in creative comparisons," in *Proc. ECAI,* pp. 765–770, Aug. 2010.

[93] F. Barbieri, H. Saggion, and F. Ronzano, "Modelling sarcasm in Twitter, a novel approach," in *Proc. WASSA,* pp. 50-58, June 2014.

[94] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *Proc EMNLP*, pp.1003-1012, Sep. 2015.

[95] Z. Wang, Z. Wu, R. Wang, and Y. Ren, "Twitter sarcasm detection exploiting a context-based model," in *Proc. Web Inform. Syst. Eng. (WISE),* pp. 77–91, Nov. 2015.

[96] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive sentiment and negative situation," in *Proc. Conf. Empirical Methods Natural Language Processing,* pp. 704–714, Oct. 2013.

[97] S. Muresan, R. Gonzalez-Ibanez, D. Ghosh, and N. Wacholder, "Identification of nonliteral language in social media: A case study on sarcasm," *Journal Assoc. Inform. Sci. and Technology,* Jan. 2016.

[98] E. Fersini, F. A. Pozzi, and E. Messina, "Detecting irony and sarcasm in microblogs: The role of expressive signals and ensemble classifiers," in *Proc. IEEE Data Sci. and Advanced Analytics (DSAA),* pp. 1–8, Oct. 2015.

[99] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in Twitter data," in *Proc. IEEE/ACM ASONAM 2015,* pp. 1373-?1380, Aug. 2015.

[100] A. Reyes, P. Rosso, and D. Buscaldi, "From humor recognition to irony detection: The figurative language of social media," *Data & Knowledge Engineering,* vol. 74, pp. 1–12, Apr. 2012.

[101] A. Reyes, P. Rosso, and T. Veale, "A multidimensional approach for detecting irony in Twitter," *Language Resources and Evaluation,* vol. 47, no. 1, pp. 239–268, Mar. 2013.

[102] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proc. Annu. Meeting Assoc. Computational Linguistics, Int. Joint Conf. Natural Language Processing (ACL-IJCNLP),* vol. 2, pp. 757?-762, July 2015.

[103] D. Bamman and N. A. Smith, "Contextualized sarcasm detection on Twitter,"in *Proc. AAAI Int. Conf. on Web and Social Media (ICWSM),* pp. 574–77, May 2015.

[104] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva, "Twitter part-of-speech tagging for all: Overcoming sparse and noisy data," in *Proc. Int. Conf. RANLP,* pp. 198–206, Sept. 2013

[105] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten "The WEKA data mining software: An update',' *SIGKDD Explor. Newsk.,* vol. 11, no. 1, pp. 10–18, June 2009.

[106] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology,* vol. 2, no. 3, pp 20:1–27:27, Apr. 2011.

[107] E. Camp, "Sarcasm, pretense, and the semantics/pragmatics distinction*," *Noûs,* vol. 46, No. 4, pp. 587?-634, Dec. 2012.

[108] M. S. Neethu and R. Rajasree, "Sentiment analysis in Twitter using machine learning techniques," in *Proc. 4th Int. Conf. Computing, Commun. and Networking Technologies*, pp. 1–5, July 2013.

[109] L. Breiman, "Random Forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Jan. 2001.

[110] S. Attardo, "Irony markers and functions - Towards a goal-oriented theory of irony and its processing" *Rask,*, vol. 12, no. 1, pp. 3–20, 2000.

[111] P. Rockwell. "Vocal features of conversational sarcasm: A comparison of methods," *Journal of psycholinguistic research,* vol. 36, no. 4, pp. 361–369, Sep. 2007.

[112] P. Rockwell, "Empathy and the expression and recognition of sarcasm by close relations or strangers," *Perceptual and Motor Skills,* vol. 97, no. 1, pp. 251–256, Aug. 2003.

[113] M. Bouazizi and T. Ohtsuki, "Sentiment Analysis in Twitter for Multiple Topics - How to Detect the Polarity of Tweets Regardless of Their Topic," in *Proc. IEICE ASN*, pp 91–96, Feb. 2015.

[114] B. O'Connor, R. Balasubramanyan, B. Routledge, and N. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proc. Int. AAAI Conf. Weblogs and Social Media*, pp. 26–33, May 2010.

[115] K. Ghag and K. Shah, "Comparative analysis of the techniques for sentiment analysis," in *Proc. Int. Conf. Advances in Technology and Eng.*, pp. 1–7, Jan. 2013.

[116] Y. R. Tausczik J. W. Pennebaker, "The psychological meaning of words: LIWC and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, Dec. 2010

[117] W. Gao and F. Sebastiani, "Tweet Sentiment: From Classification to Quantification," in *Proc. IEEE/ACM ASONAM*, pp. 97–104, Aug. 2015.

[118] Y.H.P.P. Priyadarshana, K.I.H. Gunathunga, K.K.A. Nipuni N.Perera, L. Ranathunga, P.M. Karunaratne, and T.M. Thanthriwatta, "Sentiment analysis: Measuring sentiment strength of call centre conversations," in *Proc. IEEE ICECCT*, pp.1–9, March 2015.

[119] R. Srivastava and M.P.S. Bhatia, "Quantifying modified opinion strength: A fuzzy inference system for Sentiment Analysis," in *Proc. Int. Conf. Advanced in Computing, Communications and Informatics*, pp.1512–1519, Aug. 2013.

[120] K.H. Lin, C. Yang and H.Chen, "What emotions do news articles trigger in their readers?," in *Proc. ACM SIGIR '07*, pp. 733–734, July 2007.

[121] K.H. Lin, ; C. Yang and H Chen, "Emotion classification of online news articles from the reader's perspective," in *Proc. IEEE/WIC/ACM WI-IAT '08*, vol.1, pp.220–226, Dec. 2008.

[122] L. Ye, R. Xu and J. Xu, "Emotion prediction of news articles from reader's perspective based on multi-label classification," in *Proc. Int. Conf. Machine Learning and Cybernetics*, vol.5, pp. 2019–2024, July 2012.

[123] W. Liang, H. Wang, Y. Chu and C. Wu, "Emoticon recommendation in microblog using affective trajectory model," in *Proc. Annual Summit and Conf. Asia-Pacific Signal and Information Processing Association (APSIPA)*, pp.1–5, Dec. 2014.

[124] R. Xia, F. Xu, C. Zong, Q. Li, Y. Qi and T. Li, "Dual sentiment analysis: Considering two sides of one review," *IEEE Trans. Knowledge and Data Engineering (TKDE)*, vol. 27, no. 8, pp. 2120–2133, Aug. 2015.

[125] C. Fellbaun, *WordNet: an Electronic Lexical Database*, Cambridge, Massachusetts, 1998.

[126] M. Bouazizi and T. Ohtsuki, "Sarcasm Detection in Twitter:"All Your Products Are Incredibly Amazing!!!" - Are They Really?" in *Proc. IEEE Globecom*, pp. Dec. 2015.

[127] M. Bouazizi and T. Ohtsuki, "Sentiment snalysis: from binary to multi-class classification - A pattern-based approach for multi-class sentiment analysis in Twitter," in *Proc. IEEE ICC*, pp. 1–6, May 2016.

[128] M. Bouazizi and T. Ohtsuki, "Sentiment analysis in Twitter: From classification to quantification of sentiments within tweets" in *Proc. IEEE GLOBECOM*, pp. 1–6, Dec. 2016.

[129] M. Bouazizi and T. Ohtsuki, "A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," in *IEEE Access*, vol. 5, pp. 20617–20639, 2017.

[130] A. Kumar and T. M. Sebastian, "Sentiment Analysis on Twitter," in *Int. Journal Computer Science Issues (IJCSI)*, vol. 9, no. 3, pp. 372–378, Citeseer, 2012.

[131] E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Urena-López and A. R. Montejo-Ráez, "Sentiment Analysis in Twitter," in *Natural Language Engineering*, vol. 20, no. 1, pp. 1–28, Cambridge University Press, 2014.

[132] H. Saif, Y. He and H. Alani, "Semantic Sentiment Analysis of Twitter," in *Proc. Int. Semantic Web Conf.*, pp. 508–524, 2012.

[133] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno and J. Caro, "Sentiment Analysis of Facebook Statuses Using Naive Bayes Classifier for Language Learning," in *Proc. Int. Conf. Information, Intelligence, Systems and Applications (IISA)*, pp. 1–6, 2013.

[134] H. M. Zin, N. Mustapha, M. A. A. Murad, and N. M. Sharef, "Term Weighting Scheme Effect in Sentiment Analysis of Online Movie Reviews," in *Advanced Science Letters*, vol. 24, no. 2, pp. 933–937, American Scientific Publishers, 2018.

[135] Z. Kuncheva and G. Montana, "Community detection in multiplex networks using locally adaptive random walks," in *Proc. IEEE/ACM ASONAM,* pp. 1308–1315, Aug. 2015.

[136] I. Bizid, N. Nayef, P. Boursier, S. Faiz and J. Morcos, "Prominent users detection during specific events by learning on-and off-topic features of user activities," in *Proc. IEEE/ACM ASONAM,* pp. 500–503, Aug. 2015.

[137] N. R. Griffin, C. R. Fleck, M. G. Uitvlugt, S. M. Ravizza and K. M. Fenn, "The tweeter matters: Factors that affect false memory from Twitter," in *Computers in Human Behavior*, vol. 77, pp. 63–68, Elsevier, Dec. 2017.

[138] H. Allcott and M. Gentzkow, "Social media and fake news in the 2016 election," in *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 211–36, American Economic Association, 2017.

[139] P. Achananuparp, E. P. Lim, J. Jiang and T. A. Hoang, "Who is retweeting the tweeters? modeling, originating, and promoting behaviors in the twitter network," in *ACM Trans. Management Information Systems (TMIS)*, vol. 3, no. 3, pp 13, Oct. 2012.

[140] E. Kassens-Noor, "Twitter as a teaching practice to enhance active and informal learning in higher education: The case of sustainable tweets," in *Active Learning in Higher Education*, vol. 13, no. 1, pp. 9–21, SAGE Publications Sage UK, Feb 2012.

[141] A. Dhir, K. Buragga and A. A. Boreqqah, "Tweeters on campus: Twitter a learning tool in classroom?," in *J. Universal Computer Science*, vol. 19, no. 5, pp. 672–691, 2013.

[142] N. Zainuddin, A. Selamat and R. Ibrahim, "Hybrid sentiment classification on twitter aspect-based sentiment analysis," in *Applied Intelligence* vol. 48, no. 5, pp. 1218–1232, Springer, May 2018.

[143] A. Bhoi and S. Joshi, "Various Approaches to Aspect-based Sentiment Analysis," arXiv:1805.01984 [cs], May 2018.

[144] A. Y and D. Chang, "Multiclass Sentiment Prediction using Yelp Business Reviews," 2015.

[145] O. Araque, I. Corcuera-Platas, J. F. Sanchez-Rada and C.A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," in *Expert Systems with Applications,* vol. 77, pp. 236–246, Elsevier, Feb. 2017.

[146] B. Krawczyk, B. T. McInnes and A. Cano, "Sentiment Classification from Multi-class Imbalanced Twitter Data Using Binarization," in *Proc. Int. Conf. Hybrid Artificial Intelligence Systems*, pp. 26–37, June 2017.

[147] J. Barranquero, J. Diez and J. J. del Coz., "Quantification-Oriented Learning Based on Reliable Classifiers," in *Pattern Recognition*, vol. 48, No. 2, pp. 591–604, 2015.

[148] A. Bella, C. Ferri, J. Hernandez-Orallo and M.J. Ramirez-Quintana, "Quantification via probability estimators," in *Proc. 11th IEEE Int. Conf. Data Mining (ICDM 2010)*, pp. 737–742, Sydney, AU, 2010.

[149] A. Esuli and F. Sebastiani, "Optimizing Text Quantifiers for Multivariate Loss Functions" in *ACM Trans. Knowledge Discovery from Data*, vol. 9, no. 4, pp. 1–27, 2015.

[150] G. Forman, "Quantifying Counts and Costs Via Classification, in *Data Mining and Knowledge Discovery*, vol. 17, no. 2, pp. 164–206, 2008.

[151] R. Quinlan "C4.5: Programs for Machine Learning," Morgan Kaufmann Publishers, San Mateo, CA, 1993.