MASTER IN COMPUTER SCIENCE PROFESSIONAL FOCUS IN SOFTWARE ENGINEERING

Twitter sentiment analysis using n-grams approach and Deep Learning

Rémi, Cornet

Award date:
2018

Awarding institution:
University of Namur

Université de Namur
Faculty of Computer Science
Academic Year 2017–2018

Twitter Sentiment Analysis using n-grams approach and Deep Learning

Rémi CORNET

Internship mentor: Miguel GARCIA TORRES

Supervisor: _____ (Signed for Release Approval - Study Rules art. 40)
Wim VANHOOF

A thesis submitted in the partial fulfillment of the requirements
for the degree of Master of Computer Science at the Université of Namur

UNIVERSITY OF NAMUR

Abstract

Faculty of Computer Science

Master Degree in Computer Science

by Rémi CORNET

Today, social networks and e-commerce platforms occupy a huge place in our society. These media are an important source of messages from users expressing an opinion or sentiment whether about an event or a commercial product. These subjective messages contain a wealth of information that is difficult to analyze manually and, for several years, a discipline has emerged that seeks to automate the analysis of this data: sentiment analysis. The Twitter micro-blogging platform, by its number of users and its number of daily messages is an interesting resource to work on this kind of content. In this document, existing sentiment analysis techniques are presented and various publications in the field are detailed. The main role of this document is to investigate the ability of an approach coupling neural networks and n-grams of messages posted on Twitter to provide good results as part of a sentence level sentiment classification. To achieve this objective, a pipeline was set up to cover all the operations required to carry out this experiment: data collection and cleaning, dataset preparation and training of the neural network.

Aujourd'hui, les réseaux sociaux et les plateformes de commerce électronique occupent une place énorme dans notre société. Ces médias sont une source importante de messages d'utilisateurs exprimant une opinion ou un sentiment, qu'il s'agisse d'un événement ou d'un produit commercial. Ces messages subjectifs contiennent une mine d'informations difficiles à analyser manuellement et, depuis plusieurs années, une discipline qui cherche à automatiser l'analyse de ces données a vu le jour : l'analyse des sentiments. La plateforme de micro-blogging Twitter, par son nombre d'utilisateurs et son nombre de messages quotidiens est une ressource intéressante pour travailler sur ce type de contenu. Dans ce document, les techniques d'analyse de sentiment existantes sont présentées et diverses publications sur le sujet sont détaillées. Le rôle principal de ce document est d'étudier la capacité d'une approche couplant l'utilisation des réseaux de neurones et de n-grammes de messages postés sur Twitter à fournir de bons résultats dans le cadre d'une classification des sentiments au niveau de la phrase. Pour atteindre cet objectif, un pipeline a été mis en place pour couvrir toutes les opérations nécessaires à la réalisation de cette expérience : collecte et nettoyage des données, préparation du jeu de données et entrainement du réseau de neurones.

Acknowledgements

The following work was completed during the 2017-208 academic year. This work stems from the research internship I did between September and December 2017 at the Pablo de Olavide University in Seville.

I would like to thank my supervisor, Mr Vanhoof, for introducing this subject and for putting me in contact with the University Pablo de Olavide. I would also like to thank him for the help he gave me in writing this work through his advice and guidance.

I would also like to thank my internship mentor, Mr. Garcia Torres, for his advice and follow-up during my internship. This experience allowed me, under his guidance, to familiarize myself with a field of research that aroused my curiosity.

Contents

List of Figures

List of Tables

List of Abbreviations

ML	**M**achine **L**earning
SA	**S**entiment **A**nalysis
NLP	**N**atural **L**anguage **P**rocessing
DP	**D**eep **L**earning
POS	**P**art-**O**f-**S**peech
NN	**N**eural **N**etwork
RNN	**R**ecurrent **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
DT	**D**ecision **T**ree
SVM	**S**upport **V**ector **M**achines
TF-IDF	**T**erm **F**requency-**I**nverse **D**ocument **F**requency
BoW	**B**ag of **W**ords

Chapter 1

Introduction

1.1 Context

Humans have a unique ability: they are able to express an opinion on a subject. They have always used this ability within the limits of the means of communication available to them. Today, the Internet has become the preferred platform for self-expression and communication: web users can express their opinions on important events or political decisions to a large public in no time. On 30 June 2017, there were nearly 3.9 billions Internet users, representing 51.7% of the world population [1]. In parallel with this growth, we saw the emergence of the concept of social networking in the 2000s [2], which brought a new way of communicating and expressing opinions to a much larger audience.

The micro-blogging platform Twitter is one of the leaders in this field. It alone has 330 million monthly users in the third quarter of 2017 [3] and this mass of user writes an average of 500 million Tweets per day [4].This affluence of subjective publications has made the task of analyzing opinions and sentiments in Twitter posts very relevant both for private companies seeking to know what is being said about them but also for researchers who want to study the link between these opinions and the real world.

1.2 What is Sentiment Analysis ?

What we call Sentiment Analysis (or Opinion Mining) is an area of research aiming at collecting and analyzing subjective data from humans in order to extract relevant information by algorithmic means. It is a recent discipline whose development took place in conjunction with the emergence of social networks in the 2000s. The growth of this field has been so rapid during this period that 99% of the publications on this subject were published after 2004 [5]. This characteristic makes the Sentiment Analysis domain one of the most rapidly evolving domains in the world.

Humans have always been interested in the opinions of others, but until recently they did not have effective ways of gathering and analyzing a large quantity of these opinions. Text analysis techniques and Natural Language Processing have changed the game by providing researchers and businesses with tools to effectively collect and analyze such data. Sentiment analysis

is a field with varied applications such as [6]: politics, sales performance, products ranking or stock market prediction.

Sentiment analysis is a classification task sometimes made difficult by the lack of formalism and consistency in human language. Indeed, there are many factors that may seem obvious to a human but that pose problems to a program: irony, sarcasm, abbreviations or typing errors, smileys.

There are several ways to approach sentiment analysis, the two main approaches are based on the use of lexicon or machine learning. Techniques based on lexicons use semantic information of words to achieve their mission while the machine learning approach relies on the use of dataset to train a program to deduce this information.

This thesis will try to study sentiment analysis on data coming from Twitter and will focus on the machine learning approach and more specifically on a sub-branch of the machine learning: Deep Learning.

1.3 What is Deep Learning ?

Deep Learning, a sub-branch of Machine Learning, is a set of methods whose general idea is to simulate a network of neurons like those present in the human brain to create computer programs capable of learning from their experiences to improve the way they perform a task. Deep Learning beginnings date back to the 1940s, at the dawn of computing. Although of a certain age, it was only recently that we were able to witness the massive appearance of Deep Learning in the industry when it was previously confined to academia.

The reason for this is that this technology has not always been as successful as it is today because it did not give satisfactory results and it is only thanks to the increase in computing power available in recent years that these methods have been able to provide convincing results.

1.4 Problem definition

In the course of 2017, Catalonia, an autonomous community of Spain held a referendum on the independence of this region from the country. This referendum was the scene of intense tensions between Spanish citizens and beyond.

This event generated a significant amount of content on the Internet in general and on the micro-blogging platform Twitter. This platform allowed users to express their opinion on the referendum and the events surrounding it.

In a situation of instability such as this, a country's economy can be undermined by the lack of confidence of national and international actors in the stability of its institutions. In this particular situation, following the announcement of the results of the referendum, we have seen a sharp fall of more than 2% in the IBEX35 index on the Madrid Stock Exchange [7].

In this thesis, we will analyze the content generated on Twitter around this event.

More generally, we are developing a framework to collect data from Twitter related to a particular topic and analyze the general sentiment around it.

1.5 Thesis contribution

This thesis aims, firstly, to study and present the state of the art in the area of Sentiment Analysis. This chapter will present the general problems behind the sentiment analysis and the different approaches used. The difference between the lexicon approach and the machine learning approach will be explained and in these two families, the sub-branches they contain will be explored.

Second, it aims to build a framework for collecting data about a particular topic or event on Twitter and pre-process it to prepare it for use by Deep Learning technique. This part occupies an important place in the global process because in the field of machine learning, it is imperative to obtain data of sufficient quality and quantity to achieve an effective training.

And finally, it aims to parameterize and train a model, based on the data collected previously, allowing to analyze the subjective sentiment behind a Tweet and by extension, to analyze daily the global feeling that emerges around an event.

This framework will be complete in the sense that it will cover the whole process of sentiment analysis: from collecting data on a particular subject from the Twitter platform to creating a sentiment analysis model.

1.6 Thesis organization

This thesis is organized as follows :

- **Chapter 2 : State of the art Sentiment Analysis :** presents an overview of Sentiment Analysis most used techniques.
- **Chapter 3 : Processing Pipeline :** presents our complete approach from data collection to results.
- **Chapter 4 : Result Analysis :** presents the experimental results and their interpretation.
- **Chapter 5 : Conclusion :** presents the thesis conclusion.

Chapter 2

Sentiment Analysis State of the Art

This chapter presents an overview of the different techniques used to perform sentiment analysis. Different approaches and algorithms are also presented.

2.1 Opinion

Before beginning to present the different techniques of sentiment analysis, it is necessary to define what is meant by sentiment or opinion. In [6], Bing Liu defines an opinion as follows:

> An opinion is a quadruple, (g, s, h, t), where g is the opinion (or sentiment) target, s is the sentiment about the target, h is the opinion holder and t is the time when the opinion was expressed.

We can try to clarify this definition by taking a simple example:

> SomeUser, 01/02/2018 : "After testing the new iPhone for a whole week, I have to say that I am seduced by this device. It is light, simple and very powerful. The quality of the photos is sublime. I do not regret my purchase although my wallet does :("

In this example, we quickly identify h and t, which are respectively SomeUser and February 1, 2018.

Where the task becomes more complex is when it comes to identifying the target of the opinion (g): the author mentions a new iPhone but the exact model of the device in question is not explicitly specified. A human reader could be aware of the latest high-tech releases but this is not the case of a program that would read this message.

Secondly, it is necessary to identify the opinion expressed by the message (s). Again, a human can easily detect the different opinions expressed and relate them to the device they describe. The task is not so simple when it comes to automatic processing especially when the opinion is spread over several sentences or when it contains a note of humour or smileys.

This example allows a superficial understanding of the complexity of sentiment analysis and to understand why the development of tools to perform such tasks is difficult.

2.2 Sentiment Analysis Definition

Sentiment analysis techniques have been studied for about 30 years now. During the 2000s, this discipline attracted many scientists because it had applications in many fields and the amount of data with sentimental connotations exploded [5].

In [6], Bing Liu defines sentiment analysis as follows:

> "Sentiment analysis, also called opinion mining, is the field of study that analyzes people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes."

This definition provides a good overview of the problems the discipline is trying to solve: extract the opinion underlying a set of text, classify it into a category, analyze these results and use them to make informed decisions.

There are several ways to classify opinions. The most classical way to classify them is to make a binary classification separating negative and positive texts. It is also possible to achieve a more granular classification by increasing the number of target categories: we can for example use a "neutral" category or subdivide existing categories to separate sentiments by intensity level (by trying, for example, to detect if a text is very negative or simply negative) [6].

2.3 Level of analysis

Another dimension in which we can study feelings is the granularity of the text from which we will try to extract information. In [6], Bing Liu details 3 levels of granularity :

- Document level: At this level of granularity, we try to gather the general opinion that emerges from an entire document without distinguishing between the different elements that make up the document. For this reason, it is important to realize that this technique cannot be applied on a document expressing opinions on different entities since only one general opinion will be extracted.
- Sentence level: At this level, we analyze a sentence and try to discover what opinion it express.
- Aspect level: At this level, we look directly at the opinion rather than analyzing language structure like documents or sentences.

2.4 Different approaches

Since the emergence of this discipline in the research community. Many approaches have been developed to try to solve the problems of sentiment analysis. Figure 2.1, inspired of the sentiment analysis algorithms and applications survey of Walaa Medhat et al. [8], provides an overview of the different approaches that have been taken to date. We can see that there are two main families of techniques that coexist: those based on machine learning and those based on the lexicon approach.

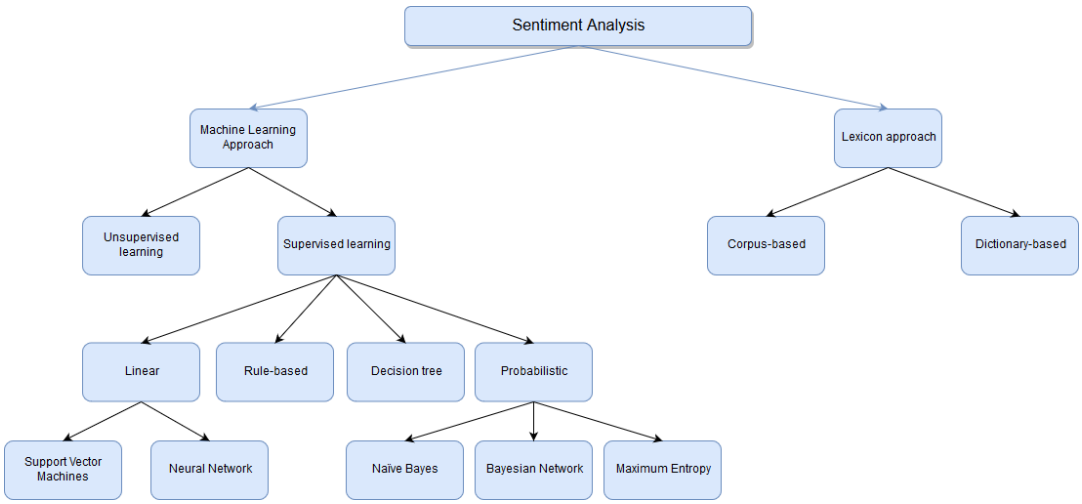

FIGURE 2.1: Sentiment analysis approachs. Inspired of [8]

2.5 Machine learning approach

This approach to sentiment analysis relies on the use of machine learning techniques (supervised and unsupervised) to train and construct models capable of classifying texts according to the opinion they contain.

2.5.1 Unsupervised learning

In Machine Learning, an algorithm is said to be "unsupervised" when it works on unlabeled datasets to find patterns in the data. This algorithm family is very relevant since it is extremely difficult, in some situations, to annotate a huge amount of data [9].

An example of work using this kind of technique is provided by John Rothfels in [10]. In this paper, the author applied unsupervised sentiment analysis techniques to classify sentiments in movie reviews.

To achieve this objective, the authors adapted a method designed by Zagibalov and Carroll to analyze Chinese reviews [11]. The authors drew on this idea to build a similar system to analyze opinions in English reviews.

The idea behind that approach is based on "positive seed words". These words appear in a document in two possible forms: normal ("it's good") or

with a negation ("it's not good"). The first case is the more common of the two. The "seed words" list is first created by hand by the authors and then extended iteratively by the algorithm.

The text is divided into areas corresponding to text passages between punctuation marks. Each of these areas is then classified as positive or negative and the general sentiment of the text is inferred from the predominance of one or the other sentiment in the text.

A second example of unsupervised learning for sentiment analysis is provided by Turney in [12]. In this paper, the author use unsupervised learning to classify, for example, automobile or movies reviews.

To obtain his results, Turney worked on the detection of patterns likely to be used in the expression of an opinion. Its solution relies on the use of part-of-speech tags [6]. The algorithm developed by the author is composed of three steps that we will detail.

- First step: The text is analyzed in a way that we will extract two consecutive words if their part-of-speech tagging follow a set of predefined patterns. These patterns were chosen because they tend to correspond to turns of phrase containing expressions of opinion. One of these patterns is a part of sentence corresponding to the following structure: [Adjective + Name]. If we take the following example sentence "This film was a brilliant work", the passage "brilliant work" will be extracted from the text because it corresponds to this pattern. The complete list of patterns to extract is detailed by Bing Liu in [6].

- Second step: A semantic orientation is calculated for each couple of words extracted in the first step. Turney define the semantic orientation formula by using Pointwise Mutual Information and Information Retrieval Algorithm. Pointwise Mutual Information is a statistical measure of the dependence between two words. The semantic orientation of a sentence corresponds to the difference between PMI(sentence, "poor") and PMI(sentence, "excellent"). These words were chosen arbitrarily because they represent a scale of opinion for a product ranging from 1 star ("poor") to 5 stars ("excellent").

- Third step: For all reviews, the algorithm computes the average semantic orientation of the review and labels it as negative is the average is negative or positive if the average is positive.

By applying this approach to a set of automobile reviews, the author achieves a score of 84% of correct results.

2.5.2 Supervised learning

Unlike unsupervised machine learning, supervised machine learning algorithms work with labeled data. These data are used to train the algorithm to know what output it must provide for a given input.

In the case of sentiment analysis, the data are texts (messages, reviews, etc.) and each of these texts is given a label (negative, positive).

Features

The key to achieve an effective classification is to identify and generate a series of relevant features. What follows is a non-exhaustive list of features used in sentiment analysis [6]:

Terms The most obvious characteristic when it comes to analyzing a text is obviously the terms of a text. There are several ways to consider the terms of a text: one can consider the terms one by one (1-gram) or consider the terms by grouping them by order of appearance in the text (2-grams, 3-grams, ... n-grams).

Terms frequency The term frequency–inverse document frequency (TF-IDF) statistic is a widely used measure in sentiment analysis. It is a calculation of the frequency of appearance of a term in a document weighted by the total frequency of appearance of the term in all the documents studied in order to avoid giving too much importance to terms too banal (e.g.: "a", "an", "the", ...).

Part-of-speech (POS) In sentiment analysis, part-of-speech is very important. Indeed, for example, we realize quite intuitively that an adjective (e.g.: "beautiful", "convenient") is a very strong indicator of the opinion of a text whereas a noun is generally less marked in this direction. Figure [13] shows the list of possible part-of-speech tags.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

FIGURE 2.2: Part-of-speech taggings list. From [13]

Sentiment words and phrases In human language, there are adjectives, adverbs (and nouns, to a lesser extent) that are used to express an opinion (e.g.: "rich", "bad", "happy", "hate", "love", ...). These indicators are very strong clues to determine the opinion of a text. Similarly, there are complete sentences or idioms that express an opinion (e.g. "It's not rocket science", "Miss the boat").

Linear classification

Linear classification is a supervised machine learning technique that consists in determining in which class an object should be placed. Linear classifiers determine this class by trying to construct a linear combination of the object's features. These features are presented in the form of a features vector to be programmatically exploitable [8].

In sentiment analysis, there are two ways to build a linear classifier: Support Vector Machines (SVM) or Neural Networks (NN). We will detail these two techniques.

Support Vector Machines An SVM algorithm is designed to find a linear separation in the search space. This separation is intended to be optimized in the sense that it should generate two separate classes separated optimally [8]. In other words, it is a matter of finding the hyperplane that maximizes the minimum distance between the two generated classes [14].

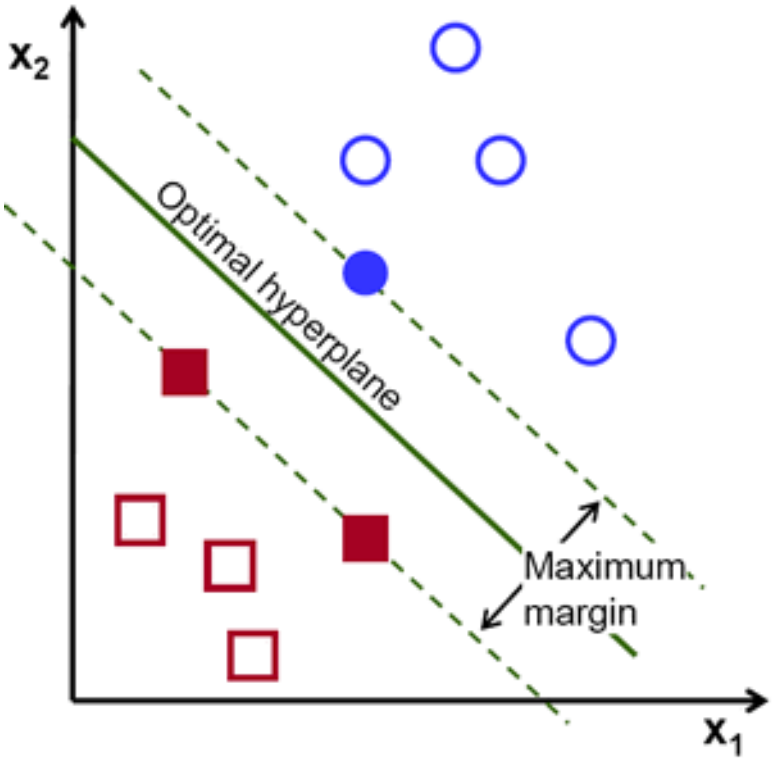

FIGURE 2.3: SVM optimal hyperplane example. From [14]

There are several famous works centered on the use of SVM for sentiment analysis.

In [15], Li and Li propose a framework for analyzing the sentiment polarity of texts coming from micro-blogging platforms using SVM. In this article, authors use data from Twitter and try to provide support tools for decision makers.

In [16], Chen and Tseng propose classifying product reviews according to their quality level to enable decision-makers to make the right decisions by analyzing the most relevant and informative reviews. To achieve this result, the authors explored two approaches based on the use of SVM. This approach has provided excellent results that were more effective than the methods used so far.

In [17], the authors proposed a state-of-the-art SVM to classify messages (SMS or Tweets) at a sentence level. This realization was realized for a competition organized for the Conference on Semantic Evaluation Exercices (SemEval) in 2013. With their implementation of SVM, the authors took first place in the competition with a f1 score of 69.02 for a three-class classification (negative, neutral, positive).

Translated with www.DeepL.com/Translator

Neural Networks Artificial neural networks are loosely based on the structure of biological neural networks in the human brain. This structure is intended to be trained to recognize patterns in the data by adjusting the weights of the input received by the neurons [18].

Given the recent advances and successes brought by neural networks in the field of artificial intelligence [18], it is not surprising that many researchers have tried to apply this technique to sentiment analysis.

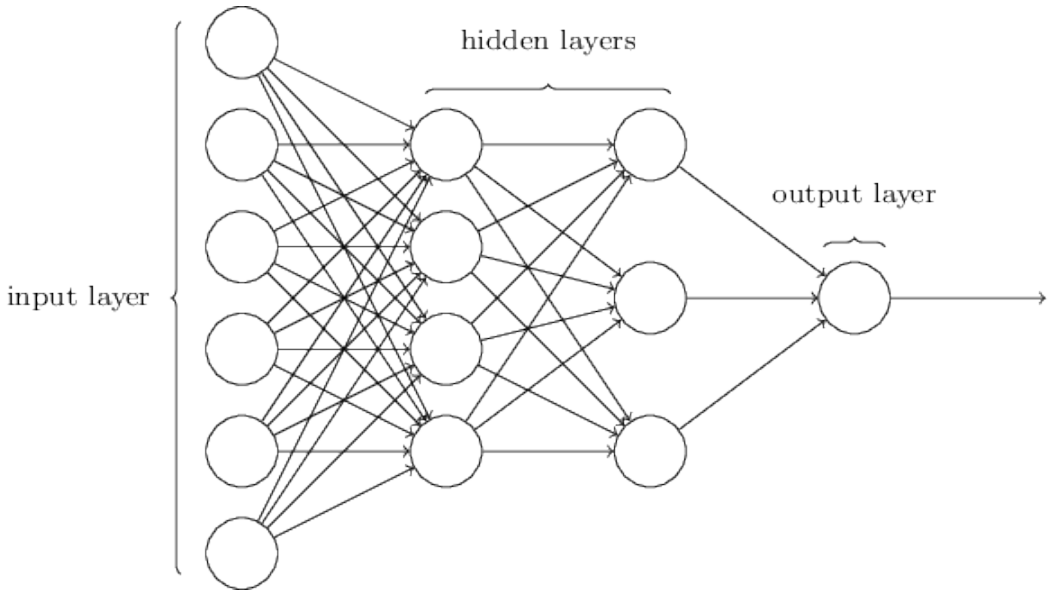

FIGURE 2.4: Example of a NN structure. From [19]

There are different varieties of neural networks. Some are better suited than others to certain tasks and this is also the case when it comes to sentiment analysis.

The two main kind of networks used in sentiment analysis are Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN).

A Recurrent Neural Network is a network with feedback connections that allows the different layers to keep information about previous iterations of data passed through the network: the result of the nth iteration is used to provide the n+1th one. This architecture is particularly adapted to the study of texts because it makes it possible to extract information on the context of a word in a sentence according to the preceding words and does not treat the words as independent entities without link between them.

In [20], the authors used an Recurrent Neural Network to perform text classification. They used a multi-tasking learning framework to teach the network to perform multiple tasks at the same time. The different tasks that the network has been trained to perform are the following: binary classification (negative, positive) at sentence level, binary classification (negative, positive) at document level, binary classification (objective, subjective), classification on 5 classes (from very negative to very positive).

To achieve this objective, the authors proposed three different network architectures:

- The first architecture uses a common layer for all the tasks listed above.
- The second architecture uses different layers for different tasks.
- The third architecture is a mixture of the two previous ones: it assigns a particular layer for each task but they still share a common layer.

This experience provided, for some of the tasks performed, results superior to the current state of the art in sentiment analysis.

A Convolutional Neural Network is another class of Neural Networks build using interleaving layers. This class of networks has been used successfully in image analysis as well as in text analysis [18]. The convolutional layers change the input by applying a convolution operation on it before sending it to the following layer. This technique was copied from the reaction of neurons responsible for responding to visual stimuli.

In [21], the authors build such a Convolutional Neural Network to analyze opinion of Tweets. The architecture used is detailed in figure 2.5. This architecture is divided in four steps.

- First step: The first step in their method involves harvesting and preparing the tweets that will be used to train the model. For each of these Tweets, the authors propose the construction of a "Sentence Matrix" which will contain, in its columns, the different words of the Tweet.
- Second step: The second layer of the network, the convolution layer, aims to find patterns in the data presented.
- Third step: The purpose of the third layer is to aggregate the results of the previous layer in order to reduce the size of its outputs.

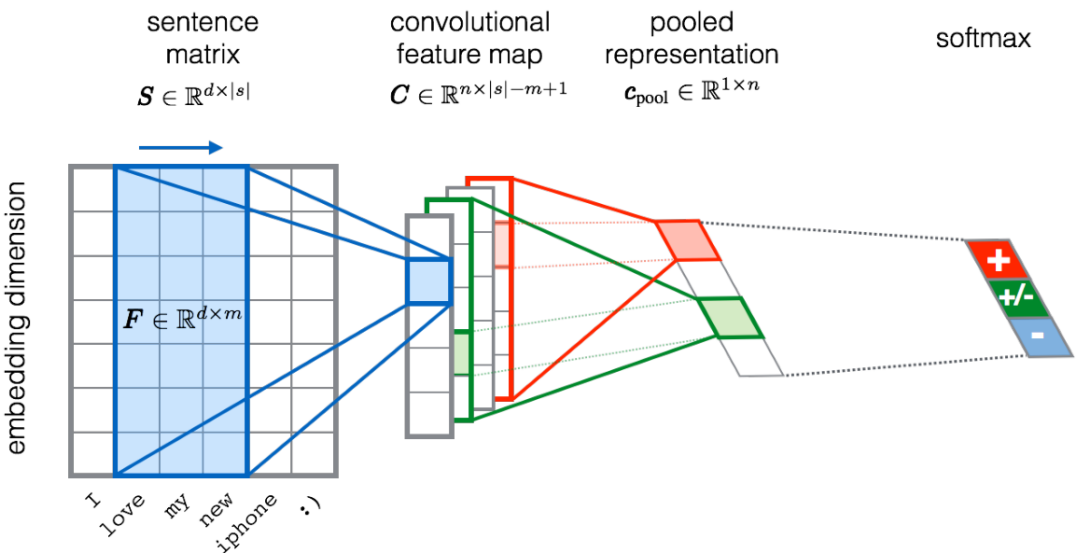

FIGURE 2.5: CNN Sentiment Analysis Architecture. From [21]

- Fourth step: Finally, the fourth and last layer of the network retrieves the results in order to determine, for each Tweet, the probabilities it has of belonging to a sentiment class (e.g.: negative, neutral, positive).

In [22], Yoon Kim uses a Convolutional Neural Network approach to perform classic text classification tasks including sentiment analysis. The architecture used to build the neural network, which is shown in 2.6 is substantially similar to the previous approach.

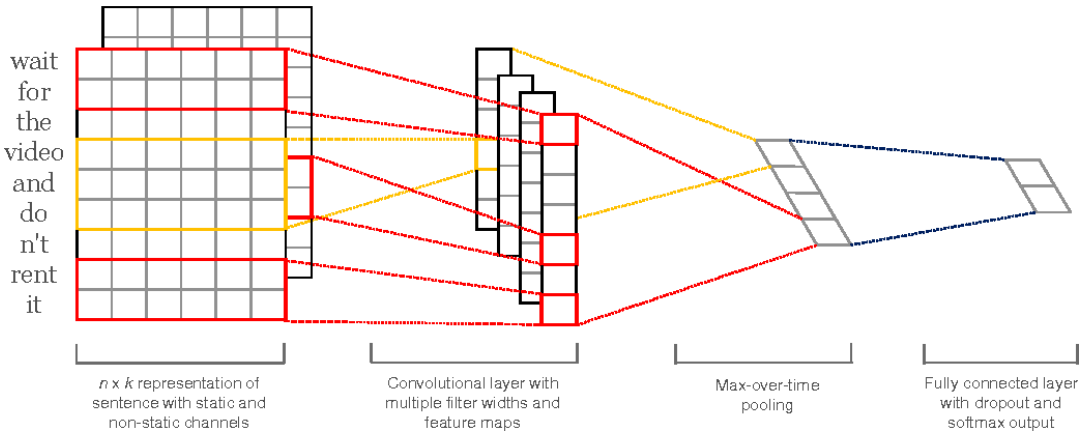

FIGURE 2.6: CNN Sentiment Analysis Architecture. From [22]

Rule-based

Rule-based classifiers are classifiers that use sets of rules to model the space of possibilities. These rules are divided into two parts: a left part and a right part [8].

The left part of the rule corresponds to a pattern of characteristics and the right part corresponds to a class. During a classification, the system will try

to find the rule corresponding to the pattern of the element we are classifying and will deduce the class to which it belongs.

The rules are generated during the training phase on the basis of criteria defined in advance.

Two criteria are often used for this type of task: confidence, which corresponds to the conditional probability of corresponding to the right part if the left part is satisfied, and support, which corresponds to the number of instances in the document whose pattern corresponds to the left side rule.

Decision tree

Decision trees make it possible to construct a hierarchical decomposition of data on the basis of whether or not a condition is met (a condition chosen precisely for its effectiveness in separating data into distinct groups). This separation is repeated until a tree is thin enough to classify the data efficiently [8, 23].

There are several ways to choose how to divide the data. One of these ways is the division based on a single attribute: the group is separated in two, on one side the elements which contain this attribute, on the other side those which do not contain it. A second way to divide the data is to look at multiple attributes. This division is based on frequent word clusters to divide the data. Figure 2.7 show an example of decision tree. It was created to classify irises according to their species.

Decision tree classification algorithms are often based on or inspired by the ID3 algorithm (e.g.: C4.5 and C5 algorithms).

In [25], the authors propose, among other things, a decision tree approach for text classification. To do this, the authors used the C5 algorithm because it offered advantages in terms of accuracy, memory usage and speed compared to the C4.5 algorithm. The approach used was innovative in that it did not use only one decision tree but several and that the final result was decided on the basis of a vote of the different trees.

Probabilistic classifiers

Probabilistic classifiers are classifiers used to calculate the probability for an object to belong to a class.

They differ from previous classifiers because the latter provided, as output, only one class without providing more information. Probabilistic classifiers provide, for each class, a probability that the element belongs to that class.

This additional information allows us to be aware of the strength of the result (if a class is assigned to the object with a very low probability, we can use this information to put into perspective the accuracy of the result).

This type of classifier can be used alone or by coupling the results of different classifiers into ensembles.

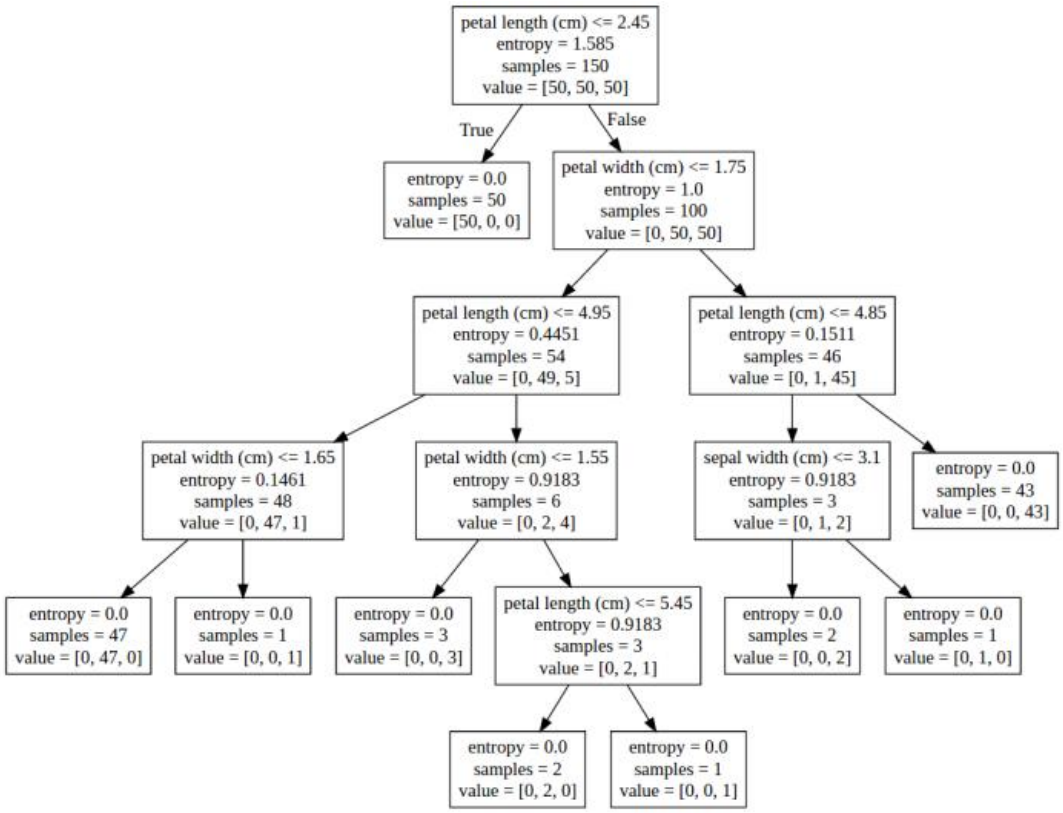

FIGURE 2.7: Example of a decision tree used to classify iris varieties according to their characteristics. From [24]

Naïve Bayes The Naïve Bayes classifier is a very simple and popular classifier. This classifier range is based on Bayes' theorem detailed in Figure 2.8. They work by calculating the probability of belonging to a class based on the distribution of words in the document [8, 26].

These classifiers operate using Bag of Words (BoW). A Bag of Words is a data structure. It is a set that does not take into account the position of words in the document nor duplicate words.

$$P(label|features) = \frac{P(label) * P(f1|label) * .... * P(fn|label)}{P(features)}$$

FIGURE 2.8: Bayes' theorem, rewritten taking into account the hypothesis of independence between variables. From [8]

Bayesian classifiers greatly simplify model creation by assuming that all features are independent. This approximation is often wrong but still provides very good results.

This assumption is one of the biggest advantages of this type of classifier: indeed, the implementation of this type of classifier is very simple and makes learning easy.

Bayesian Network A Bayesian network is a type of Probabilistic Graphical Model that allows to build models based on data [27].

This kind of network can be used for most Machine Learning tasks such as classification, prediction, automated insight, anomaly detection,...

A Bayesian network is probabilistic since it is built on the rules of probability distributions and on the laws of probability to perform the tasks mentioned above.

In a bayesian network, nodes represent features or variables and the links between these nodes represent the dependence of one variable on another variable. However, the absence of a link between two variables does not mean that the two variables are independent since they can be linked to each other via a third variable.

Formally, the dependency relationship between a node A and a node B means that the value of B depends on the value of A. In this situation, node A is considered as the parent of B and node B is the child of A. This dependency logic extends beyond two nodes in the concept of ancestors when one goes up the chain from parent to parent.

The example presented in Figure 2.9 shows the concepts defined above. This model analyses the causes of back pain [28].

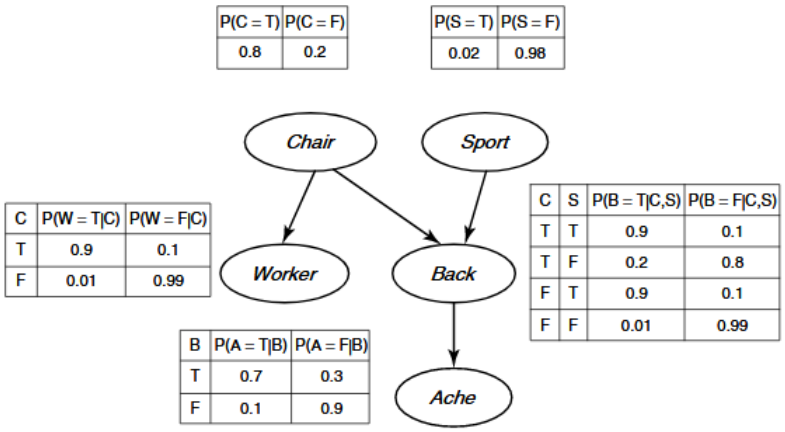

FIGURE 2.9: Bayesian Network example. From [28]

In this example, the couple formed by the variable Back (noted B) and his son, the variable Ache (noted A) indicate back pain in an employee. This pain can result from the -bad- practice of a Sport (noted S) or the use of a Chair (noted C) of poor quality.

This structure makes it possible to suppose that if the chair is of poor quality then the colleagues of work (noted W) of the subject studied also risks to suffer from a pain in the back.

By looking at the tables detailing the dependencies between the nodes, we can notice that S and C are marginally independent but when we add B into the equation, these two variables become dependent.

Network construction is based on the use of a structural learning algorithm that determines the links between nodes in the network.

In contrast to Naïve Bayes classifier which assume that all characteristics are independent [29, 8].

Although interesting, this type of model is rarely used in the field of text analysis because of its cost in calculation which is difficult to associate with a large amount of data.

Maximum Entropy Maximum Entropy Classifier are classifiers based on the Principle of Maximum Entropy [30] and working with feature sets. The Maximum Entropy Classifier selects the one with the largest entropy from all models suitable for the training data [8].

Indeed, the maximum entropy principle says that to find the correct distribution of p(a,b), one must choose the one that maximizes entropy [31].

Formally, is A is the set of possible classes and B is the set of possible contexts, p must satisfy the supplied constraints and maximize the entropy H calculated as follows [32] :

$$H(p) = -\sum_{x \in \varepsilon} p(x) \, log \, p(x)$$

where $x = (a, b)$, $a \in A$, $b \in B$ and $\varepsilon = AxB$

In [33], the author use such a classifier to find matching pairs of sentences in a corpus containing a text in one language and its translation into another language. The advantage of this method lies in the small amount of data needed to drive the model.

2.5.3 Lexicon-based approach

Lexicon-based approach to sentiment analysis is based on the use of a lexicon of terms to which a sentiment or a sentiment score correspond. The overall feeling of a document is then calculated based on the presence or absence of terms in the document and the scores associated with those terms. This approach is divided into two techniques: the corpus approach and the dictionary approach [6].

Corpus approach

The corpus approach is an approach to finding the opinion behind a word in a specific context.

This method relies on the combined use of syntactic patterns and a "seed" list of words defining an opinion to find other words defining an opinion in a corpus and add them to the list.

In [34], the authors developed a method of this type using a starting list containing adjectives marked by an opinion connotation. They then used this list with a series of linguistic constraints to identify new adjectives and add them to the initial list.

Linguistic constraints were based on the presence of key words such as "and, or, but, ...". These keywords allow, in certain cases, to find words having a similar or opposite meaning to a word already known.

Dictionary approach

The dictionary approach is based on the use of a lexical database (dictionary) containing words with an opinion connotation.

This database is built iteratively on the basis of a set of words injected manually. This set does not have to contain many words, 30 words is enough to start the process [35].

The database is then enlarged by browsing the lexical database to retrieve the synonyms and antonyms of the words contained in the initial set iteratively.

In [36], the authors used a dictionary approach based on three different dictionaries to analyze sentiments in Tweets. The use of three dictionaries instead of one allowed the authors to classify Tweets that were usually difficult to classify correctly but on the other hand, the process was slowed down because of the presence of three different dictionaries.

In [35], the authors worked on the classification of customer reviews by breaking down the review according to the characteristics of the product it concerned: in the case of a review concerning a camera, the system recognized the parts of the review that spoke of the screen, the quality of the picture,... and made it possible to provide a summary of the reviews that targeted precise parts of the product.

To achieve this result, the authors identified the adjectives in each review. The polarity of an adjective was deduced from the polarity of its synonyms or from the inverse polarity of its antonyms. To obtain this information, the authors used the WordNet dictionary [37]. This method allowed the authors to obtain correct results with an average of 84%.

2.6 Summary

In this chapter, we have listed and detailed the different methods currently used to perform sentiment analysis on textual content. We have explained the concepts of these methods and, in some cases, the underlying algorithms.

More specifically, we studied the two main families of techniques used, namely machine learning techniques and lexicon based techniques.

In the fist family, we presented the different machine learning approaches that have been developed. We insisted on the simplicity of the Naïve Bayes classifiers and on the recent progress brought by the the Neural Network recent evolutions.

In this second family, we have detailed the two approaches used, namely the dictionary approach and the corpus approach. We presented some situations in which these methods were effective.

This chapter gives an overview of the range of techniques offered to solve this kind of classification problems.

Chapter 3

Methodology

This chapter presents the methodology we adopted to conduct this experiment. It details the research problem and explains the different stages of the experiment from data collection to the final result.

3.1 Problem statement

The goal of this thesis is to design a tool capable of analyzing the polarity of messages posted on Twitter. This objective will be achieved by applying a sentiment analysis approach based on machine learning and more precisely on deep learning. We have chosen this approach based on the state of the art presented in the previous chapter. Indeed, the recent advances made in the field of Deep Learning suggest that this track may offer interesting results.

The classification will be performed at the sentence level and will contain three possible classes: negative, neutral or positive.

Before performing the classification step, it will be necessary to collect data and clean them in order to prepare them for use. This cleaning operation will make it possible to remove superfluous or parasitic information from the processed messages and, incidentally, to reduce the size of the manipulated data.

When the data is cleaned, it will be necessary before training, to find a way to annotate each message with the target class. This label will be used by the neural network during his training phase to adjust its classification.

An approach based on the decomposition of messages into n-grams will be studied. To do this, we will need to build different databases. We will build a base containing the messages in the form of 2-grams, one with 3-grams and finally, one mixing 2-grams and 3-grams. The results of these different approaches will be compared to see if any of them have advantages over the others.

Before the training phase, a feature selection will be applied to the dataset to identify the features that have the greatest impact in determining the class and those that have no value. This step reduces the size of the dataset and therefore the training time of the network.

Once the data is ready, it will be used to build a "training set" that will be used to train the Deep Learning model to perform the classification task.

3.2 Approach

In this section, we present the different steps taken to achieve the objective of sentiment classification. This section covers the entire process from data collection to the final outcome.

Figure 3.1 provides an overview of the application architecture.

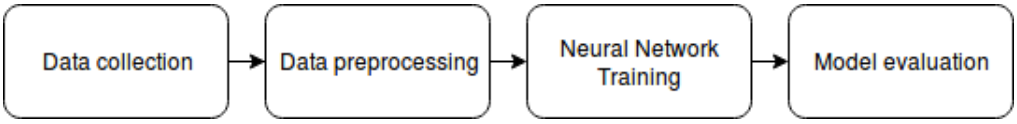

FIGURE 3.1: Application overview

3.2.1 Data collection

The first step in this project is data collection. These data were collected on Twitter using web scrapping techniques. Since we wanted to study in particular how Twitter users felt about Catalonia's independence crisis, we selected the data on the basis of several criteria. These criteria are detailed below.

The first selection criterion was the language of the message. Indeed, to simplify the cleaning, preparation and analysis of data, we have decided to limit ourselves to messages written in English in order to take advantage of all the tools available to perform these tasks. As English is the most widely used language in the field of academic research, there are more tools and resources revolving around this language. In addition, the collection of English messages makes it possible to obtain messages coming from the four corners of the world and thus to build a database of reasonable size.

The second selection criterion was the presence of specific terms in the message. In order to target only messages that address our study topic, we have selected messages that contain the term "catalonia".

The third selection criterion concerned the date of the messages. We have defined a period during which the crisis of independence of Catalonia was a prominent topic and therefore contained many messages on the subject. The period we have selected runs from 1 September 2017 to 31 December 2017.

The database thus constructed contained 522,784 records. Each record contains: the username, the message body, the date and time of the message and the language of the posted message. The language is not used in our case but is an interesting element for future work working on data in several languages.

3.2.2 Data preprocessing

In order to obtain data that could be used by a machine learning algorithm, it was necessary to carry out a series of preliminary steps. These steps were intended to clean the data of some undesirable elements when it comes to analyzing text. These steps also served to reduce the size of the dataset to

make it easier to handle and reduce computation times. Figure 3.2 shows an overview of the different data preprocessing steps.

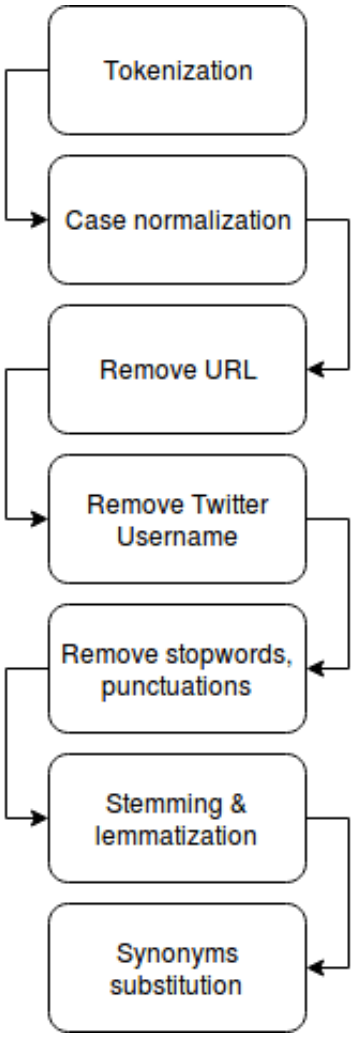

FIGURE 3.2: Preprocessing pipeline

Tokenization

In lexical analysis, a very important step is tokenization. Tokenization is a process used to cut a string of characters into sub-chains, into unit pieces that can be handled individually. These pieces are called tokens. There are several ways to define the tokens of a string and a key question is how this processing will be done. Tokenization is a step that seems trivial. Indeed, we easily imagine that it is enough to cut a sentence using white spaces as delimitation but that sometimes, the tokenizer is confronted with certain special cases that it is necessary to treat correctly.

Based on the following example, we will show some ambiguous situations that a tokenizer must face.

> "Mr. O'Neill thinks that the boys' stories about Chile's capital aren't amusing." [38]

In this example, how should the words "Mr.", "O'Neill", "aren't", "Chile's", "boys" be cut off to keep their meaning? It is important to note that each language has its own set of rules when it comes to tokenization.

In our case, we decided to make a basic tokenization based on white spaces. Some bad tokens that may result from this approach are managed further in the preprocessing stages.

Case normalization

The case normalization step consists, as its name indicates, in reducing all the characters of the document to the same case. This step avoids the presence of duplicates (e.g.: "word" and "Word") in our corpus.

Remove URL

As a microblogging platform and given the short messages, Twitter generates a lot of messages containing links to other websites. These URLs are, in our case, parasitic information that does not contain sentimental connotation and must be removed from messages.

This cleaning is carried out thanks to a regular expression in charge of identifying the various possible patterns for a URL.

Twitter usernames

Again, because of its status as a microblogging platform, we often find, in messages posted on Twitter, references to user names, personalities or groups of people.

Fortunately for us, removing these elements is facilitated by the presence of an"@" symbol (e.g.: @AlanTuring) in front of the elements in question. Again, these elements are identified and removed through a regular expression.

Stopwords and punctuation

In order to keep only the core of each message, punctuation marks are removed from the token list.

Still in this perspective, a set of words that give little weight to lexical analysis are removed from the messages. These words, called "stop words", are generally the most common words in a language. This status ensures that their presence or absence in a document does not bring additional information during the analysis because they are too banal.

There are different types of stop word lists. These lists are used according to the context of lexical analysis to be carried out. In some cases, such as phrase searching [39], it is even counterproductive to use such a list.

Well-known examples of stop words are: "a", "the", "at", ...

Stemming and lemmatization

The stemming and lemmatization steps are designed to find the root of a word.

Stemming is a process whose objective is to find the root of a word that has been conjugated. In the context of a computer stemming, it is not necessary that the word is reduced to a linguistically correct root. It is enough that words having the same root are brought back to the same stem.

In the following example, we can see the stemming result of the derivatives of the verb "to be":

am, are, is => be [38]

If we apply this process to an example sentence, we get the following result :

the boy's cars are different colors => the boy car be differ color [38]

For the human eye, this treatment seems to make the sentence lose meaning, but the point here is to reduce all the words in the document to a smaller corpus that the program will be better able to treat. Moreover, the fact of bringing a word back to its root does not take away its sentimental connotation.

There are many ways to achieve this result. In this project, we used lemmatization. Lemmatization is a two-step approach.

The first step is the application of part-of-speech tags on the different words of the sentence.

The second step consists, on the basis of the word and its POS tag, of identifying the stemming rule which corresponds best to it and of applying it in order to recover the root of it.

The logic behind this approach is based on the fact that it is easier to return a word to its stem if we know better the nature of the word: identifying that the word to be processed is a verb allow to easily identify the rule to apply to it to return it to its stem.

On the other hand, this additional step is a source of errors if it is not done correctly: if a word is classified in a category that does not correspond to it, there is a good chance that its lemmatization will contains errors.

Synonyms substitution

The last step in this pre-processing process is to return all words to their most common synonym. This step is done using a lexical database called WordNet [37].

WordNet is a lexical database of the English language. This database groups nouns, verbs and adverbs into groups of cognitive synonyms that express a particular concept. Overall, WordNet groups words with a similar meaning together as a navigable network.

This tool allows, among other things, to identify the synonyms of a word. In this project, we decided to bring all the words in our corpus back to the

first synonym in the list of synonyms for that word. This treatment considerably reduces the number of different words in our corpus. The following example shows how synonyms have been replaced in the project.

car, automobile, jeep, van => car

3.2.3 Sentiment labeling

Being in a supervised learning machine context, we need data with a label. Since it is not possible to annotate the 522,784 messages by hand, we decided to use a tool to automatically annotate the texts. To accomplish this task, we used the Stanford CoreNLP [40].

The Stanford CoreNLP is a toolkit providing a set of instruments for performing natural language processing tasks. The module for sentiment analysis allowed us to create labels for each message to allow our neural network to learn to classify based on these examples.

3.2.4 Dataset building

To feed the neural network, we need to build a dataset respecting a precise structure. Our dataset must be in the form of a matrix. Technically speaking, this format is a CSV file.

In order to build this file, we must first build the set of all the words (or group of words) present in our corpus. Once this step is completed, we can begin to build the matrix.

Each line (i) in the matrix represents a Twitter message. Each column (j) of this matrix represents a word (or group of words) from our corpus. Intersections between rows and columns can contain two values: 0 or 1.

If the element located in M_{ij} is 0, it means that the message located at i-th line does not contain the word contained at j-th column. If it is a 1, then the word is included in the message.

The last column of the matrix contains the targets to be predicted. These targets are encoded as numbers ranging from 0 (the message has been labelled "negative") to 2 (the message has been labelled "positive").

N-grams approach

In natural language processing, it is common to use character, syllable or word groups instead of limiting the analysis to only one of these elements at a time. This technique allows the elements to be analyzed together and not separately and to provides more information on the context of the analysis.

This type of grouping is called a n-grams [41]. In the case of a grouping by words, we also speak of shingles. In the case of a grouping of all possible consecutive word pairs, we speak of 2-grams (or bigrams). In the case of a grouping by three words, it is a 3-grams (or trigrams).

As part of this project, we conducted several different types of groupings to compare results. We have realized three different approaches, namely a 2-grams dataset, a 3-grams dataset and a 2-grams and 3-grams dataset.

Feature selection

After having built the dataset, it is advisable to reduce the relatively important size of these data by identifying, among all the features of this dataset, those which are important in the determination of the sentimental label of a message. This step responds to a technical constraint. Indeed, as an example, once built, a 1-gram dataset on our data, is around 40GB. The size increases further when building a 2-gram or 3-gram dataset. This mass of data would require a very long time during the training phase.

To solve this problem, we used a feature selection technique based on Mutual Information: Symmetrical Uncertainty. This technique is a measure of dependence between two distinct variables. It is used to remove messages from the dataset that have no (or very little) influence on the final result.

Symmetrical Uncertainty is defined as [42] :

$$SU(X,Y) = 2 * \frac{I(X;Y)}{(H(X) + H(Y))} \tag{3.1}$$

where I(X;Y) is the mutual information defined as :

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\, p(y)} \right) \tag{3.2}$$

where $p(x,y)$ is the joint probability function of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of X and Y respectively.

In our case, we will calculate the SI for each column (therefore each feature). The variable X will be the vector composed of all 0 and 1 designating the presence of this token in a message and Y will be the sentiment column. The objective of the operation is to identify features that are not related to the feeling column.

3.2.5 Dataset structure

Following the previous steps, our dataset is ready to be manipulated by the neural network.

The dataset obtained is a M-by-N+1 matrix. The number in the ijth cell indicates the presence or absence of token j in the message i. The last column represents the sentiment label assigned to the message in the line.

Table 3.1 gives an overview of the dataset format.

	N-gram1	N-gram2	N-gram3	...	N-gramN	Sentiment
Message1	1	1	1		0	0
Message2	0	0	0		1	2
Message3	1	0	0		0	1
...						
MessageM	0	0	1		0	1

TABLE 3.1: Dataset structure

3.2.6 Dataset characteristics

Table 3.2 presents the characteristics of the data we collected for this project.

Capture dates	01/09/2017 to 31/12/2017
Number of tweets	522,784
Number of negative tweets	440,598
Number of neutral tweets	55,789
Number of positive tweets	26,397

TABLE 3.2: Dataset characteristics

3.3 Neural Network Training

This section presents the architecture and characteristics of the neural network we trained to try to determine the opinion of a message.

3.3.1 Network Parameters

As is often the case with Machine Learning, Neural Networks have a number of hyperparameters [43] that adjust their structure and how the network will learn.

These hyperparameters are directly responsible for the results obtained by the final model and it is therefore essential to select and adjust them rigorously.

This section presents the different hyperparameters used in our network.

Number of inputs

The number of inputs defines the number of neurons present in the very first layer of the network. In our project, the number of neurons in the first layer is defined by the width of the dataset matrix we are using. Indeed, the dataset generated with 2-grams has not the same width as the dataset generated with 3-grams.

Each neuron in the first layer receives a single input number that represents the absence or presence of a certain token in the message being passed through the network.

Hidden Layers number

In a neural network, the number of hidden layers corresponds to the number of layers of the network without the input layer or the output layer.

Activation function [44]

A neural network is composed of neurons. Each of these neurons calculates a weighted sum of the values it received in input (with the addition of a bias).

Depending on the result obtained, the neuron decides whether it is activated or not.

The problem with this calculation is that we do not know the limits in which its result will appear and therefore, it is difficult to define if the neuron should activate or not. In order to solve this problem, it is necessary to introduce after the previous calculation, an activation function which will allow to normalize the result between certain known limits and to decide if the neuron should be activated or not.

This function is called an activation function. There are several that have their advantages and disadvantages. When it comes to classification tasks, sigmoid functions are preferred because they are more efficient [45].

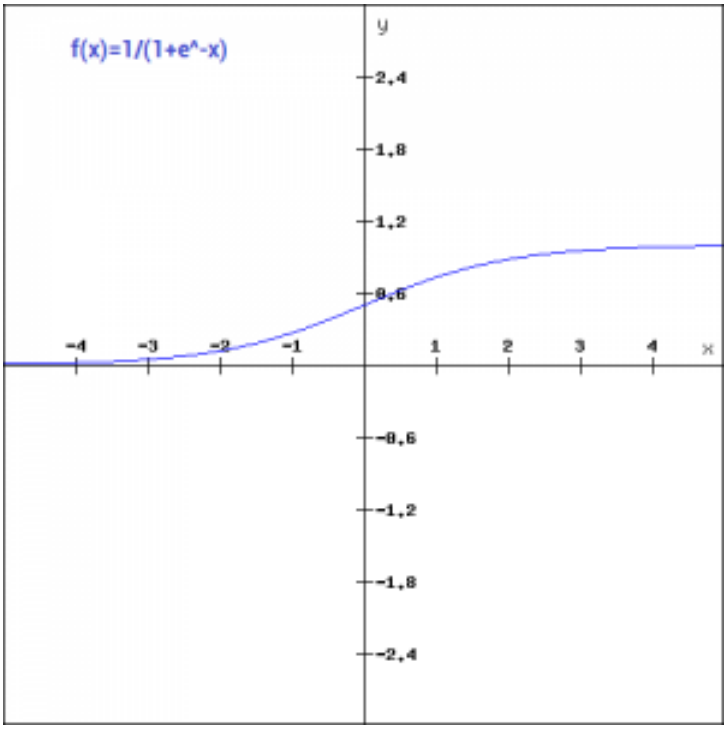

FIGURE 3.3: Sigmoid activation function. From [45]

Learning rate

A neural network aims to iteratively advance towards a model that will provide the smallest possible error. This iterative learning phase is called the "Gradient descent". The Gradient Descent is an iterative optimization approach aimed at finding the smallest value in the error curve.

In order to achieve this result, a descending gradient must proceed in small steps. These small steps are a parameter of the descending gradient and are called learning rate.

The learning rate determines how fast the network will converge to the target value. If the rate is too small, the network will converge too slowly towards its target but if it is too large, the network may diverge and move away from the target value.

Figure 3.4 shows how the learning rate influences the learning process. In particular, we note how the approach may diverge if the learning rate is too high.

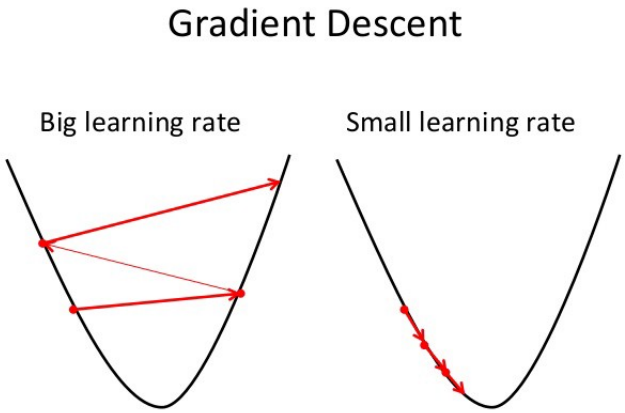

FIGURE 3.4: Exemple of small and big learning rate

Number of epochs

An epoch represents the number of times the entire dataset has passed through the neural network to adjust it [46].

Since the neural network is not iterative learning, it is important that it is confronted several times with the entire dataset.

If the network is only confronted once with the dataset, it risks being in an underfitting situation (the network is under-trained and does not function correctly).

As the number of epochs increases, the weights of the neurons in the network are adjusted and the network becomes more efficient.

However, if the network is confronted too many times with the dataset, it can find itself in an overfitting situation (the network is too trained on a particular dataset and can no longer correspond to other data).

Stopping criteria

At each epoch, the different weights of the network are adjusted. In each new state, the network is tested to see if it delivers good results. Normally, each epoch should improve the accuracy of the results obtained, but it is possible that the network reaches a saturation point and that a new epoch brings nothing (or almost nothing) to the current results.

In this situation, there is no point in continuing to train the network over and over again. To avoid this problem, we define a shutdown condition for our network. As long as the stop condition is not respected, the network continues its training but if the condition is met then the training stops.

Typically, training stop is triggered when the difference between the error rate of an iteration and the previous iteration is less than the stop criterion.

3.3.2 Hyperparameters of our network

Inputs number	Token number
Hidden layers numbers	2
Activation function	Sigmoid
Learning rate	0.1
Number of epochs	1000
Stopping criteria	error ≤ 0.1

TABLE 3.3: Network hyperparameters

3.4 Summary

In this chapter, we have presented the methodology we have developed to conduct our different experiments.

First, we presented the problematic that led us to conduct these experiments. Next, we presented an overview of the experience. This view presented the different stages of the experiment, namely: data collection, data preparation, model training and finally the evaluation of model results. Throughout the chapter, each of these steps has been presented in more detail to provide the reader with precise explanations of the treatments performed.

These explanations help us understand the dataset structure we built and the architecture of the neural network we used.

Chapter 4

Results

In this chapter, we discuss the results of our experiment.

The first part of the chapter lists the different evaluation criteria we selected and explains what they measure.

The second part of the chapter uses these criteria to evaluate the three approaches used for this experiment.

4.1 Evaluation criteria

In order to evaluate the quality of our model, it is necessary to select criteria that will allow us to quantify its results [47]. In this perspective, we have selected four measures which are: accuracy, recall, precision and F1 score. These four measures are frequently used in the fields of pattern recognition, information retrieval and binary classification.

4.1.1 Positive and negative results

Before talking about the criteria, it is important to introduce some concepts that will be used later. These notions are part of the vocabulary used to designate a classification [47].

True positive

When evaluating the results of a classification, a true positive rate measures the number of instances that have been classified in a category and actually belong to that category. We will write it TP.

True negative

The true negative rate describes the number of instances that have not been classified in a class and that effectively do not belong to that class. We will write it TN.

False positive

The false positive rate indicates the number of instances that have been classified in a class but actually do not belong to that class. We will write it FP.

False negative

The false negative rate indicates the number of instances that have not been classified in a class but that do belong to that class. We will write it FN.

4.1.2 Confusion Matrix

In supervised automatic learning, a confusion matrix is a result that quantifies the quality of a classification system. The idea behind a confusion matrix is to count the number of times an instance of a particular class is wrongly classified (that instance is classified in another class). By counting the number of misclassification for each class, we can build a matrix to get an overview of model classification errors [48].

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

FIGURE 4.1: An example of confusion matrix. From [48]

Figure 4.1 gives an example of a confusion matrix. In this matrix, we can see that the model has classified 165 instances. In these data, 60 instances were to be classified as "No" and 105 as "Yes". By observing the results of the model, we count 55 "No" and 110 "Yes".

Figure 4.2 shows the link between the confusion matrix and the concepts presented above.

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

FIGURE 4.2: Confusion matrix and classification concepts. From [48]

4.1.3 Accuracy

Accuracy is the simplest metric used in classification. It simply calculates the number of correctly classified instances []. Although intuitive, accuracy is not a sufficient measure to determine the quality of the classifier. Indeed, take for example a situation where our dataset contains 95% of its instances belonging to one class and 5% to the other class. Even if the binary classifier classes 100% of instances in the first class, it would have an accuracy of 95%.

$$accuracy = \frac{TP + TN}{TotalPopulation}$$

4.1.4 Precision

Precision is a metric used to quantify the accuracy of positive predictions. It is defined by the ratio of the number of true positives divided by the number of true positives plus the number of false positives [47]. Precision is an important metric when we are trying to create a model in which the FP rate is very important. An example of this type of model is Spam detection. In this situation, classifying a legitimate email in the Spam folder can have important consequences for the user.

$$precision = \frac{TP}{TP + FP}$$

4.1.5 Recall

Recall is a metric used to quantify the ratio of positive instances that are correctly classify. It is defined by the ratio of the number of true positives divided by the number of true positives plus the number of false negatives [47]. Recall is an important metric when we are trying to create a model in which the FN rate is very important. In some cases, such as detecting a health problem or identifying a serious threat, a NF has an extremely high cost and must be avoided at all costs.

$$recall = \frac{TP}{TP + FN}$$

4.1.6 F1 Score

There is a way to combine precision and recall in a single metric. This metric is called F1 Score. The F1 Score is the harmonic mean of precision and recall. The harmonic mean differs from a classical mean because it gives more weight to the low results and therefore, the F1 Score will be high only if the recall and the precision are both high [47].

$$F1Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2 * \frac{precision * recall}{precision + recall} = \frac{TP}{TP + \frac{FN+FP}{2}}$$

4.2 Results analysis

This section presents the results obtained with the three approaches we used for this experiment. These three approaches are: 2-grams, 3-grams and 2-3-grams. The results are presented in tabular form containing the results measured with the metrics previously presented for the three approaches.

4.2.1 Results of the three approachs

Accuracy	Precision	Recall	F1 Score
0.671	0.399	0.431	0.398

TABLE 4.1: 2-grams approach results

Accuracy	Precision	Recall	F1 Score
0.726	0.413	0.445	0.423

TABLE 4.2: 3-grams approach results

Accuracy	Precision	Recall	F1 Score
0.657	0.404	0.431	0.404

TABLE 4.3: 2-3-grams approach results

4.3 Comparison of results

In this section, we present the results obtained by the three approaches for each metric. This presentation makes it easier to compare the results of each metric.

4.3.1 Accuracy comparison

Figure 4.3 shows the accuracy measured for the three approaches. The 3-grams approach offers the best accuracy with an increase of 0.055 compared to the 2-grams approach which offers the second best result for this metric. The 2 & 3 grams approach offers a lower score than the two simple approaches.

4.3.2 Precision comparison

Figure 4.4 shows the precision measured for the three approaches. Again, it is the 3-grams approach that offers the best result for this metric. It is ahead of the 2-3-grams approach, the second highest score, by 0.009. The 2-grams approach is last with an precision of 0.399.

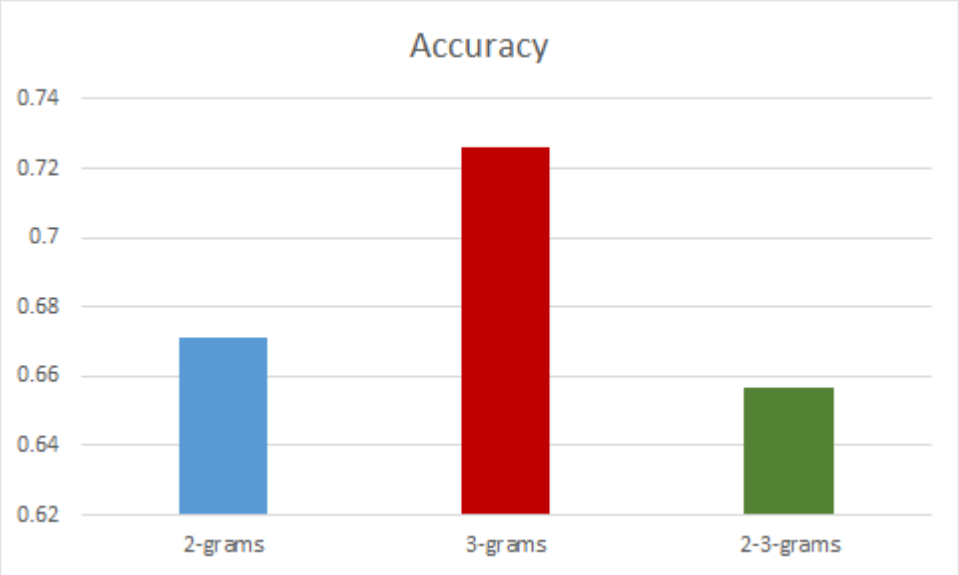

FIGURE 4.3: Comparison of the accuracy of the three approaches.

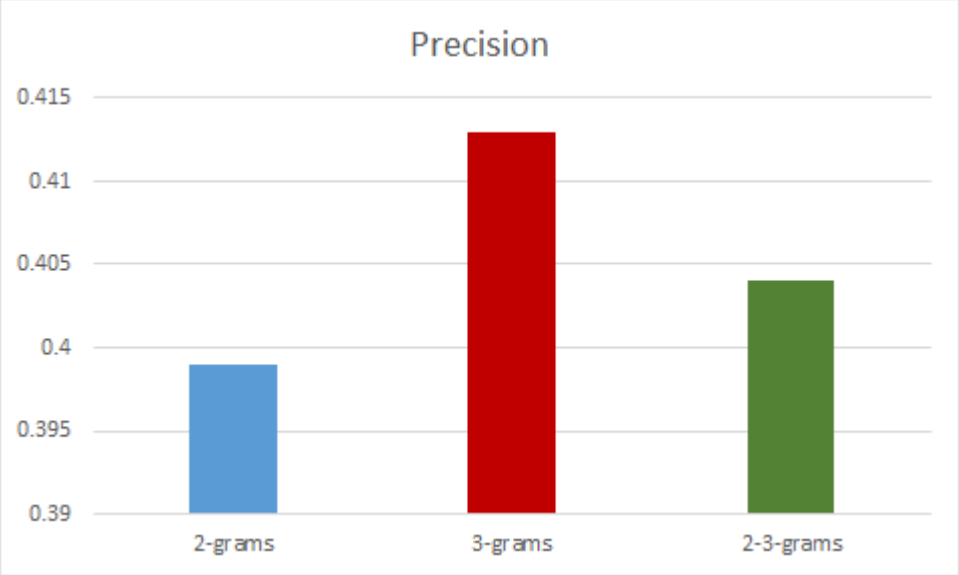

FIGURE 4.4: Comparison of the precision of the three approaches.

4.3.3 Recall comparison

Figure 4.5 shows the recall measured for the three approaches. Once again, the 3-grams approach gives the best results. With its recall score of 0.445, it leads the other two approaches (which are tied at 0.431) from 0.014.

4.3.4 F1 Score comparison

Figure 4.6 shows the f1 score measured for the three approaches. Logically given the results presented above, the 3-grams approach offers the best f1

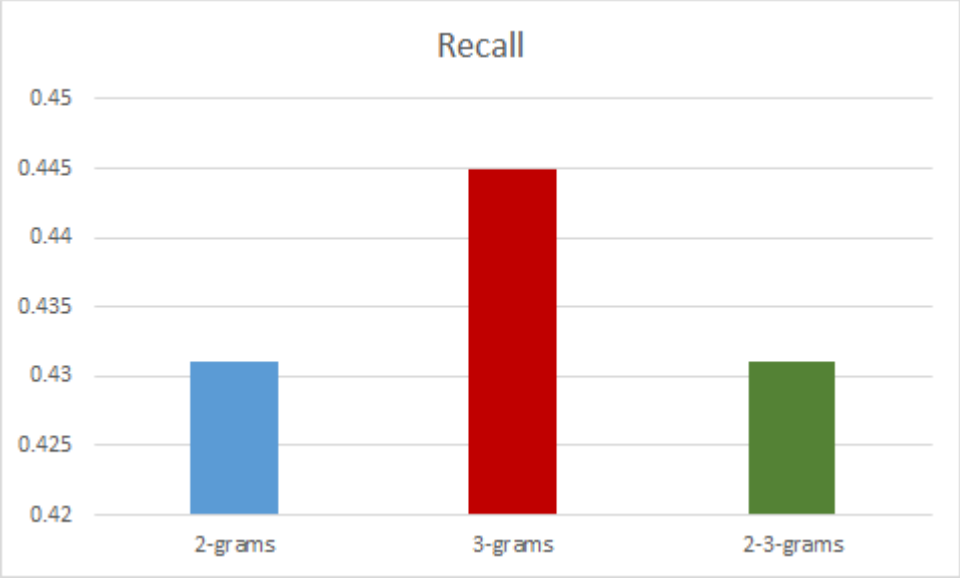

FIGURE 4.5: Comparison of the recall of the three approaches.

score with a score of 0.423. This approach is 0.019 higher than the 2-3-grams approach and 0.025 higher than the 2-grams approach.

The f1 score is the metric most likely to provide a global overview of model quality. Indeed, this metric takes into account both the precision and the recall of the model studied and thus provides a balanced means to evaluate the best model.

In our case, it seems that the 3-grams approach offers the best overall results.

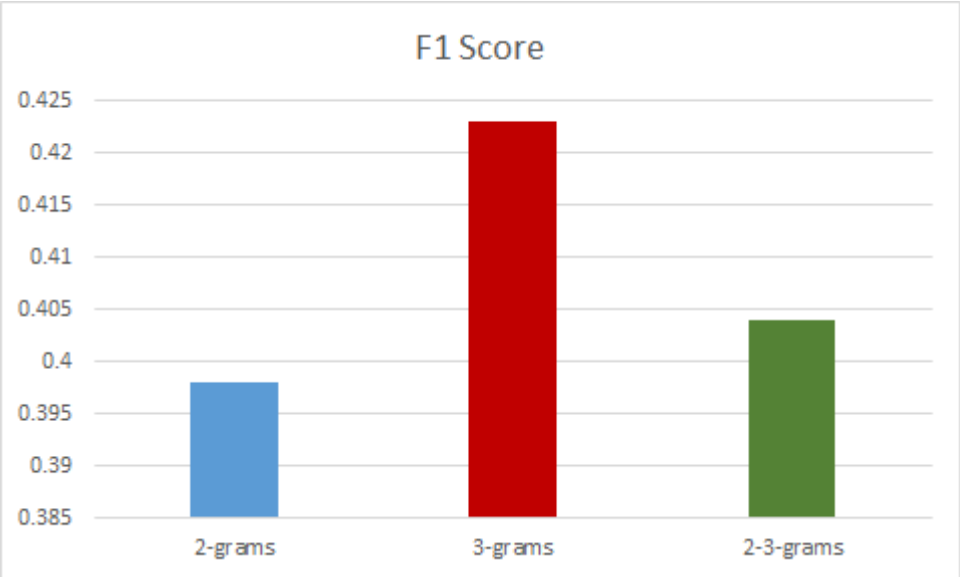

FIGURE 4.6: Comparison of the f1 score of the three approaches.

4.4 Summary

In this chapter, we have presented the results obtained during our three experiments. We trained a classification model based on three different datasets. The datasets used contained n-grams built on Tweets. The first experiment consisted in training the model with a 2-grams dataset. The second experiment consisted in training the model with a 3-grams dataset and finally, the last experiment consisted in a hybrid approach that contained both 2-grams and 3-grams.

We have found that the 3-grams approach provides the best results on all the metrics used. The hybrid approach does not seem to provide interesting results.

Overall, the results obtained are quite poor and do not seem to be able to provide an effective tool to perform sentiment analysis on Twitter messages.

The results seem weak when compared to the current state of the art presented in Chapter 2. Indeed, the authors of [17] obtain a f1 score of 69.02 for a task identical to that carried out for our experiments using a Support Vector Machines approach to the problem. The following chapter provides some suggestions for improving these results.

Chapter 5

Conclusion

This chapter concludes the work we have done and presents opportunities for improvement.

5.1 Conclusion

In this work, we explored the scientific discipline seeking to automatically analyze the sentiment or opinion behind a message written in natural language: sentiment analysis. This discipline has been in effervescence since the 2000s and the emergence of social media and e-commerce platforms that generate a gigantic number of subjective messages. The sentiment analysis research community has since been working on techniques and technologies to analyze and value this mass of information.

This work presented, in Chapter 2, the different approaches used to perform sentiment analysis. We have studied "lexical based" and "machine learning based" approaches. After presenting a whole series of works using different approaches, we decided to concentrate on a machine learning technique whose effectiveness has been revealed on a large scale in recent years: neural networks.

In the third chapter, we presented the approach we decided to adopt to carry out our experiments. So, we explained the objective we set ourselves: the classification of Tweets into three sentiment classes by relying on a dataset containing consecutive word groupings (2-grams, 3-grams and 2-3-grams). We detailed all the operations that brought us to the end of the experiment: data collection and preparation, dataset construction, neural network training and results evaluation. In this chapter, we have also presented the parameters that define the architecture of a neural network and we have stated the parameters that make up ours.

After training and evaluating the neural network, we studied, in chapter 4, the different metrics commonly used to analyze the results of a classification task like ours, namely accuracy, precision, recall and f1 score. For each of the three approaches we used these metrics and compared their results. Based on these results, we were able to conclude that our approach offered low quality classifiers. Indeed, the best f1 score obtained with one of our approach was 0.423 while the current state of the art obtains much higher scores (0.692 for a similar task in [17]). In the same chapter, we have presented some

reflections on the quality of the results and the possibilities to be explored to improve them.

5.2 Future work

Following this conclusion, we can think about ways to improve this score in the future.

First, we could imagine using a larger dataset. Indeed, even if the size of the dataset is not the only element that is taken into account in obtaining an efficient model, it is an important parameter that allows the model to be fine-tuned. Our dataset consists of about 522,784 Tweets and could be expanded.

Second, the dataset in question is very unbalanced (there are many more negative than positive messages). This observation is quite logical given that we built the dataset around a particular event (the Catalan independence crisis) and that this event generated many negative reactions on Twitter. An imbalance in a dataset helps create bad results if it is not managed.

Third, we have trained our neural network with a set of hyperparameters but there are many others. One way to improve the results obtained would be to set up a selection of hyperparameters using a random search or a grid search [49] and to compare the results of the different models generated to select the best.

Bibliography

[1] *Internet World Stats : Internet usage statistics*. http://www.internetworldstats.com/stats.htm. Accessed: 18-01-2018.

[2] Danah M. Boyd and Nicole B. Ellison. "Social network sites: Definition, history, and scholarship". In: *Journal of Computer-Mediated Communication* 13.1 (2007). URL: http://jcmc.indiana.edu/vol13/issue1/boyd.ellison.html.

[3] *Number of monthly active Twitter users*. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users. Accessed: 18-01-2018.

[4] *Twitter Usage Statistics*. http://www.internetlivestats.com/twitter-statistics. Accessed: 18-01-2018.

[5] Mika Viking Mäntylä, Daniel Graziotin, and Miikka Kuutila. "The Evolution of Sentiment Analysis - A Review of Research Topics, Venues, and Top Cited Papers". In: *CoRR* abs/1612.01556 (2016). arXiv: 1612.01556. URL: http://arxiv.org/abs/1612.01556.

[6] Bing Liu. *Sentiment Analysis and Opinion Mining*. 2012.

[7] *Spanish stocks drop sharply after Catalonia declares independence*. http://uk.businessinsider.com/spanish-stocks-drop-catalonia-declares-independence-2017-10. Accessed: 18-01-2018.

[8] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey". In: (2014).

[9] Mohssen Mohammed, Muhammad Badruddin Khan, and Eihab Bashier. *Machine Learning: Algorithms and Applications*. July 2016. ISBN: 9781498705387.

[10] John Rothfels and Julie Tibshirani. "Unsupervised sentiment classification of English movie reviews using automatic selection of positive and negative sentiment items". In: (2010).

[11] Taras Zagibalov and John Carroll. "Automatic Seed Word Selection for Unsupervised Sentiment Classification of Chinese Text". In: COLING '08 (2008), pp. 1073–1080. URL: http://dl.acm.org/citation.cfm?id=1599081.1599216.

[12] Peter D. Turney. "Thumbs Up or Thumbs Down?: Semantic Orientation Applied to Unsupervised Classification of Reviews". In: ACL '02 (2002), pp. 417–424. DOI: 10.3115/1073083.1073153. URL: http://dx.doi.org/10.3115/1073083.1073153.

[13] *Part-of-speech tagging using OpenNLP*. http://blog.thedigitalgroup.com. Accessed: 02-04-2018.

[14] *Introduction to Support Vector Machines*. https://docs.opencv.org. Accessed: 02-04-2018.

[15] Yung-Ming Li and Tsung-Ying Li. "Deriving Market Intelligence from Microblogs". In: *Decis. Support Syst.* 55.1 (Apr. 2013), pp. 206–217. ISSN: 0167-9236. DOI: 10.1016/j.dss.2013.01.023. URL: http://dx.doi.org/10.1016/j.dss.2013.01.023.

[16] Chien Chin Chen and You-De Tseng. "Quality Evaluation of Product Reviews Using an Information Quality Framework". In: *Decis. Support Syst.* 50.4 (Mar. 2011), pp. 755–768. ISSN: 0167-9236. DOI: 10.1016/j.dss.2010.08.023. URL: http://dx.doi.org/10.1016/j.dss.2010.08.023.

[17] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets". In: *CoRR* abs/1308.6242 (2013). arXiv: 1308.6242. URL: http://arxiv.org/abs/1308.6242.

[18] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[19] *Neural Networks And Deep Learning*. http://neuralnetworksanddeeplearning.com. Accessed: 02-04-2018.

[20] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. "Recurrent Neural Network for Text Classification with Multi-Task Learning". In: *CoRR* abs/1605.05101 (2016). arXiv: 1605.05101. URL: http://arxiv.org/abs/1605.05101.

[21] Aliaksei Severyn and Alessandro Moschitti. "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification". In: (2015).

[22] Yoon Kim. "Convolutional Neural Networks for Sentence Classification". In: *CoRR* abs/1408.5882 (2014). arXiv: 1408.5882. URL: http://arxiv.org/abs/1408.5882.

[23] Charu C. Aggarwal and ChengXiang Zhai. "A Survey of Text Classification Algorithms." In: (2012). Ed. by Charu C. Aggarwal and ChengXiang Zhai, pp. 163–222. URL: http://dblp.uni-trier.de/db/books/collections/Mining2012.html#AggarwalZ12b.

[24] *Simplifying Decision Tree*. https://www.kdnuggets.com. Accessed: 02/04/2018.

[25] Y. H. Li and A. K. Jain. "Classification of Text Documents". In: *The Computer Journal* 41.8 (1998), pp. 537–546. DOI: 10.1093/comjnl/41.8.537. eprint: /oup/backfile/content_public/journal/comjnl/41/8/10.1093/comjnl/41.8.537/2/410537.pdf. URL: http://dx.doi.org/10.1093/comjnl/41.8.537.

[26] Irina Rish. "An empirical study of the naive Bayes classifier". In: 3.22 (2001), pp. 41–46.

[27] *Bayesian networks - an introduction*. https://www.bayesserver.com/docs/introduction/bayesian-networks. Accessed: 29/04/2018.

[28] *Bayesian Networks*. http://www.eng.tau.ac.il/~bengal/BN.pdf. Accessed: 29/04/2018.

[29] Nir Friedman, Dan Geiger, and Moises Goldszmidt. "Bayesian Network Classifiers". In: *Mach. Learn.* 29.2-3 (Nov. 1997), pp. 131–163. ISSN: 0885-6125. DOI: 10.1023/A:1007465528199. URL: https://doi.org/10.1023/A:1007465528199.

[30] *Maximum Entropy Principle*. https://en.wikipedia.org/wiki/Principle_of_maximum_entropy. Accessed: 06/04/2018.

[31] *Sentiment identification using Maximum Entropy Analysis of Movie Reviews*. https://web.stanford.edu/class/cs276a/projects/reports/nmehra-kshashi-priyank9.pdf. Accessed: 29/04/2018.

[32] *A simple introduction to Maximum Entropy Models for Natural Language Processing*. https://repository.upenn.edu/cgi/viewcontent.cgi?article=1083&context=ircs_reports. Accessed: 29/04/2018.

[33] Kaufmann JM. "A Maximum Entropy Sentence Alignment Tool." In: ().

[34] Vasileios Hatzivassiloglou and Kathleen R. Mckeown. "Predicting the Semantic Orientation of Adjectives". In: (May 2002).

[35] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews". In: KDD '04 (2004), pp. 168–177. DOI: 10.1145/1014052.1014073. URL: http://doi.acm.org/10.1145/1014052.1014073.

[36] Seongik Park and Yanggon Kim. "Building thesaurus lexicon using dictionary-based approach for sentiment classification". In: (June 2016), pp. 39–44.

[37] *WordNet Dictionnary*. https://wordnet.princeton.edu/. Accessed: 05/04/2018.

[38] *Stanford NLP*. https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html. Accessed: 13-04-2018.

[39] *Phrase Searching*. https://www.library.ohio.edu/research/the-research-process/basic-search-techniques/phrase-searching/. Accessed: 05/05/2018.

[40] *Stanford CoreNLP – Natural language software*. https://stanfordnlp.github.io/CoreNLP/index.html. Accessed: 14/04/2018.

[41] Payal B. Awachate and Vivek P. Kshirsagar. "Improved Twitter Sentiment Analysis Using N Gram Feature Selection and Combinations". In: *IJARCCE* 5 (2016). URL: https://www.ijarcce.com/upload/2016/september-16/IJARCCE%2035.pdf.

[42] S. I. Ali and W. Shahzad. "A feature subset selection method based on symmetric uncertainty and Ant Colony Optimization". In: (2012), pp. 1–6. DOI: 10.1109/ICET.2012.6375420.

[43] *What are Hyperparameters ? and How to tune the Hyperparameters in a Deep Neural Network?* https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a. Accessed: 24/04/2018.

[44] *Understanding Activation Functions in Neural Networks*. https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0. Accessed: 24/04/2018.

[45] *Fundamentals of Deep Learning – Activation Functions and When to Use Them?* https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/. Accessed: 29/04/2018.

[46] *Epoch vs Batch Size vs Iterations*. https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9. Accessed: 24/04/2018.

[47] Aurlien Geron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 1st. O'Reilly Media, Inc., 2017. ISBN: 1491962291, 9781491962299.

[48] *Confusion Matrix*. http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/. Accessed: 06/05/2018.

[49] *Arbiter, a tool dedicated to tuning (hyperparameter optimization) of machine learning models.* https://github.com/deeplearning4j/Arbiter. Accessed: 19/05/2018.