UNIVERSITÉ DU QUÉBEC À MONTRÉAL

SENTIMENT ANALYSIS OF CULTURAL PRODUCT REVIEWS

DISSERTATION

PRESENTED

AS PARTIAL REQUIREMENT

TO THE MASTERS IN COMPUTER SCIENCE

BY

MAHTAB ABBASIGARAVAND

SEPTEMBER 2020

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

ANALYSE DE SENTIMENTS DANS LES REVUES DE PRODUITS CULTURELS

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

MAHTAB ABBASIGARAVAND

SEPTEMBRE 2020

First and foremost, I would like to express my special thanks to my adviser, Professor Hafedh Mili, for the continuous support, his patience and motivation.

There are no proper words to convey my deep gratitude and respect for my parents, Soraya and Sohrab for their love and moral support.

My sincere thanks must also go to all the staff members of the Computer Science department at UQAM for their helps during my studies.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

RÉSUMÉ

Avec la croissance rapide des médias sociaux, une très grande quantité de données est générée. Ces données contiennent des informations utiles qui peuvent être utilisées par les organisations et les gouvernements pour prendre des décisions ou faire des prévisions ou pour des raisons d'assurance qualité. Une approche automatisée est nécessaire pour extraire des informations des données non structurées. C'est la tâche de l'exploration des données. Plus précisément, l'analyse des sentiments, en tant que sous-domaine de l'exploration des donnes, se concentre sur l'extraction d'opinions à partir du texte en langage naturel. L'analyse du sentiment des produits culturels a ses propres défis et c'est l'un des domaines les plus difficiles pour effectuer une analyse du sentiment. En effet, les gens n'expriment généralement pas directement leurs sentiments et les figures de style comme le sarcasme, l'ambiguïté du langage, les allégories, abondant.

Cette thèse a pour but d'effectuer une analyse de sentiments dans le cadre des revues des produits culturels. Deux approches sont étudiées pour réaliser cette tâche. La première est l'approche basée sur le lexique en utilisant POS, les adjectifs sont séparés et recherchés dans les sacs de lexique des sentiments négatifs et de lexique des sentiments positifs pour trouver à quelle catégorie ils appartiennent. Dans l'approche basée sur l'apprentissage machine, Weka est utilisé pour trouver la combinaison de prétraitement des données et d'algorithmes d'apprentissage machine qui donne la plus grande précision. L'API de données Youtube est utilisée pour extraire les commentaires d'une vidéo YouTube. Ensuite, en utilisant l'API Weka, l'algorithme sélectionné, qui est LibSVM, a été ainsi entraîner par l'ensemble de données "IMDB movie reviews". Ensuite, le classificateur formé est utilisé pour classer le texte. Les résultats montrent que l'approche de l'apprentissage machine donne une bonne précision. Afin d'obtenir de meilleurs résultats, il est préférable d'utiliser ensemble des approches basées sur la langue et une approche d'apprentissage automatique.

Mots clés : analyse des sentiments, PNL, traitement du langage naturel, API de données YouTube, Weka, extraction d'opinion, basé sur un lexique, apprentissage automatique.

ABSTRACT

With the rapid growth of social media, a very large amount of data is generated. These data contain useful information that can be used by organizations and governments to make decisions or predictions or for quality assurance reasons. An automated approach is needed to extract information from the unstructured data. This is the task of data mining. More specifically, sentiment analysis as a subdomain of data mining, that focuses on mining opinions from the text in natural language. Analyzing the sentiment of cultural product reviews has its challenges and it is one of the difficult domains for performing sentiment analysis tasks. Indeed reviewers usually don't express their feelings directly and use complex figures of speech like sarcasm, language ambiguity, allegories, etc to express their sentiment.

This thesis aims to perform the sentiment analysis task on cultural product reviews. Two approaches to perform the sentiment analysis task are studied. The first one is the lexicon-based approach from a POS tagged text, the adjectives are separated and searched in the bags of negative sentiment lexicon and positive sentiment lexicon to find out which category they belong to. The polarity of the text is relative to the number of positive and negative words. In the machine learning-based approach, Weka is used to find the combination of data pre-processing and machine learning algorithms that gives the highest accuracy. Then using Weka API the selected algorithm which is LibSVM was trained by the "IMDB movie reviews" data set. Youtube Data API is used to extract the comments of a YouTube video. Then the trained classifier is used to classify the extracted comments from the youtube video. The results show that the machine learning approach gives good precision. To obtain better results, it is better to use linguistic-based approaches with a machine learning approach together.

Keywords: sentiment analysis, NLP, Natural language processing, YouTube data API, Weka, opinion mining, lexicon-based, machine learning.

INTRODUCTION

0.1 WHAT IS SENTIMENT ANALYSIS?

The internet has changed all the aspects of our lives. One of the most affected things is communication and social life. Now people can make new friends, connect with old ones, share their feelings and emotions about a subject, and discuss their opinions with other people over the internet. There are many different platforms (*social media*) that allow users to share their opinions and feelings about various subjects in different forms like comments, reviews, etc.

The rapid growth of social media has generated a huge volume of data. That data is full of useful information that can be used to support important decisions. For example, a company may be interested in finding the opinions and perceptions of its customers, about the company itself, its competitors, its products and services, and even perceptions about its customers. Similarly, government agencies or political parties may be interested in finding out how the concerns of the citizenry, or how its services, policy proposals, or actual policies, are perceived by the public. They can do that by analyzing the mass of data available online, whether it is online product reviews, discussion forums, or entries in Facebook, Twitter, etc.

Data mining aims to find patterns and structures in data and to extract useful and actionable information from that data. When the data is unstructured natural language text, the mining of the data requires *Natural Language Processing* (NLP) techniques to: 1) 'clean' the data, such as removing punctuation, filtering out articles or pronouns such as "the", "that", "a", and 2) 'understand' the data, i.e. the meaning carried by the text. *Sentiment analysis* is an example of *data mining for natural language text* where the goal is to identify the "sentiments" or "opinions" of individuals about certain objects or subjects.

As a natural language processing (NLP) problem, sentiment analysis is difficult. The level of difficulty varies based on the domain and the type of text. For example, extracting opinions from discussion forums about politics is more difficult than performing the same task on tweets. Indeed, in forums, more than one author is expressing their opinions and the comments are usually longer, whereas tweets are short and written by a single author. Product reviews, in general, are simpler. They tend to be written by a single person, about a single product. In the simplest case, we might be interested in classifying the opinions of individuals about a product as either positive or negative, or neutral.

However, product reviews themselves can be complex. If the product is utilitarian, such as a washing machine, it does not evoke complex sentiments in its customers. However, if it is a so-called *lifestyle* product, such as fashion or sporting equipment, the 'sentiments' can be more complex than the binary like/dislike. The reviews of *cultural products*, such as movies, music, books, concerts or shows are probably the most complex because such products are to be 'consumed' or 'experienced' entirely subjectively.

This thesis is concerned with sentiment analysis within cultural product reviews. Our goal is to compare the different approaches to sentiment analysis and to try to identify which combination of linguistic resources, NLP techniques, and machine learning algorithms, yield the best results for cultural product reviews–in this case, youtube videos.

In this remainder of this chapter, we will first describe the context of this research (section 0.2). Next, we will describe briefly the objectives of this thesis (Section

0.3). Finally, we will describe the contents of this thesis (Section 0.4).

0.2 CONTEXT: A STUDY OF THE DISCOVERABILITY OF NATIONAL CULTURAL PRODUCTS WITHIN MULTINATIONAL PLATFORMS

This thesis is part of a team project led by Dr Michèle Rioux, a political science professor who is an associate member of LATECE. The project, titled "Mesure de découvrabilité des produits musicaux et audiovisuels québécois sur les plateformes numériques" and funded by Québec's FRQ-SC (*Société et Culture*) (2017-2019), aims at *measuring* the *discoverability* of musical and audiovisual products originating from Québec, within multinational platforms such as Youtube, Netflix, iTunes, Spotify, and the like. As more and more people consume their cultural products on such platforms, Prof. Rioux and her teamates (Diane-Gabrielle Tremblay, of TELUQ, and Hafedh Mili) wanted to explore whether Québec products are accessible through such platforms, with the intent to propose, at a later stage, *tools* and *policies* to the various stakeholders to help promote Québec products to Québec audiences, and beyond.

To this end, they set out to develop various *discoverability metrics*. The first of these metrics tests the *presence* or *absence*, of a particular product within the offering of the platform. More advanced metrics include the preeminence on these multinational platforms of such products and their recommendation to specific user categories. For example, one would expect Québec-based French-language movies or video-clips to be recommended–e.g. appear on the landing page–of Québec residents. The project involves, among other things, the development of web scraping tools that check the presence of Québec-issued products on such platforms either by using the APIs published by such platforms, or through "page scraping" of specific user sessions using these platforms.

One of the hypotheses that Professors Rioux and Mili wanted to test was whether *the quality of the reviews (positive versus negative) of a particular product influenced its discoverability*, and more specifically, the extent to which it would be 'recommended', or show-up higher in the suggested lists to specific user groups. A first step, to test this hypothesis, is to be able to accurately assess the sentiments expressed by reviews of such products. Preliminary experimentation showed this to be a particularly difficult problem.

0.3 CONTRIBUTIONS

As mentioned above (Section 0.1), sentiment analysis is a difficult problem, *in general.* While product reviews are *generally* easier to analyze than political discussion forums, for example, *cultural* product reviews tend to be particularly difficult, for at least two reasons. First, as mentioned above, cultural products tend to be consumed *emotionally*, i.e. for the extent to which they procure certain feelings and emotions, including, but not limited to, *pleasure.* For example, somebody who goes to watch a horror movie expects to be afraid; another who goes to watch a melodrama considers it a success if they cry during the movie. Thus, "I cried from start to end" review for the movie *Philadelphia*, say, is positive, whereas "I laughed from start to end" is not; the opposite is true for a comedy!

The second reason why the analysis of cultural product reviews is more difficult is that they usually contain very complex figures of speech, unlike the reviews of products such as a dishwasher, or a circular saw. Indeed, authors of such reviews would use things like metaphors, irony, sarcasm, plays of words, allegories, and the like. Further, this is a function of both the *writer* of the review, and its *intended audience.* A music professor would use a different language to describe the latest Beethoven concert she watched than the electrical drill she bought from Home

Depot. We even observed differences between subcategories. For example, the types of comments on classical or jazz video clips tend to use a distinctly different sublanguage from pop or country music reviews, say.

The following are examples of positive reviews on a video recording of a concert by Michel Camillo, a Jazz pianist, at the San Javier Jazz Festival of 2014 (https://youtu.be/ojByzJhwVFE), which would be particularly challenging:

- I hope these guys closed the festival out because no one should have to try to follow that
- Unreal. Stupifying. So dangerously well performed, so intensely pleasureable, exciting, it may soon be declared a controlled substance. Smoke it if you got it. And ty Michel etc al
- god has a master plan ....Michel Camilo
- OMFG. WTF. I honestly just keeping losing my shit, the further and further I watch. I'm quitting piano tomorrow, btw
- There's a solid pulse of a thousand silent cowbells driving St Thomas - don't doubt it

Exploring the problem in its entirety, i.e. being able to interpret "I cried from start to end" about the movie *Philadelphia*, and "I am quitting piano tomorrow" about Michel Camillo, as *positive comments*, is a major undertaking, and is beyond the scope of this thesis.

The purpose of this thesis is to propose *an incremental approach to enhance the accuracy of sentiment analysis of cultural product reviews by exploring different combinations of NLP and machine learning techniques.* To this end, we will start with a *baseline* that consists of the best combination of automated tools and available resources, and then propose enhancements relative to that baseline. The proposed enhancements will be evaluated along two important dimensions: 1) the resulting change in the accuracy of the analysis, and 2) the required effort.

To find the baseline, we will try two representatives of the main approaches to sentiment analysis: 1) the lexicon-based approach, and 2) the machine learning approach, using the WEKA workbench. Unsurprisingly, the machine learning approach yielded much better results, and will thus be used as a baseline. In particular, the libSVM algorithm, with basic text pre-processing (stop word removal and word stemming), yielded an 85 % accuracy rate on the IMDB data set, which is a dataset of movie reviews.

Our next step is to test different enhancements on the basic algorithm, by trying one of, or a combination of the following:

- More advanced natural language processing techniques: for example, use part-of-speech tags to exclude certain word from the analysis
- Different encodings of the reviews (e.g. n-grams as opposed to words)
- The use of readily accessible external linguistic resources

0.4 CONTENTS OF THE THESIS

The first chapter provides a literature review of sentiment analysis. In particular, we discuss the different levels of sentiment analysis, and the methods to solve each level are explained. The chapter ends by providing descriptions of possible and most commonly used data pre-processing techniques.

Chapter 2 includes a very high-level overview of machine learning techniques. It will be expanded ine the final version of the thesis.

Chapter 3 includes, 1) an explanation of the two methods used to obtain a baseline, namely the lexicon-based and the machine learning-based approaches, 2) a description of the tools developed by the candidate to perform the analysis, and 3) the results of the experiments. In the final version of the thesis, these contents will be in three different chapters.

Chapter 4 concludes the thesis in its current form. It will be revised once the work described in 0.3 is completed.

CHAPTER I

STATE OF THE ART ON SENTIMENT ANALYSIS

When people need to make a decision, they usually seek for the opinions of others. Social media that allows users to communicate and share information and opinions over the Internet is generally the best way to find or request the opinions of others online. In these media, people share their opinions and feelings about the various subjects in a natural language text (English, French, etc.) in the form of messages, comments, reviews, etc.

In recent years, the rapid growth of social media has generated a huge amount of unstructured data. Analyzing this data requires efficient automated techniques. Data mining techniques, allow us to extract useful information from unstructured data such as the polarity of opinions or reviews, finding patterns in data, or making predictions. Sentiment analysis, or opinion mining, is a sub-domain of Data Mining which specifically consists of extracting the opinions of individuals according to certain objects in a text in natural language.

In the following, first of all, sentiment analysis is defined. Then possible inputs and outputs of the sentiment analysis task as well as techniques to resolve each of them are explained in detail. The inputs can be classified into three different levels: document level, sentence level, and aspect level. At the end, some common techniques for preparing data before doing sentiment analysis are discussed. One of the best books in the sentiment analysis field is Liu's book (Liu, 2015). In writing this chapter we used this book as a reference and many of the definitions are based on this book.

1.1 SENTIMENT ANALYSIS

Many social networking sites were created within the 1990s. Later in 2000, social media received a great boost thanks to the birth of many social networking sites (Edosomwan et al., 2011). Because of the rapid growth of social media, a very large amount of data recorded on digital forums was quickly generated. The analysis of this amount of data is not possible manually, so techniques for sentiment analysis have been developed to solve this problem. Since the beginning of 2000 at the same time as the growth of social media, sentiment analysis has become one of the most active research domains in natural language processing (Liu, 2015).

Based on Liu's definition in his book, "sentiment analysis is a field of study that aims to extract opinions and feelings from a text in natural language text using calculation methods" (Liu, 2015).

The problem of sentiment analysis is generally considered as a text classification problem. Considering a text document as input, the task of sentiment analysis is to categorize an opinion into a positive or negative opinions (Kamal et al., 2016).

There are different approaches to solve the problem of sentiment analysis such as the lexicon-based, machine learning-based, or linguistic approaches (Taboada et al., 2011). The lexicon-based approach assumes that the contextual sentiment orientation of a document is the summary of the sentiment orientation of each word or phrase. Machine learning algorithms that are commonly used to solve text classification problems can also be applied to sentiment analysis problems. In this approach, classifiers can be built from labeled instances (positive or negative) of texts or sentences (Palanisamy et al., 2013). The linguistic approach uses the semantic characteristics of words or sentences, negation, and the anatomy of the text (Shirsat et al., 2018).

Sentiment analysis is also considered a Natural Language Processing (NLP) problem. In order to explain the relationship between sentiment analysis, natural language processing, and machine learning, we will first briefly describe natural language processing and machine learning and their place in artificial intelligence.

Artificial Intelligence (AI) "A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Nowak,). It aims to simulate the human brain to create intelligent systems. AI enables computers to do the processes as learning, reasoning, and self-correction. (Kok et al., 2010) .

Natural Language Processing (NLP) "Natural Language Processing is the analysis of linguistic data, most commonly in the form of textual data such as documents or publications, using computational methods" (Verspoor & Cohen, 2013). It enables computers to understand human languages (Khurana et al., 2017).

Machine learning (ML) is also a sub-field of artificial intelligence that allows computers to learn from data and not by being explicitly programmed (Langley & Carbonell, 1984).

Sentiment analysis can be defined as a process of extracting opinions, sentiment, emotions and so on from text in natural Language (A. & Sonawane, 2016).

The relationship between these four concepts is demonstrated in figure 1.1.

Natural language processing helps computers understand the human-generated text. this is usually done using learning techniques and algorithms. Thus, as in

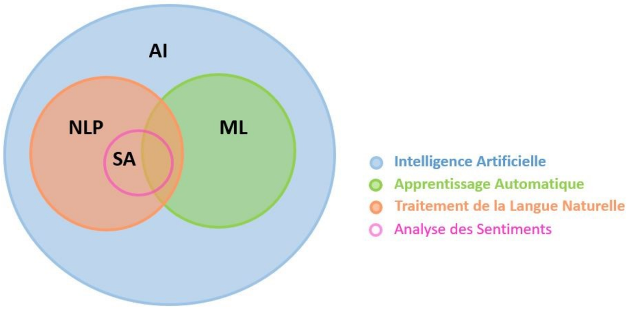

AI
NLP
ML
SA
Intelligence Artificielle
Apprentissage Automatique
Traitement de la Langue Naturelle
Analyse des Sentiments

Figure 1.1 Relationship between AI, NLP, Ml and SA

sentiment analysis, we have to process human-generated text, we can say that sentiment analysis is an application of NLP. As a text classification problem, sentiment analysis can be done by using machine learning techniques to classify opinions, so we can say that it is also an application of machine learning.

1.1.1 DEFINITIONS

Before going into details, we first explain some definitions that are used in this document.

Sentiment Lexicon: "List of words or expressions that people often use to express positive or negative opinions. They are sentiment indicators. Apart from individual words, some phrases and idioms that can also be used to indicate sentiments" (Liu, 2015).

Sentiment (s): "The underlying feeling, attitude, evaluation or emotion associated with an opinion" (Liu, 2015).

Sentiment Intensity (i): Sentiment can be expressed in different levels of strength or intensity. Intensity can be defined by using sentiment lexicons or by using "intensifiers" or "diminishers" to increase or decrease the degree of expressed sentiment. Intensifiers like very, so, really, and so on, and diminishers like a little bit, barely, pretty (Liu, 2015).

Sentiment rating: It is used to define the sentiment intensity. Liu categorizes sentiment rating into five commonly used ratings (Liu, 2015):

- Emotional positive(+2or 5 stars)
- Rational positive(+1 or 4 stars)
- Neutral(0 or 3 stars)
- Rational negative(-1 or 2 stars)
- Emotional negative(-2 or 1 star)

Having more rating levels increases the complexity of the sentiment analysis problem as well as the accuracy of the results. For example, in "I love painting" the sentiment expressed about painting is stronger than "I love painting" (Liu, 2015).

Sentiment orientation or polarity (o): Sentiment can be defined as positive, negative, or neutral. When no sentiment or no feeling is expressed the sentiment orientation is defined as neutral (Liu, 2015).

Sentiment target (g): "The entity or attribute of the entity that the sentiment has been expressed upon" (Liu, 2015). For example in the phrase "I love the color of this car", the sentiment target or the opinion target is the aspect "color" of the entity "car" (Liu, 2015).

Liu defines an entity as a pair e:(T, W), in which "T" is the hierarchy of parts, subparts, and so on and "W" is a set of attributes of the entity. For example for an entity "cellphone", size, camera quality, network technology are some of the attributes. The processor, battery, display, RAM are parts of it. Each part can have its attributes, like the picture quality of the camera or battery life.

Opinion holder (h): The person or entity expressing the opinion is the opinion holder of that opinion.

Because of the subjective nature of opinion and sentiment, sentences expressing them are usually subjective. Subjective sentences usually state sentiment whereas objective sentences usually state facts (Liu, 2015). This doesn't mean that objective sentences cannot contain sentiment. For example "Since I went to that restaurant last night, I have been having stomachache" is an objective sentence that describes a fact, but we can see that the sentence author feels negative about the restaurant (Liu, 2015).

Opinion: "A view, judgment, or appraisal formed in the mind about a particular matter" (Liu, 2015). There are two different types of opinions, regular and comparative opinions. Regular ones express an opinion about one single entity or an aspect of it (ex: in "The price of iPhone 11 pro is high", "price" is an aspect of the entity "iPhone" that the sentiment is expressed about it.) but comparative opinions compare multiple entities based on some of their shared aspects (ex: in "The price of iPhone 11 pro is higher than Samsung A60", compares two entities "iPhone 11 pro" and "Samsung A60" based on their common aspect "price"). The most common types of opinions are the regular ones.(Liu, 2015).

From the subjectivity point of view, opinions can be classified into two types, subjective opinions, and fact-implied opinions. A subjective opinion, given in a subjective sentence, can be a regular or comparative opinion. For example "The price of iPhone 11 pro is high" or "The price of iPhone 11 pro is higher than Samsung A60" (Liu, 2015).

A fact-implied opinion, given in an objective sentence, can be also a regular or comparative opinion. For example "I got the phone last night and it broke today" or "My phone is cheaper than my dad's phone" (Liu, 2015).

An opinion also can be expressed by the first person or the non-first person. For example in the phrase "I think my mom likes her new car", the opinion holder is not the one writing the review or comment, and the writer is stating someone else's opinion. Based on Liu, an opinion is a quadruple (g,s,h,t). Where "g" is the sentiment target, "s" is the sentiment of the opinion about that target, "h" is the opinion holder, and "t" is the time when the opinion is expressed. These four components are essential. For example, we might study the opinion of a specific opinion holder or we might want to study the reviews after a change in product quality happened. Also important to know the sentiment is related to which target (Liu, 2015).

The opinion can also be defined as quintuple:

(e,a,s,h,t) (Liu, 2015)

where "e" is the target entity and "a" is the target aspect of entity "e". The opinion should be given about "a". When opinion is expressed on the entity itself, not its aspects, we use the special aspect "GENERAL" for "a". (Liu, 2015).

Reason for opinion "is the cause or explanation of the opinion" (Liu, 2015) and *Qualifier for opinion* "limits or modifies the meaning the opinion" (Liu, 2015). For example in the phrase "This camera is not good for photography at night because it doesn't have flash", "photography at night" is the qualifier of the opinion, and "it doesn't have flash" is the reason for the opinion. Qualifiers for opinion statements are not common but reasons are quite common (Liu, 2015).

Liu defines the difference between opinion and sentiment as following: "Opinion is more of a person's concrete view about something, the sentiment is more of a feeling". We can't agree/disagree to sentiment but to an opinion, we can't agree/disagree (Liu, 2015).

Text content in social media can be divided into two categories: standalone posts, and online dialogues. In standalone posts, such as reviews, each opinion holder's review is independent of another opinion holder. In online dialogues, such as debates and discussions, multiple opinion holders exchange opinions (Liu, 2015). Comments (for example comments about a video or a picture) are a mix of standalone posts and online dialogues.

The degree of difficulty of the sentiment analysis problem is related to the text content type as well as the application domain. For example, as tweets are short and have a length limit, opinion holders are usually straight to the point. Consequently, sentiment analysis methods tend to work rather well and tend to have high accuracy rates. Debates such as social and political discussions are the hardest to deal with as users can discuss anything and there are many participants involved. Reviews about products or services are the easiest to deal with because they are highly concentrated on the target (Liu, 2015).

Based on Liu, Sentiment analysis task is to discover all the quintuples (e,a,s,h,t) in document "d" which is a finite set of entities: $\{e_1, e_2, \ldots, e_r\}$. The task can be divided into eight subtasks defined as following:

Task1 (entity extraction and resolution): The first step is to extract all entities in the document. Then synonymous entity expressions (word or phrase that indicates an entity) should be grouped into a set called entity clusters (or categories). This categorization is important to be done as an entity can be referred to in several ways. For example "Instagram" and "Insta" are pointing to the same entity. In the end, each entity expression cluster refers to a unique entity (Liu, 2015).

Task2 (aspect extraction and resolution): This task is the same as task one but for aspects of the entity. In this task, all aspect expressions of the entities are extracted and grouped into clusters (Liu, 2015).

Task3 (opinion holder extraction and resolution): Same as what is done in task one and two, but for opinion holder. For each opinion, the holder expression is extracted and grouped into clusters (Liu, 2015).

Task4 (time extraction and standarization): The posting time of each opinion is extracted and different time formats is standardized (Liu, 2015).

Task5 (aspect sentiment classification or regression): Determining the polarity of an entity or aspect (Liu, 2015).

Task6 (opinion quintuple generation): By using the outputs of tasks 1-5, all opinion quintuples (e, a, s, h, t) in the document should be generated (Liu, 2015).

Task7 (opinion reason extraction and resolution): This task focuses on extracting all reason expressions for each opinion and categorizing them into clusters (Liu, 2015).

Task8 (opinion qualifier extraction and resolution): This task is similar to task7 but for opinion qualifiers. For each opinion, qualifier expressions are extracted and grouped into clusters (Liu, 2015).

1.2 LEVELS OF SENTIMENT ANALYSIS

Based on granularity, different levels of sentiment analysis can be envisaged. We can consider an entire document as a single entry or consider each sentence separately. Generally, sentiment analysis is classified into three levels, document level, sentence level, and aspect level. In the following, each level is explained in more detail.

1.2.1 DOCUMENT LEVEL

This level is the simplest. "At the document level, first we assume that the entire document is about a single entity" (Liu, 2015), then we classify the entire document. It is called document-level analysis because it considers each document as a whole and does not look at entities or aspects within the document (Liu, 2015). This level does not consider the target of the sentiment and cannot be applied on documents that contain evaluations or comparisons of more than one entity.

In this level, it is assumed that the entire document contains opinions about a single entity "e" from a single opinion holder "h". we can formulate a document-level sentiment analysis as below: (Liu, 2015).

(-,GENERAL,s,-,-) (Liu, 2015)

where "e", "h", "t" are assumed to be known or irrelevant.

If the output takes categorical values (positive or negative or neutral), sentiment analysis is a classification problem, a numerical value (1..5), is a regression problem (Liu, 2015).

The disadvantage of this approach is considering an entire document as being about a single entity. There can be usually multiple entities discussed in a document. The sentiment expressed for each entity can be different, so considering a whole document as being about a single entity degrades the accuracy of the results. This level is useful for special types of document like reviews, as reviews are generally about one single entity but not for documents like forum discussions or dialogues.

1.2.1.1 SUPERVISED SENTIMENT CLASSIFICATION METHODS

As Liu says in his book, sentiment analysis is a text classification problem. All supervised machine learning algorithms can be used to solve the document level sentiment classification problem. Pang et al. used naive bayes and support vector machines (SVM) to classify movie reviews into two category positive and negative. They showed that using unigram (bag of words) as the fautures, results quiet well (Pang et al., 1988). To simplify the classification problem, researchers eliminate the neutral class and just consider the sentiment to be positive or negative.

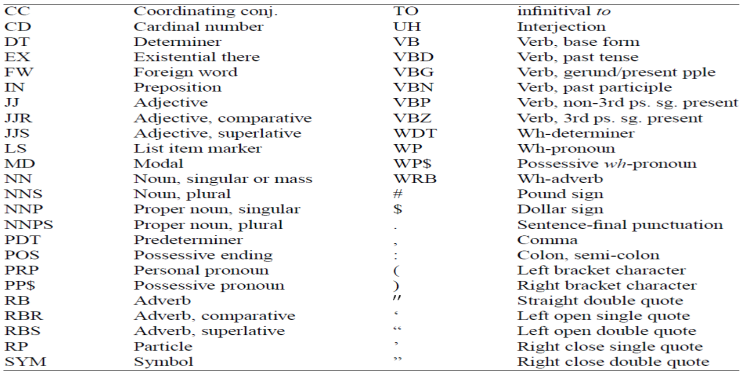
The combination of features and algorithms has been the focus of many researchers. In the following, some of the features are listed (Liu, 2015).

Terms and their frequency: Word (unigram) or a set of co-occurring words (n-grams) and their frequencies (number of times that they occur in the document or sentence) are considered as features. In traditional text classification, this is the most common feature used (Liu, 2015).

Part of speech: Part of speech (POS) defines the functional role of each word in a sentence such as nouns, verbs, adverbs, adjectives, etc (Jatav et al., 2017). It has been proven that adjectives are the main indicators of opinions in the phrases. Table 1.1 describes the standard Penn TreeBak POS tags (Liu, 2015).

Sentiment lexicon: Previously explained in section 1.1.1. They are mainly adjectives and adverbs.

Sentiment shifters: They are mainly words that can change the orientation of



CC	Coordinating conj.	TO	infinitival *to*
CD	Cardinal number	UH	Interjection
DT	Determiner	VB	Verb, base form
EX	Existential there	VBD	Verb, past tense
FW	Foreign word	VBG	Verb, gerund/present pple
IN	Preposition	VBN	Verb, past participle
JJ	Adjective	VBP	Verb, non-3rd ps. sg. present
JJR	Adjective, comparative	VBZ	Verb, 3rd ps. sg. present
JJS	Adjective, superlative	WDT	Wh-determiner
LS	List item marker	WP	Wh-pronoun
MD	Modal	WP$	Possessive *wh*-pronoun
NN	Noun, singular or mass	WRB	Wh-adverb
NNS	Noun, plural	#	Pound sign
NNP	Proper noun, singular	$	Dollar sign
NNPS	Proper noun, plural	.	Sentence-final punctuation
PDT	Predeterminer	,	Comma
POS	Possessive ending	:	Colon, semi-colon
PRP	Personal pronoun	(	Left bracket character
PP$	Possessive pronoun	)	Right bracket character
RB	Adverb	″	Straight double quote
RBR	Adverb, comparative	‘	Left open single quote
RBS	Adverb, superlative	“	Left open double quote
RP	Particle	’	Right close single quote
SYM	Symbol	”	Right close double quote

Table 1.1 Penn TreeBak POS tags (Mititelu, 2007)

the expressed sentiment. For example, if the sentiment is positive, by using shifter words the sentiment will be negative. Examples of shifters are the negation words as "not", "but", and so on. In the phrase "I didn't like the party", although the verb "like" has a positive sentiment, the negation word "not" changes the sentiment to negative (Liu, 2015).

Another feature can be word substitution. For example, replacing the entity name with a token "_entityName" or replacing rare words by "_unique" (Liu, 2015). Based on (Dave et al., 2003), other linguistic modifications using WordNet, stemming, negation, and collocation are not that helpful and even reduced the accuracy of the classification.

1.2.1.2 UNSUPERVISED SENTIMENT CLASSIFICATION METHODS

As mentioned previously, supervised or unsupervised approaches can be used to solve the problem of sentiment analysis. In the following, we briefly review some of the unsupervised approaches. One of them is based on sentiment lexicons.

This method uses a dictionary of sentiment words and phrases with the associated sentiment orientation and sentiment intensity of each word or phrase. This dictionary is called sentiment or opinion lexicon (Liu, 2015).

A positive "SO" (sentiment orientation) value is defined to each positive word or phrase, and a negative "SO" for the negative ones. In the end, all the SO values are summed up. If the total SO is positive, the document is classified positive, and if the sum is negative, the document is classified as negative. If the total is 0 then the document is neutral. This method can be improved in many ways. For example, we can reverse the sentiment if sentiment shifters are used (Hu & Liu, 2004).

Kennedy & Inkpen considered the intensifiers and diminishers in their work (Kennedy & Inkpen, 2006). Intensifiers increase the degree of intensity of the sentiment and diminishers decrease the degree. They assigned the value 2 to each positive expression and if there is an intensifier, the value 3 is assigned to the expression. A negative expression gets the value -2, the value -1 is assigned if it is preceded by a diminisher and -3 if it is preceded by an intensifier (Liu, 2015).

Taboada et al, extended this method (Taboada et al., 2011). Values -5 to +5 are assigned to sentiment expressions from extremely negative to extremely positive (value 0 is not used). Then to each intensifier or diminisher, a weight in percentage is assigned (ex: +100 for most, +25 for very, pretty -10, so on) (Liu, 2015). And then the SO of each sentiment expression is calculated by using these values and weights.

Although Sentiment lexicons play an important role in defining the sentiment, accurate sentiment analysis cannot be done by only focusing on sentiment lexicons. Depends on the context and domain, a sentiment word may express an opposite polarity. for example, the word "suck" has a negative polarity in general, but in the phrase "This vacuum cleaner really sucks" (Liu, 2015), it implies a positive sentiment. Sometimes a phrase containing sentiment words may not express any sentiment. This is mainly happening when we are talking about interrogative or conditional sentences (Liu, 2015). But we can't also extend this rule to all these two types of sentences. For example, although there is the sentiment word "good" in the phrase "Is Asus laptop good?", the phrase expresses no sentiment. The other exception is sarcastic sentences. Whether they contain sentiment words or not, they might express an opposite polarity. For example in the phrase "What a great car? It stopped working in two days!" we can't define the accurate polarity of the phrase by only concentrating on the sentiment word. And at the end, it is possible that a phrase without any sentiment word expresses sentiments. For example, the phrase "This washer uses a lot of water" expresses a negative sentiment about the washer (Liu, 2015).

1.2.1.3 CROSS-DOMAIN SENTIMENT CLASSIFICATION

In sentiment classification problem, the input is a set of words structured as sentences and documents. Different domains use different special words and expressions. So when solving a sentiment classification problem, we should always consider the domain. In supervised machine learning approaches, the training dataset used for one domain should not be used for another domain. In case we use it to train a classifier, it might lead to wrong results.

Aue & Gamon proposes some approaches to apply classifiers of one domain to another domain (Aue & Gamon, 2005). One is to make the training data from labeled data from other source domain and then do the testing on the target domain. Another is to use the training set from other domains but we only use the features of the source domain that exist in the target domain then we do the test on the target domain.

In (Yang et al., 2006), features that are highly ranked in two labeled training sets from two different domains are selected as the domain-independent features. Then the classifier can be trained using this training data and can be applied to any domain.

1.2.1.4 CROSS-LANGUAGE SENTIMENT CLASSIFICATION

An international company prefers to have a single sentiment analysis system that can be applied to classify the reviews on products or services in different countries. Also, a document can be in any language, So there is a need to have a sentiment analysis system that can be used on languages other than English. The problem here is to use an existing sentiment analysis system in English to build another sentiment analysis system that can be used in other languages (Liu, 2015).

Wan defined an algorithm that at first step uses multiple translators to translate the Chinese reviews to English (Wan, 2008). Then the English translations were classified using a lexicon-based approach. If the Chinese lexicon is available, the lexicon-based approach can be applied to the Chinese ones. Then the results of these two approaches can be combined. Wan showed that the combined technique is effective (Wan, 2008).

Co-training is a semi-supervised learning technique that requires two views of data. They are used when there is a shortage of labeled data when the amounts of labeled data are smaller than the amounts of unlabeled data (Liu, 2015). The basic two view co-training suggests to described data by two disjoint sets of features or views. An initial (small) set of labeled training data and a (large) set of unlabeled data from the same distribution is used as a training set.

Two classifiers are first trained on the initial labeled training set using the two views separately (Du et al., 2011). Then, each classifier classifies the unlabeled data, chooses the few unlabeled examples whose labels it predicts most confidently, and adds those examples (with the predicted labels) to the training set. The classifiers are retrained, and the process repeats until some stopping criterion is met. That is, the two classifiers "teach" each other with the additional examples whose labels are given by the other classifier, so as to improve the classification accuracy.(Du et al., 2011)

(Wan, 2009) used a co-training method that was using labeled English corpus and without using any Chinese resources. The training set contained labeled English reviews and unlabeled Chinese reviews. The unlabeled Chinese reviews used for co-training do not include the unlabeled Chinese reviews for testing.

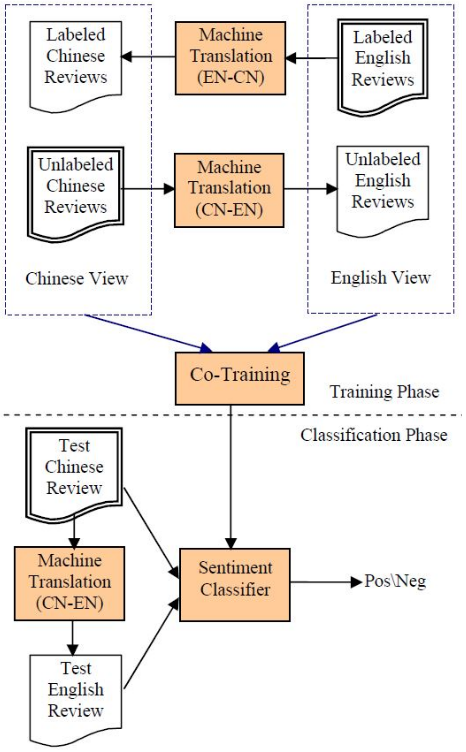


Figure 1.2 Framework of proposed approach of (Wan, 2009)

Labeled English reviews were translated to Labeled chinese and unlabled chinese reviews were translated to unlabled English. Each review was associated to a chinese version and an English version. English features and chinese features were considered as two independent and redundant views (Liu, 2015). A co-training SVM algorithm then trained two classifiers as figure 1.2. After the training was done, to classify the test data, each chinease review was translated to English the trained classifier was used to classify them as positive or negative.

1.2.2 SENTENCE LEVEL

Based on Liu's book, at this level, we consider each sentence as an entry and then we classify it. If we consider each sentence as a document, this level is very similar to the document level. Compared to the sentiment classification at the document level where we usually classify documents into two categories (positive and negative), at the sentence level we usually classify sentences into three categories (positive, negative, and neutral).

The reason for this is that sometimes there are sentences that have no feelings. Subjective sentences contain sentiment and opinions and objective sentences express factual information. Subjective sentences can be classified in positive or negative categories although objective sentences belong to the neutral class (Shirsat et al., 2018). This is one reason that sentence-level sentiment classification is more difficult than document level, as classifying text into three classes is more difficult than classifying into two classes. The other reason is that sentences are shorter than documents, so they contain less information.

Researchers assume that each sentence is expressing a single sentiment. In this level, sentiment targets are not taken into account which is problematic in some cases. For example when the sentence is comparative or when there is more than one sentiment in a phrase, the result of classification might not be accurate. For example in the sentence: "Our company is doing well in this COVID19 situation", the sentiment for entity "company" is positive but the sentiment for the entity "situation" is negative (Liu, 2015).

Liu mentions two approaches to solve the problem of sentence sentiment classification. One is to consider it as a three-class classification problem and the other is to consider it as two separate two-class classification problem (Liu, 2015).

The first approach is called "subjectivity classification" which is used to determine if a sentence contains subjective information or objective information (Liu, 2015). But as mentioned before, a subjective sentence might express no opinion. The sentiment is a subconcept of subjectivity. So better to call the classes "opinionated" if it is expressing a positive or negative opinion, and "not-opinionated" if it implies no positive or negative opinion, regardless of considering if the sentence is objective or subjective, then classifying sentences of class "opinionated" (Liu, 2015).

In the second approach, the first step is to define if a sentence expresses an opinion or not. The second step is to classify the sentences that express opinions, into a positive or negative class. This way we eliminate the neutral class.

Supervised learning is the common approach used for subjectivity classification problems. M., Janyce Wiebet, Rebecca used Naive Bayes classifier to perform subjectivity classification (M., Janyce Wiebet, Rebecca, 1998). The presence of pronoun, adjective, a modal other than "will" was considered as features. Classifying tweets also can be considered as a sentence-level classification problem as tweets are usually short.

Researches are mainly focused on solving the problem of sentence sentiment classification without considering the structure of sentences. One challenge is to deal with conditional sentences. Conditional sentences contain two clauses, one is the condition and the other one is the consequent. The relationship between the condition and consequent clauses has a lot of impact on the final sentiment of the sentence (Liu, 2015).

Below is a list of some of the patterns in Liu's book that indicate sentiment in conditional sentences. These patterns are mainly used in documents like reviews, online discussions and are not useful for non-conditional sentences or other types of documents.

```
POSITIVE ::= ENTITY is for you
           | ENTITY is it
           | ENTITY is the one
           | ENTITY is your baby
           | go (with | for) ENTITY
           | ENTITY is the way to go
           | this is it
           | (search | look) no more
           | CHOOSE ENTITY
           | check ENTITY out
NEGATIVE ::= forget (this | it | ENTITY)
           | keep looking
           | look elsewhere
           | CHOOSE (another one | something else)
CHOOSE   ::= select | grab | choose | get | buy | purchase | pick | check | check
ENTITY   ::= this | this ENTITY_TYPE | ENTITY NAME
```

In which, ENTITY_TYPE is a product type for example a computer and ENTITY NAME is the name of the entity like Nissan.

Another challenging sentence type is the interrogative sentence. Sometimes an interrogative sentence expresses no sentiment, For example, "Where can I buy a good TV?" is expressing no sentiment. But "Why Samsung phones are crashing all the time?" expressing a negative sentiment. (Liu, 2015) believes that to have a more accurate sentiment analysis, each type of sentence should be handled differently. Little work has been done in this area.

Same as conditional and interrogative sentences, dealing with sarcastic sentences is challenging. When someone says something positive but he/she means something negative and vice versa. Based on (Liu, 2015) some researchers have worked on this problematic recently but much research is needed in this direction.

Tsur et al used a semi-supervised approach to identify sarcasms (Tsur et al., 2010). They used a small set of labeled data. But instead of using unlabeled data in the training set, they expand the labeled data automatically through a web search. They considered the experience that sarcastic sentences frequently co-occur in text with other sarcastic sentences (Liu, 2015). Then they used up to fifty search engines to perform an automated search by using each sarcastic sentence to find other sarcastic sentences. Then the results were added to the training dataset. This training dataset was used to train a classifier.

1.2.3 ASPECT LEVEL

Sentiment analysis at the document and sentence level does not take into account target opinions. These levels assume that the entire document or sentence is about one single entity. But it is also possible that a document or sentence may contain opinions about more than one entity. Even if we assume that the opinion of a document or sentence about one single entity is positive or negative, we cannot say that the opinion is positive or negative about every single aspect of that entity.

To obtain accurate results, it is important to define the sentiment target. Based on Liu, in the aspect level sentiment analysis instead of considering linguistic units (documents, paragraphs, sentences, etc.), we look at the opinion targets and the related opinion. The objective of this level is to find out the opinions about entities and their aspects (Shirsat et al., 2018).

For example in the sentence: "Although the service is not great, I still love the restaurant" we know that the sentence has a positive tone. But we cannot say it is entirely positive. We can say that it is positive about the restaurant and negative about the service.

To perform the aspect-based sentiment analysis deep NLP capabilities are required. As Liu mentions in his book, researches' focus is mainly on two tasks, aspect extraction, and aspect sentiment classification. In the aspect extraction part, the focus is on extracting the entities and their aspects that are mentioned in the text. Aspect sentiment classification is focusing on finding the sentiment orientation of the entity or its aspect.

To solve the problem of aspect-based sentiment classification supervised learning and unsupervised lexicon-based approaches can be used. Aspect-based sentiment classification also is called target-based sentiment analysis or entity-based sentiment classification in different application domains.

1.2.3.1 SUPERVISED ASPECT-BASED SENTIMENT CLASSIFICATION

In supervised learning, the same learning algorithms as SVM or Naive Bayes can be used. The difference is that opinion targets should be considered in the learning phase. Liu mentions two common approaches. The first approach consists of generating a set of features in each document or sentence that are linked to entities or their aspects. The second is to find the application scope of the sentiment, then verify if the entity or its aspects are included in that scope.

Jiang et al use a syntactic parse tree to represent the syntactic relationship between target entities or their aspects and the other words in the tweets (Jiang et al., 2011). The approach is below:

Based on pre-defined rules, for any word stem w_i in a tweet corresponding target-dependent features is generated", below some of the rules are listed (Jiang et al., 2011):

- if the target "T" is the object of the transitive verb w_i, A feature w_i_arg2 is generated. For example, for the target "this movie" in "I love this movie", we generate "love_arg2" as a feature (Jiang et al., 2011).
- if the target "T" is the subject of the transitive verb w_i; A feature w_i_arg1 similar to Rule 1 is generated. For example, for the target "Joe" in "Joe discussed the problems with her", we generate "discussed_arg2" as a feature (Jiang et al., 2011).
- if the target "T" is the subject of the intransitive verb w_i; a feature w_i_it_arg1 is generated. For example, for the target "Mary" in "Mary laughed", we generate "laughed_arg2" as a feature (Jiang et al., 2011).
- if the target "T" appears in the previous sentence of the adjective or intransitive verb appearing alone as a sentence w_i; a feature w_i_arg is generated. For example, in "You made it. Perfect!", "Perfect" appears alone as a sentence, so we generate "Perfect_arg" for the target "You".
- if the target "T" is the subject for the verb that the adverb w_i modifies; a feature arg1_v_w_i is generated.

When any word used in the generated target-dependent features is changed by negation, then in the generated features a prefix "neg-" is applied to it.

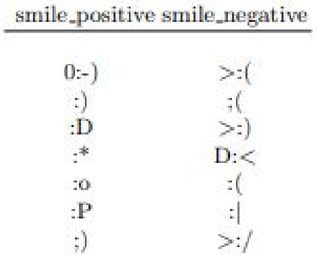
1.3 TEXT PRE-PROCESSING TECHNIQUES

The first step of sentiment analysis is the preprocessing of raw data sets or normalization and noise removal. As data is extracted from online social media it typically contains lots of noise such as irrelevant text, emojis, special characters, HTML tags, etc. (Shirsat et al., 2018). Preprocessing improves the performance of sentiment analysis. Some useful techniques for preparing the data are explained in the following.

1.3.1 REPLACE URLS, REFERENCES, HASHTAGS AND SMILIES

Hashtags are keywords that are used to identify a topic. A hashtag is a set of keywords immediately prefixed by the pound(#) symbol. Using hashtags is very common on Twitter. Davidov et al used Twitter hashtags to learn a wide range of emotions and the relations between the different emotions (Davidov et al., 2010). A tweet includes more than a single hashtag. In the pre-processing phase, they replaced URL connexions, hashtags, and references with meta-words for URL/REF/TAG. They used the Amazon Mechanical Turk (AMT) service to obtain a list of the most commonly used and unambiguous ASCII smileys.

Angiani et al converted types of emoticons shown in figure 1.3 into tags that express their sentiment (Angiani et al., 2016). For example ":)" is converted to a "smile happy" tag. The list of emoticons that they used was taken from Wikipedia. Then they classified emoticons to only two categories: smile positive and smile negative.



smile_positive	smile_negative
0:-)	>:(
:)	;(
:D	>:)
:*	D:<
:o	:(
:P	:|
;)	>:/

Figure 1.3 List of substituted smilies (Angiani et al., 2016)

1.3.2 LOWERCASING

Lowercasing is a common and simple text pre-processing technique. Although lowercasing is usually useful, it may not apply to all cases or scenarios. For example, if the task is detecting the programming language from the given text, one of the important features to be used is that some languages are case-sensitive, if we lowercase the text, we cannot use this feature anymore. Another approach is to only lowercase the initial words.

1.3.3 REMOVING STOP WORDS

Stop words are words that have a high frequency in text, such as "a", "or", "the", etc. Javed & Kamal used Python NLTK to remove the stop words from the text (Javed & Kamal, 2018). There is no such fixed set of stop words that can be used in any domain. A stop words in a domain can be a key sentiment in another domain. There are benefits to removing stop words, for example, reducing the size of the data set which can increase the performance, because when the number of features to be considered are less, training time will be reduced. Another benefit can be that by removing extra words the focus will be on more important words.

1.3.4 STEMMING AND LEMMATIZATION

Stemming is the process of reducing all the terms with the same stem to a common form (Javed & Kamal, 2018). In stemming, letters from the end of words are removed until the stem is reached. By using stemming, words like "cute", "cuter" and "cutest" are considered as the word "cute".

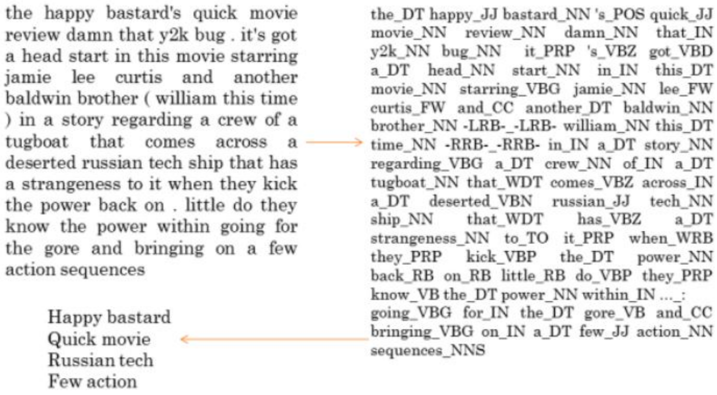
Lemmatization is the process of removing inflectional endings and replacing this inflected word with a base word (Javed & Kamal, 2018). The Lemmatizer which is used to do the lemmatization uses an additional dictionary to replace the inflected forms into its base form. In lemmatization, "feet" becomes "foot" and "wolves" becomes "wolf".

The output generated by Lemmatizers is more accurate comparing to the output generated by stemmer but the task of lemmatization is also more difficult than stemming. For example, in stemming the terms "begging", "beggar" and "beginning" will be replaced with the terms "beg" while in lemmatization these terms will be replaced with terms "beg", "beg" and, "begin", or feet becomes foot and wolves becomes wolf. The problem is that not all forms are reduced using lemmatization.

1.3.5 PART OF SPEECH TAGGING

Part of Speech (POS) Tagging is the process of assigning parts of speech to each desired term (Javed & Kamal, 2018).

(Das & Chandra Balabantaray, 2014) used POS tagger which is a tool developed by Stanford University. It takes as a file text as an input and returns the annotated text in which each word is followed by its parts of speech tag. python NLTK is another tool used by researchers to perform the part of speech tagging. The



the happy bastard's quick movie review damn that y2k bug . it's got a head start in this movie starring jamie lee curtis and another baldwin brother (william this time) in a story regarding a crew of a tugboat that comes across a deserted russian tech ship that has a strangeness to it when they kick the power back on . little do they know the power within going for the gore and bringing on a few action sequences

the_DT happy_JJ bastard_NN 's_POS quick_JJ movie_NN review_NN damn_NN that_IN y2k_NN bug_NN it_PRP 's_VBZ got_VBD a_DT head_NN start_NN in_IN this_DT movie_NN starring_VBG jamie_NN lee_FW curtis_FW and_CC another_DT baldwin_NN brother_NN -LRB-_-LRB- william_NN this_DT time_NN -RRB-_-RRB- in_IN a_DT story_NN regarding_VBG a_DT crew_NN of_IN a_DT tugboat_NN that_WDT comes_VBZ across_IN a_DT deserted_VBN russian_JJ tech_NN ship_NN that_WDT has_VBZ a_DT strangeness_NN to_TO it_PRP when_WRB they_PRP kick_VBP the_DT power_NN back_RB on_RB little_RB do_VBP they_PRP know_VB the_DT power_NN within_IN ..._: going_VBG for_IN the_DT gore_VB and_CC bringing_VBG on_IN a_DT few_JJ action_NN sequences_NNS

Happy bastard
Quick movie
Russian tech
Few action

Figure 1.4 The output of POS (Das & Chandra Balabantaray, 2014)

output of POS is illustrated in figure 1.4.

1.3.6 REMOVAL OF PUNCTUATION AND SYMBOLS

Punctuations are like the exclamation, question, and stop marks. They are used to divide the text into sentences, paragraphs, and phrases (Etaiwi & Naymat, 2017). The presence of punctuation marks in the text is a sign of the presence of emotion as they can be used to emphasize emotions. When the sentiment analysis task depends on the occurrence frequencies of words and phrases, removing punctuations affects the results. Charalampous & B suggests to consider only three punctuation signs that are commonly used to express emotions: e the exclamation, question, and stop marks. Then they look for the number of occurrences of them in a row, if more than one, the will be replaced by a tag mentioning the type of the punctuation marks. (Charalampous & B, 2017).

1.4 CONCLUSION

In this chapter, we defined the sentiment analysis task and explained how we can structure the unstructured problem of sentiment analysis. Three main levels of sentiment analysis were explained in detail and possible methods to resolve each was studied as well as commonly used pre-processing techniques. In the next chapter, machine learning techniques and two common machine learning algorithms will be explained.

CHAPTER II

STATE OF THE ART ON MACHINE LEARNING

As mentioned previously, artificial Intelligence aims to simulate the human brain. The way the human brain works is that it analyses and compares the input data to the experiences that already exist in the memory, then it ends up with a decision or conclusion. The present data will be stored in the memory for future use (Mohammed et al., 2016).

Machines can perform specific tasks based on given instructions. They are not able to analyze previous experiences to make decisions. Machine learning is an application of artificial intelligence that allows computers to learn from data rather than through explicit programming (Langley & Carbonell, 1984). The task of machine learning is to enable computers to make decisions by analyzing the input data.

In the following, first of all, machine learning techniques are discussed and then some of the commonly used techniques are explained in more detail. After that, Naive Bayes and SVM algorithms that are widely used for classification problems are explained.

2.1 MACHINE LEARNING TECHNIQUES

There are many machine learning techniques available. For example learning problems like supervised Learning, unsupervised learning and reinforcement learning, hybrid learning problems like semi-supervised learning, self-supervised learning, and multi-instance learning, statistical inference like inductive learning, deductive inference, and transductive learning or learning techniques like multi-task learning, active learning, online learning, transfer learning and ensemble learning (Brownlee, 2019).

Depending on the available data and the nature of the problem, different machine learning techniques can be used. Three main machine learning techniques are described in figure 2.1 (Mohammed et al., 2016). In the following, some of these techniques are described.

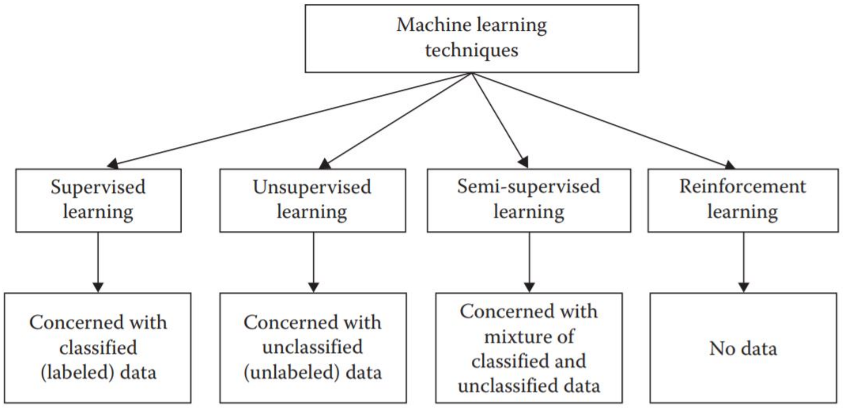

Machine learning techniques
Supervised learning
Unsupervised learning
Semi-supervised learning
Reinforcement learning
Concerned with classified (labeled) data
Concerned with unclassified (unlabeled) data
Concerned with mixture of classified and unclassified data
No data

Figure 2.1 Four machine learning techniques (Mohammed et al., 2016)

In supervised learning, as the most common machine learning techniques, the relationship between the input and output is used to learn a mapping function that best maps the input to the output (Liu & Wu, 2012). It is performed by using the values of paired input-output samples. As the output can be considered as the label of the input, so the samples of paired input and outputs can be called labeled data. Then the trained mapping function, which is called a classifier, can be used in the future to determine the output based on a given input. This technique works well with classification problems. To get the most accurate results, it is important to use the labeled data from the same domain of the problem. One of the disadvantages of supervised learning is providing the labeled data (Liu & Wu, 2012). Figure 2.2 shows the process of supervised machine learning.

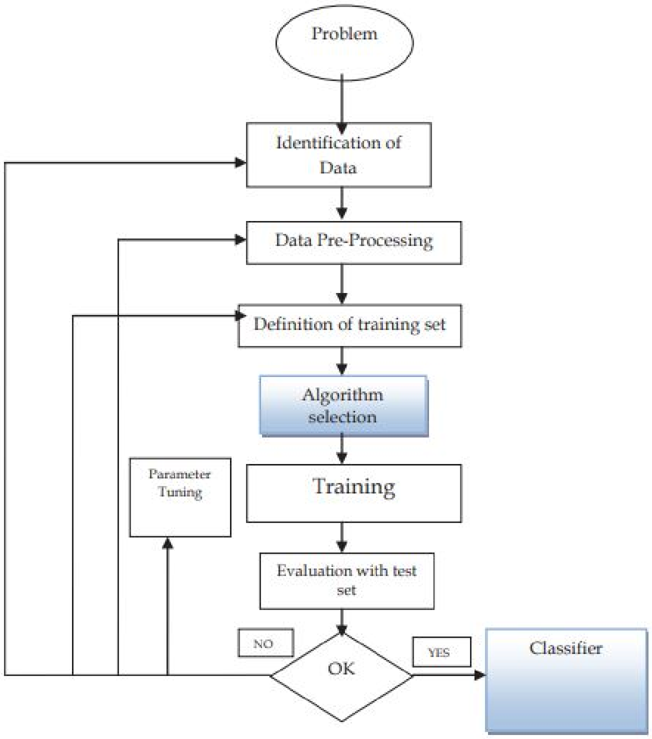


Figure 2.2 Supervised machine learning (Souza et al., 2013)

In unsupervised learning, by contrast with supervised learning, there is no labeled data to train the mapping function. In this technique, algorithms discover patterns and structures in the data to learn themselves (Brownlee, 2016). Some of the unsupervised techniques work based on the rewards system, they seek to make decisions that result in maximum reward (Souza et al., 2013). They are useful in decision-making problems. Other unsupervised techniques focus on finding similarities in the data. These techniques can be applied to clustering problems. Unsupervised learning is much harder than supervised learning and more time consuming as it works based on trial and error.

Semi-supervised techniques are used when there is a small amount of labeled data and a big amount of unlabeled data. They profit from both supervised and unsupervised techniques. Supervised learning techniques use the existing training data to label a part of unlabeled data. Then the results of the previous phase (labeled data) will be considered as the new training data set to train the algorithm again. Then the trained classifier will be applied to another part of data that hasn't been used in the training phase (Brownlee, 2016). This way the classifier enriches the training dataset by itself. In another approach, unsupervised techniques are used to find the structure in the data which will be used to label the output data. Generated labeled data will be used to train a supervised algorithm, and in the end, the supervised algorithm will be used to classify the unlabeled data.

Korovkinas & Garsva showed that SVM Naive Bayes performs well in sentiment classification problems (Korovkinas & Garšva, 2018). In the following, the Naive Bayes algorithm and SVM are explained.

2.2 NAIVE BAYES

Naive Bayes is one of the simplest and commonly used machine learning algorithms. It is a probabilistic classifier that works based on Bayes theorem. Baye's theorem can be stated as follows in equation (Jurafsky & Martin, 2019) 2.1.

$$P(\mathbf{A}|\mathbf{B}) = \frac{P(\mathbf{B}|\mathbf{A}) \times P(\mathbf{A})}{P(\mathbf{B})} \tag{2.1}$$

In the following, the algorithm is explained in the context of document sentiment classification. Given a document, the task is to classify it as positive or negative. To do so Naive Bayes algorithm is used to calculate the probability of the document being negative and the probability of the document being positive, then the highest value defines the class of the document. This can be formulated as the following.

first step is to calculate these two propabilities using equation 2.2 and 2.3:

1)

$$P(\mathbf{Positive}|\mathbf{Document}) = \frac{P(\mathbf{Document}|\mathbf{Positive}) \times P(\mathbf{Positive})}{P(\mathbf{Document})} \tag{2.2}$$

2)

$$P(\mathbf{Negative}|\mathbf{Document}) = \frac{P(\mathbf{Document}|\mathbf{Negative}) \times P(\mathbf{Negative})}{P(\mathbf{Document})} \tag{2.3}$$

2.2 and 2.3 are calculated as following:

if we consider only two classes (negative and positive), then P(**Negative**) = 0.5 and P(**Positive**)=0.5 and P(**Document**) is always equal to 1. Then the task is to calculate the P(**Document**|**Positive**) and P(**Document**|**Negative**). Considering that a document is a bag of n words($\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n$), the values will be calculated as following:

$$P(\mathbf{Document}|\mathbf{Positive}) = P(\mathbf{w}_1|\mathbf{Positive}) \times \ldots \times P(\mathbf{w}_n|\mathbf{Positive}) \tag{2.4}$$

and

$$P(\mathbf{Document}|\mathbf{Negative}) = P(\mathbf{w}_1|\mathbf{Negative}) \times \ldots \times P(\mathbf{w}_n|\mathbf{Negative}) \quad (2.5)$$

P($\mathbf{w}_1$|**Positive**) is number of occurrences of $\mathbf{w}_1$ in the positive documents divided by total number of all the words in the positive documents. P($\mathbf{w}_1$|**Negative**) can be calculated in the same way. Now by having the values of all the variables in the equations 2.2 and 2.3, the polarity of the given document can be defined. If P(**Negative**|**Document**) is greater than P(**Positive**|**Document**) then the document is classified as negative, if P(**Negative**|**Document**) is equal to P(**Positive**|**Document**) then the document is neutral and if P(**Negative**|**Document**) is less than P(**Positive**|**Document**), the document is classified as positive.

2.3 SUPPORT VECTOR MACHINE

The support vector machine(SVM) is a linear classifier. Linear classifiers aim to find a hyperplane that can split the data into two classes (see figure 2.3). SVM is one of the practical algorithms which is used widely in sentiment analysis and concentrates on finding the best hyperplane to separate the classes (Verma, 2019).

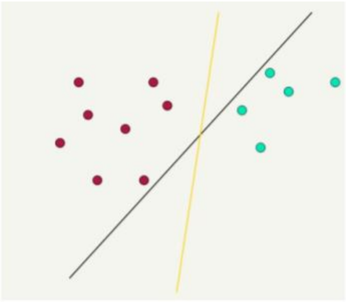


Figure 2.3 Linear classification (Berwick, 2003)

As shown in figure 2.4, an infinite number of hyperplanes exist that can separate the data points into two classes. SVM finds the optimal one by maximizing the margin around the hyperplane (Berwick, 2003). So SVM is also an optimization problem that can be solved by special techniques (Verma, 2019).

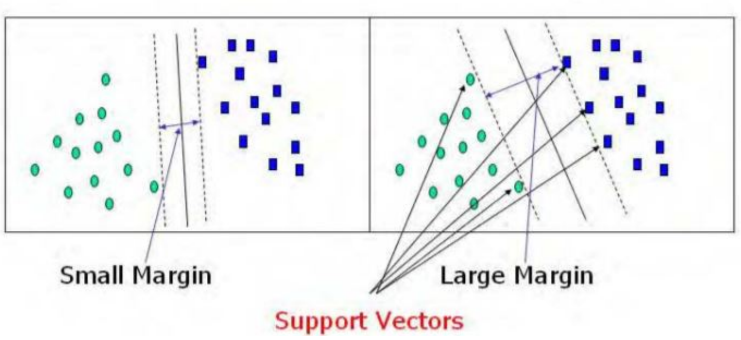


Figure 2.4 Support Vectors (Souza et al., 2013)

To find the maximum margin, the goal is to find the vectors that have the maximum distance to the hyperplane. These points are called support vectors. figure 2.4 is a visualization of SVM in two dimensions. Support vectors are the ones on the margin border.

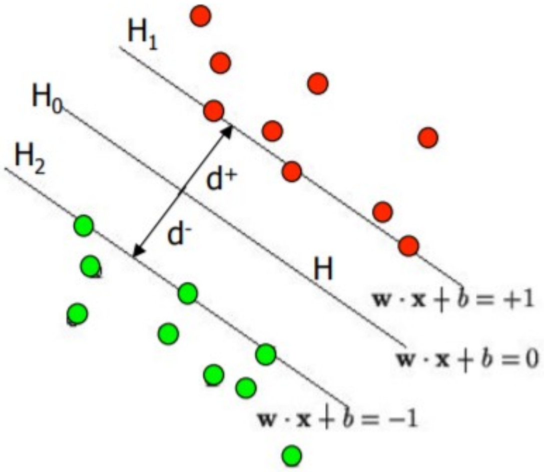

H_1
H_0
H_2
d^+
d^-
H
$\mathbf{w} \cdot \mathbf{x} + b = +1$
$\mathbf{w} \cdot \mathbf{x} + b = 0$
$\mathbf{w} \cdot \mathbf{x} + b = -1$

Figure 2.5 Support Vector machine (Berwick, 2003)

In the figure 2.5, "d+" is the shortest distance between the hyperplane H and the closest positive point and "d-" is the shortest distance between the hyperplane H and the closest negative point. The goal of SVM is to find the hyperplane H for which "d" is maximized (Berwick, 2003).

2.4 CONCLUSION

In this chapter, first of all, supervised and unsupervised machine learning techniques as the commonly used machine learning techniques for the classification problem were explained. Then Naive Bayes and SVM algorithms that are widely used for the classification problems are described. In the next chapter, we will use two sentiment analysis approaches to perform the sentiment analysis task to find out which one should be used in the base-line, then Weka will be used to find the combination of machine learning algorithms and pre-processing tools to be considered in the baseline. After that, we will look for enhancements to improve the performance of the baseline.

CHAPTER III

EXPERIMENTATION

Sentiment analysis, or opinion mining, consists of extracting the opinions of individuals according to certain objects in a text in natural language. With the growth of social media, sentiment analysis has become one of the most active research areas in natural language processing (Liu, 2015).

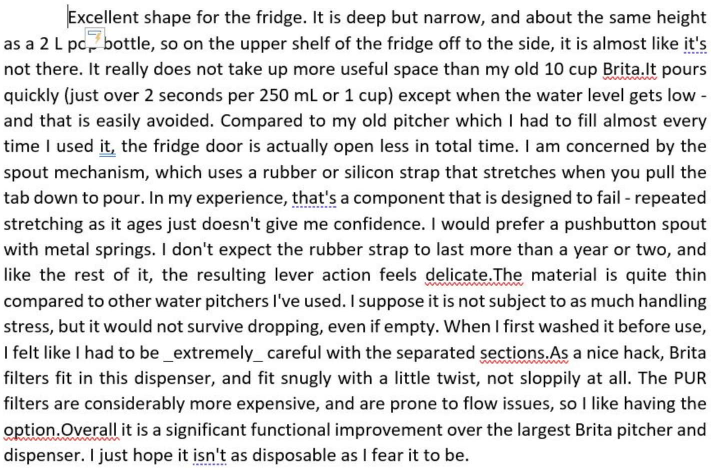
The sentiment analysis problem is a Natural Language Processing (NLP) problem, which can be also considered as a text classification problem. Different techniques can be used to solve it such as the lexicon-based, machine learning-based, or linguistic approaches (Taboada et al., 2011).

In this thesis, a lexicon-based and a machine learning-based approach were implemented. The lexicon-based approach is used to classify the POS labeled words into a positive or negative class. The machine learning-based approach was used to implement an online sentiment analysis tool. This tool can classify a text entered by the user as well as the ability to receive a URL of a YouTube video and extract all the comments of that video and then classify them.

In this chapter, we first explain the lexicon-based approach. Then we explain the method used to extract the comments of a YouTube video. Weka that is a data mining tool will be introduced. In the end, the machine learning approach that is used to implement the online sentiment analysis tool will be described. In this chapter to explain the Youtube API, we relied on the documentation provided by Google for YouTube data API.

3.1 CASE STUDY

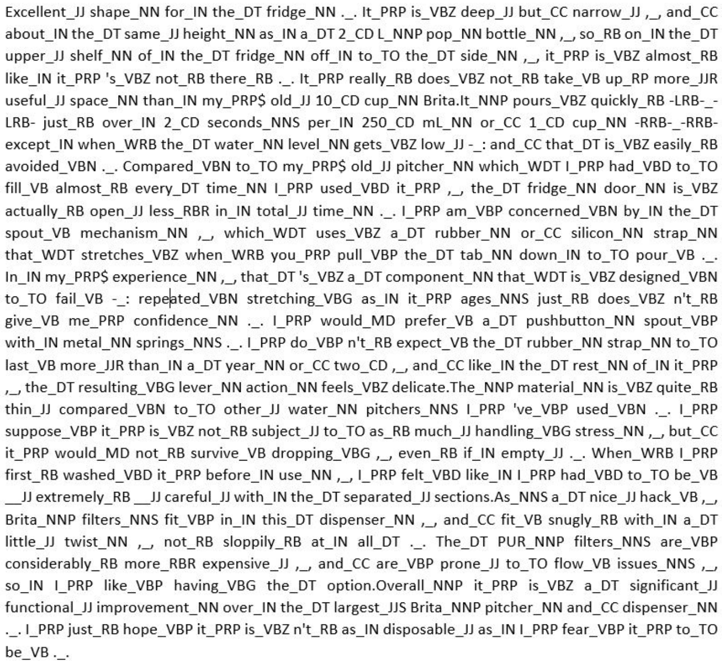
This thesis consists of reviewing two sentiment analysis approaches to do the document-level sentiment analysis. The first one is classifying text by using a lexicon-based approach and the second one is performing sentiment analysis using a machine learning approach.



Excellent shape for the fridge. It is deep but narrow, and about the same height as a 2 L pop bottle, so on the upper shelf of the fridge off to the side, it is almost like it's not there. It really does not take up more useful space than my old 10 cup Brita.It pours quickly (just over 2 seconds per 250 mL or 1 cup) except when the water level gets low - and that is easily avoided. Compared to my old pitcher which I had to fill almost every time I used it, the fridge door is actually open less in total time. I am concerned by the spout mechanism, which uses a rubber or silicon strap that stretches when you pull the tab down to pour. In my experience, that's a component that is designed to fail - repeated stretching as it ages just doesn't give me confidence. I would prefer a pushbutton spout with metal springs. I don't expect the rubber strap to last more than a year or two, and like the rest of it, the resulting lever action feels delicate.The material is quite thin compared to other water pitchers I've used. I suppose it is not subject to as much handling stress, but it would not survive dropping, even if empty. When I first washed it before use, I felt like I had to be _extremely_ careful with the separated sections.As a nice hack, Brita filters fit in this dispenser, and fit snugly with a little twist, not sloppily at all. The PUR filters are considerably more expensive, and are prone to flow issues, so I like having the option.Overall it is a significant functional improvement over the largest Brita pitcher and dispenser. I just hope it isn't as disposable as I fear it to be.

Figure 3.1 Example of input text

In the lexicon-based approach, a POS tagger is used to do the part of speech tagging on a given text as shown in figure 3.1. The result of this step will be as shown in figure 3.2. The POS tagged text will be the input and the output will be the polarity of the document. The issues to be addressed are to detect the lexicon words in the text and then to count the number of positive and negative words to classify the text.



```
Excellent_JJ shape_NN for_IN the_DT fridge_NN ._. It_PRP is_VBZ deep_JJ but_CC narrow_JJ ,_, and_CC
about_IN the_DT same_JJ height_NN as_IN a_DT 2_CD L_NNP pop_NN bottle_NN ,_, so_RB on_IN the_DT
upper_JJ shelf_NN of_IN the_DT fridge_NN off_IN to_TO the_DT side_NN ,_, it_PRP is_VBZ almost_RB
like_IN it_PRP 's_VBZ not_RB there_RB ._. It_PRP really_RB does_VBZ not_RB take_VB up_RP more_JJR
useful_JJ space_NN than_IN my_PRP$ old_JJ 10_CD cup_NN Brita.It_NNP pours_VBZ quickly_RB -LRB-_-
LRB- just_RB over_IN 2_CD seconds_NNS per_IN 250_CD mL_NN or_CC 1_CD cup_NN -RRB-_-RRB-
except_IN when_WRB the_DT water_NN level_NN gets_VBZ low_JJ -_: and_CC that_DT is_VBZ easily_RB
avoided_VBN ._. Compared_VBN to_TO my_PRP$ old_JJ pitcher_NN which_WDT I_PRP had_VBD to_TO
fill_VB almost_RB every_DT time_NN I_PRP used_VBD it_PRP ,_, the_DT fridge_NN door_NN is_VBZ
actually_RB open_JJ less_RBR in_IN total_JJ time_NN ._. I_PRP am_VBP concerned_VBN by_IN the_DT
spout_VB mechanism_NN ,_, which_WDT uses_VBZ a_DT rubber_NN or_CC silicon_NN strap_NN
that_WDT stretches_VBZ when_WRB you_PRP pull_VBP the_DT tab_NN down_IN to_TO pour_VB ._.
In_IN my_PRP$ experience_NN ,_, that_DT 's_VBZ a_DT component_NN that_WDT is_VBZ designed_VBN
to_TO fail_VB -_: repeated_VBN stretching_VBG as_IN it_PRP ages_NNS just_RB does_VBZ n't_RB
give_VB me_PRP confidence_NN ._. I_PRP would_MD prefer_VB a_DT pushbutton_NN spout_VBP
with_IN metal_NN springs_NNS ._. I_PRP do_VBP n't_RB expect_VB the_DT rubber_NN strap_NN to_TO
last_VB more_JJR than_IN a_DT year_NN or_CC two_CD ,_, and_CC like_IN the_DT rest_NN of_IN it_PRP
,_, the_DT resulting_VBG lever_NN action_NN feels_VBZ delicate.The_NNP material_NN is_VBZ quite_RB
thin_JJ compared_VBN to_TO other_JJ water_NN pitchers_NNS I_PRP 've_VBP used_VBN ._. I_PRP
suppose_VBP it_PRP is_VBZ not_RB subject_JJ to_TO as_RB much_JJ handling_VBG stress_NN ,_, but_CC
it_PRP would_MD not_RB survive_VB dropping_VBG ,_, even_RB if_IN empty_JJ ._. When_WRB I_PRP
first_RB washed_VBD it_PRP before_IN use_NN ,_, I_PRP felt_VBD like_IN I_PRP had_VBD to_TO be_VB
__JJ extremely_RB __JJ careful_JJ with_IN the_DT separated_JJ sections.As_NNS a_DT nice_JJ hack_VB ,_,
Brita_NNP filters_NNS fit_VBP in_IN this_DT dispenser_NN ,_, and_CC fit_VB snugly_RB with_IN a_DT
little_JJ twist_NN ,_, not_RB sloppily_RB at_IN all_DT ._. The_DT PUR_NNP filters_NNS are_VBP
considerably_RB more_RBR expensive_JJ ,_, and_CC are_VBP prone_JJ to_TO flow_VB issues_NNS ,_,
so_IN I_PRP like_VBP having_VBG the_DT option.Overall_NNP it_PRP is_VBZ a_DT significant_JJ
functional_JJ improvement_NN over_IN the_DT largest_JJS Brita_NNP pitcher_NN and_CC dispenser_NN
._. I_PRP just_RB hope_VBP it_PRP is_VBZ n't_RB as_IN disposable_JJ as_IN I_PRP fear_VBP it_PRP to_TO
be_VB ._.
```

Figure 3.2 Example of POS tagged text

In the machine learning approach, the input can be a normal text entered by the user, or a file containing all the comments of a YouTube video that are extracted from the URL of the video. So the issue to be addressed in this approach is to extract all the comments of a video on YouTube by having its URL, then consider each comment as a document and classify it.

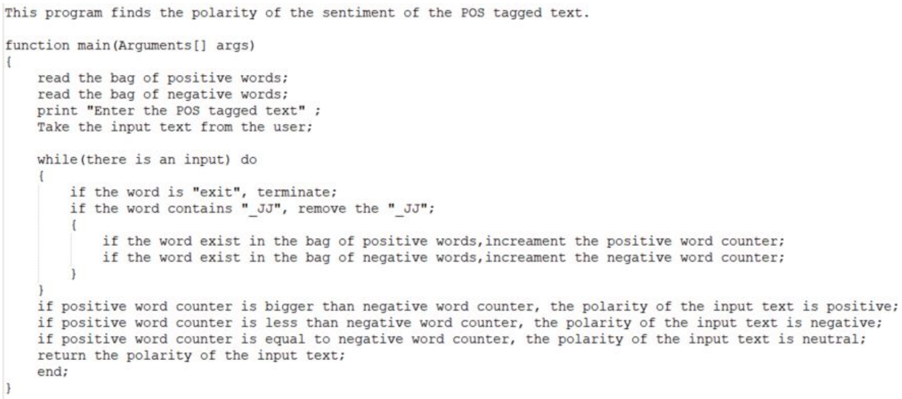
3.2 LEXICON-BASED APPROACH

The lexicon-based approach assumes that the contextual sentiment orientation of a document is the summary of the sentiment orientation of each word or phrase.

In this approach, a set of negative words and a set of positive words (Hu & Liu, 2004) are used as the lexicon words. As we mentioned before, phrases that contain sentiment are subjective. Adjectives are important indicators of subjectivity.

For this reason in this work, just the adjectives are considered to determine the sentiment of the document. In Penn Treebank (Mititelu, 2007), adjectives are tagged by adding the postfix "_JJ" to the word.

The application is written in Java. It reads the two sets of positive and negative lexicon words as well as the POS tagged text as the input and saves them in data structures. Then it loops over the words in the input text one by one and looks for a word with a postfix "_JJ", if one is found, then the positive and negative sets are searched to see if the word exists in any of them, if it is found in the positive set, the positive counter is increased and if it is found in the negative set then the negative counter is increased. In the end, the sentiment of the document is defined based on these counters, if the positive counter is bigger than the negative counter, then the document is positive and visa versa. If the negative counter is equal to the positive counter, then the sentiment of the document is neutral. The Pseudo-code of the application is shown in the figure 3.3.



```
This program finds the polarity of the sentiment of the POS tagged text.

function main(Arguments[] args)
{
    read the bag of positive words;
    read the bag of negative words;
    print "Enter the POS tagged text" ;
    Take the input text from the user;

    while(there is an input) do
    {
        if the word is "exit", terminate;
        if the word contains "_JJ", remove the "_JJ";
        {
            if the word exist in the bag of positive words,increament the positive word counter;
            if the word exist in the bag of negative words,increament the negative word counter;
        }
    }
    if positive word counter is bigger than negative word counter, the polarity of the input text is positive;
    if positive word counter is less than negative word counter, the polarity of the input text is negative;
    if positive word counter is equal to negative word counter, the polarity of the input text is neutral;
    return the polarity of the input text;
    end;
}
```

Figure 3.3 Pseudo-code of the lexicon-based

This is the simplest way of doing sentiment analysis. The results are not accurate as of the entities and their aspects are not considered.

3.3 MACHINE LEARNING-BASED APPROACH

In this approach, an online sentiment analyzer tool is implemented. The tool uses machine learning algorithms to determine the sentiment of the document. The user has the option to define the domain of the text by selecting a dataset within a list of available datasets to train the classifier as well as selecting pre-processing techniques to apply to the input data. To get more accurate results, the input text should be in the same domain as the training dataset.

Two types of inputs are considered, a text typed by the user or the URL of a YouTube video. In the first one, the input text is considered a single document to be classified. In the second one, the URL is used to find the video, then all the comments of that video are extracted and stored in a file, Each comment will be considered as a document and classified as negative or positive.

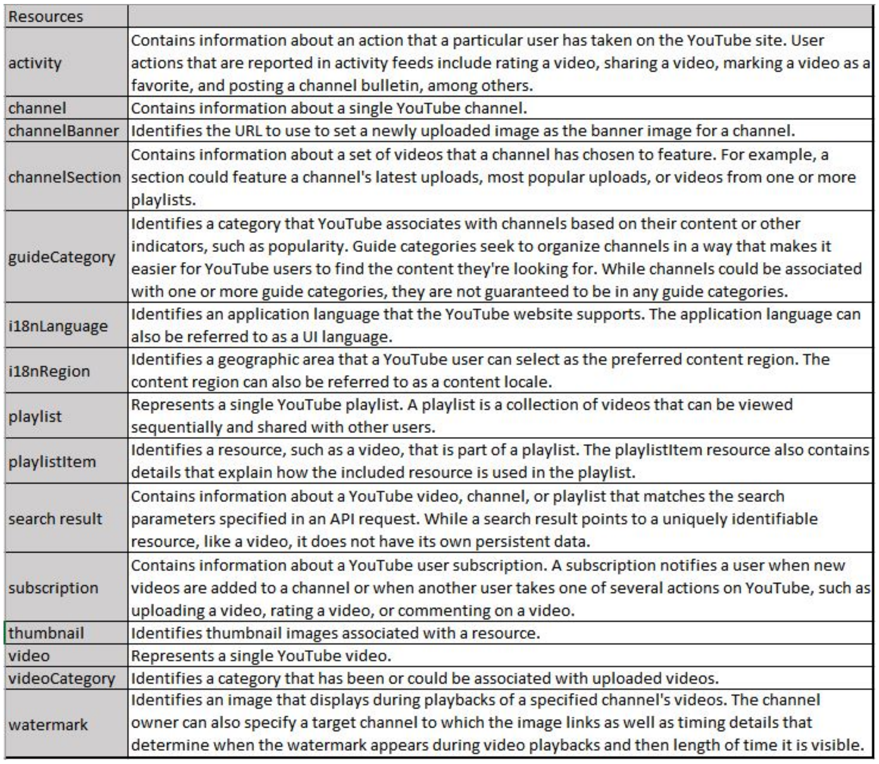
Weka which is an open-source tool is used to perform the pre-processing of data as well as the classification of the documents. Many machine learning algorithms are available in Weka. In this study, we first used Weka to apply the combination of machine learning algorithms and pre-processing techniques on data to see which combination gives the best results. Then the algorithm with the highest precision is considered to implement the online sentiment analyzer tool.

To extract the comment of a YouTube video using its URL, two approaches could be used, using the BeautifullSOAP Python library or using the YouTube WebAPI. We used the second approach as the rest of the tool was implemented by Java and YouTube web API was also available in Java. To implement web services on the server-side, Java, and to implement the user-end side, PHP is used.

In conclusion, to implement the online sentiment analyzer tool, four issues need to be addressed: 1) extracting comments of a YouTube video by having the URL (in case the input is a URL), 2) finding the dataset to train the algorithm, 3) finding the combination of pre-processing and machine learning algorithm available in Weka with the highest accuracy, 4) classifying the input using the selected machine learning algorithm. In the following, we will explain them in more details.

3.3.1 YOUTUBE DATA API

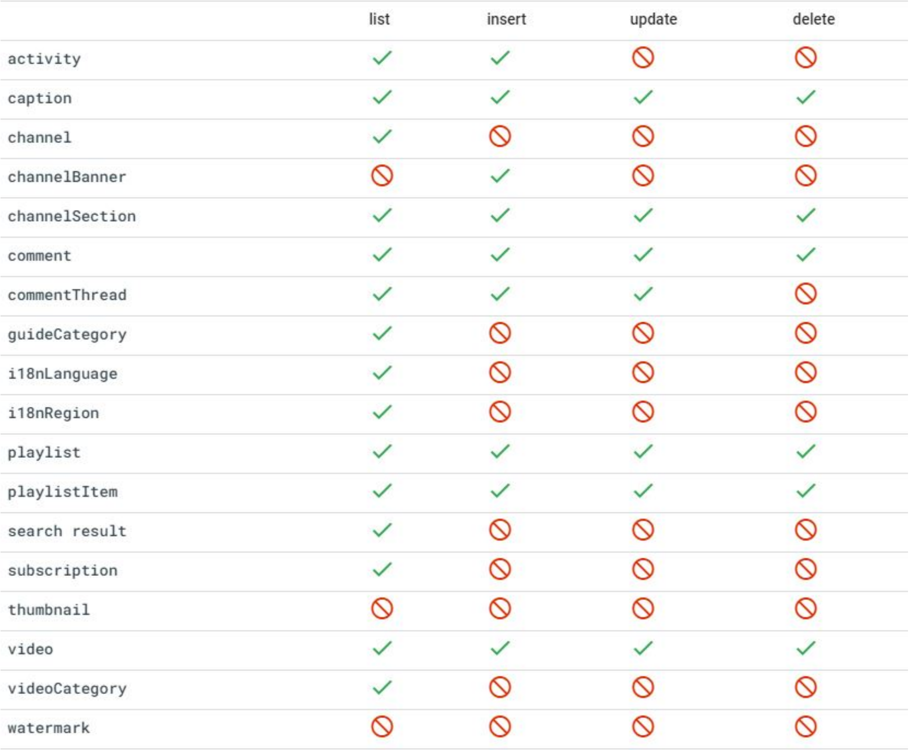
The first issue to be addressed is the extraction of comments related to a Youtube video. The YouTube Data API, which is a service provided by Google is used to perform this task. It allows developers to interact with YouTube within their applications.



Resources	
activity	Contains information about an action that a particular user has taken on the YouTube site. User actions that are reported in activity feeds include rating a video, sharing a video, marking a video as a favorite, and posting a channel bulletin, among others.
channel	Contains information about a single YouTube channel.
channelBanner	Identifies the URL to use to set a newly uploaded image as the banner image for a channel.
channelSection	Contains information about a set of videos that a channel has chosen to feature. For example, a section could feature a channel's latest uploads, most popular uploads, or videos from one or more playlists.
guideCategory	Identifies a category that YouTube associates with channels based on their content or other indicators, such as popularity. Guide categories seek to organize channels in a way that makes it easier for YouTube users to find the content they're looking for. While channels could be associated with one or more guide categories, they are not guaranteed to be in any guide categories.
i18nLanguage	Identifies an application language that the YouTube website supports. The application language can also be referred to as a UI language.
i18nRegion	Identifies a geographic area that a YouTube user can select as the preferred content region. The content region can also be referred to as a content locale.
playlist	Represents a single YouTube playlist. A playlist is a collection of videos that can be viewed sequentially and shared with other users.
playlistItem	Identifies a resource, such as a video, that is part of a playlist. The playlistItem resource also contains details that explain how the included resource is used in the playlist.
search result	Contains information about a YouTube video, channel, or playlist that matches the search parameters specified in an API request. While a search result points to a uniquely identifiable resource, like a video, it does not have its own persistent data.
subscription	Contains information about a YouTube user subscription. A subscription notifies a user when new videos are added to a channel or when another user takes one of several actions on YouTube, such as uploading a video, rating a video, or commenting on a video.
thumbnail	Identifies thumbnail images associated with a resource.
video	Represents a single YouTube video.
videoCategory	Identifies a category that has been or could be associated with uploaded videos.
watermark	Identifies an image that displays during playbacks of a specified channel's videos. The channel owner can also specify a target channel to which the image links as well as timing details that determine when the watermark appears during video playbacks and then length of time it is visible.

Figure 3.4 List of YouTube API resources (YouTube, 2020)

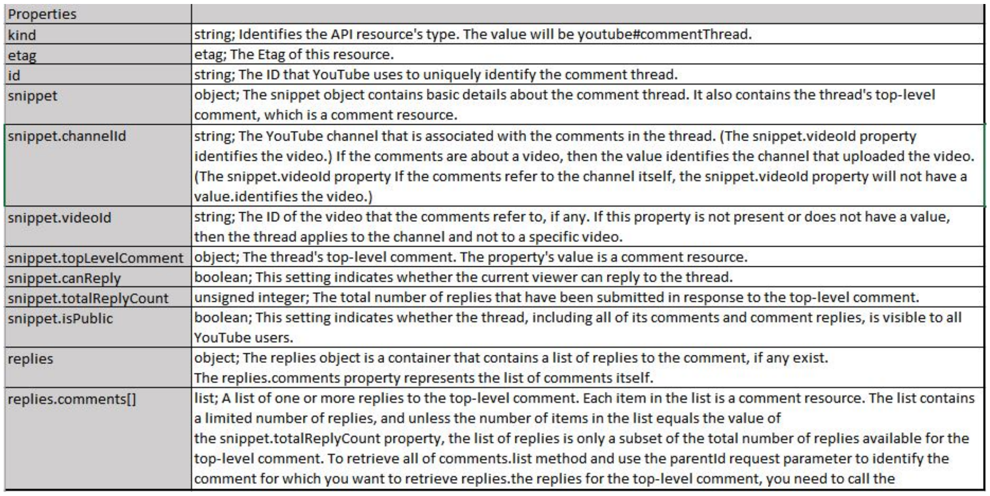
Youtube API allows users to access the data entities with unique identifiers, called resource types. Figure 3.4 is the list of some of the resource types. (YouTube, 2020). For example, the resource "video" is the representation of a YouTube video.



	list	insert	update	delete
activity	✓	✓	⊘	⊘
caption	✓	✓	✓	✓
channel	✓	⊘	⊘	⊘
channelBanner	⊘	✓	⊘	⊘
channelSection	✓	✓	✓	✓
comment	✓	✓	✓	✓
commentThread	✓	✓	✓	⊘
guideCategory	✓	⊘	⊘	⊘
i18nLanguage	✓	⊘	⊘	⊘
i18nRegion	✓	⊘	⊘	⊘
playlist	✓	✓	✓	✓
playlistItem	✓	✓	✓	✓
search result	✓	⊘	⊘	⊘
subscription	✓	⊘	⊘	⊘
thumbnail	⊘	⊘	⊘	⊘
video	✓	✓	✓	✓
videoCategory	✓	⊘	⊘	⊘
watermark	⊘	⊘	⊘	⊘

Figure 3.5 Supported CRUD actions for resource types (YouTube, 2020)

To extract data from Youtube API, the first step is obtaining an API key which is required for API version 3 and later. To obtain an API key, a google account should be used to create a project and register it, so the application can submit API requests. Youtube API supports the CRUD actions. But not all the actions are available for all the resource types. Figure 3.5 shows the supported action for each resource type. "CommentThreads" is the resource type that contains information related to comment threads. Figure 3.6, shows definition of properties of "commentThread" resource type.

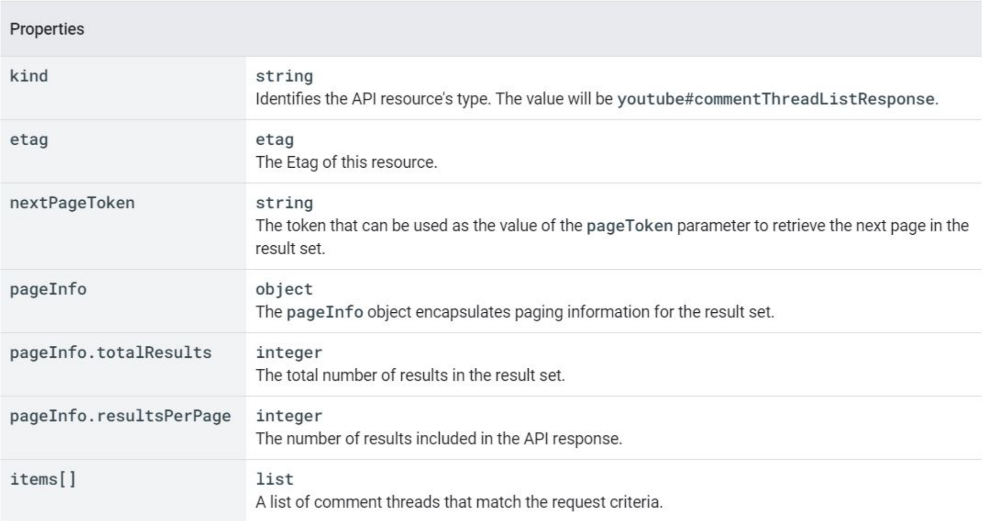


Properties	
kind	string; Identifies the API resource's type. The value will be youtube#commentThread.
etag	etag; The Etag of this resource.
id	string; The ID that YouTube uses to uniquely identify the comment thread.
snippet	object; The snippet object contains basic details about the comment thread. It also contains the thread's top-level comment, which is a comment resource.
snippet.channelId	string; The YouTube channel that is associated with the comments in the thread. (The snippet.videoId property identifies the video.) If the comments are about a video, then the value identifies the channel that uploaded the video. (The snippet.videoId property If the comments refer to the channel itself, the snippet.videoId property will not have a value.identifies the video.)
snippet.videoId	string; The ID of the video that the comments refer to, if any. If this property is not present or does not have a value, then the thread applies to the channel and not to a specific video.
snippet.topLevelComment	object; The thread's top-level comment. The property's value is a comment resource.
snippet.canReply	boolean; This setting indicates whether the current viewer can reply to the thread.
snippet.totalReplyCount	unsigned integer; The total number of replies that have been submitted in response to the top-level comment.
snippet.isPublic	boolean; This setting indicates whether the thread, including all of its comments and comment replies, is visible to all YouTube users.
replies	object; The replies object is a container that contains a list of replies to the comment, if any exist. The replies.comments property represents the list of comments itself.
replies.comments[]	list; A list of one or more replies to the top-level comment. Each item in the list is a comment resource. The list contains a limited number of replies, and unless the number of items in the list equals the value of the snippet.totalReplyCount property, the list of replies is only a subset of the total number of replies available for the top-level comment. To retrieve all of comments.list method and use the parentId request parameter to identify the comment for which you want to retrieve replies.the replies for the top-level comment, you need to call the

Figure 3.6 Properties of "commentThread" resource type (YouTube, 2020)

In most cases, we don't need to return all the information about a resource type. Youtube API allows users to select which information to be returned by using two request parameters, "part" and "fields" (YouTube, 2020). Based on the documentation of Youtube API, the "part" parameter is used to define one or more properties of the resource to be retrieved. "fields" limit the response to the specific properties of a requested part (YouTube, 2020). To get the list of comment threads, the method "list" for this resource type "CommentThreads" should be called. Two parameters to be set in the request are "part" and "videoId" (YouTube, 2020). To get the top-level comments, the parameter "part" should be set to "snippet" and to retrieve replies to top-level comments, the value "snippet, replies" should be assigned to "part". Each video on Youtube is identified by a unique id.

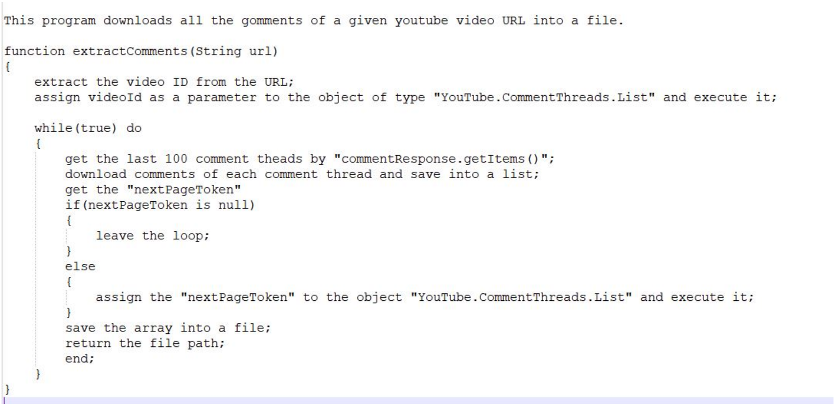
To get the comments of a video, the id of that video should be assigned to the "videoId" parameter. The number of items to be returned per request is defined by the "maxResults" parameter (YouTube, 2020). If the number of existing items is higher than the value of the "maxResults" then the response would include a property "nextPageToken" or "prevPageToken" or both. To get the rest of the items, the values of these properties should be assigned to the "pageToken" parameter in the next request. Figure 3.7 shows properties of a response of a "CommentThreads.list" request.



Properties	
kind	string Identifies the API resource's type. The value will be youtube#commentThreadListResponse.
etag	etag The Etag of this resource.
nextPageToken	string The token that can be used as the value of the pageToken parameter to retrieve the next page in the result set.
pageInfo	object The pageInfo object encapsulates paging information for the result set.
pageInfo.totalResults	integer The total number of results in the result set.
pageInfo.resultsPerPage	integer The number of results included in the API response.
items[]	list A list of comment threads that match the request criteria.

Figure 3.7 Properties of response of C̈ommentThreads.list" (YouTube, 2020)

In our experiment, the parameter "part" is set to "snippet" and "maxResults" is set to "100" which means in each request, 100 comment threads are returned. Then the video Id is extracted from the input URL and assigned to the parameter "videoId" as well as the API key. First time the request is executed, the top 100 comment threads are returned. To get the rest of the comments, in a loop the request should be executeed and each time the value of "NextPageToken" should be attached to the request. In each iteration, the comments are extracted and added to a list. AT the end, the list is saved into a file. The psudu-co of this algorithm is shown in the figure 3.8.

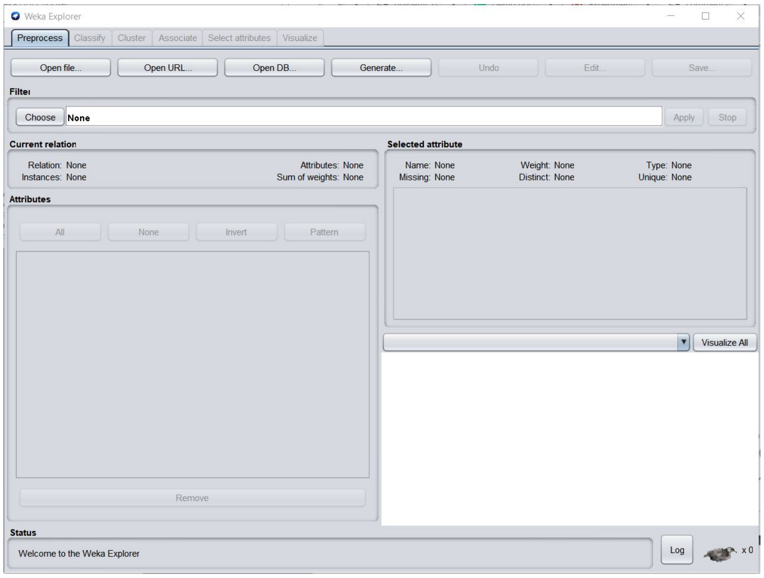


```
This program downloads all the gomments of a given youtube video URL into a file.

function extractComments(String url)
{
    extract the video ID from the URL;
    assign videoId as a parameter to the object of type "YouTube.CommentThreads.List" and execute it;

    while(true) do
    {
        get the last 100 comment theads by "commentResponse.getItems()";
        download comments of each comment thread and save into a list;
        get the "nextPageToken"
        if(nextPageToken is null)
        {
            leave the loop;
        }
        else
        {
            assign the "nextPageToken" to the object "YouTube.CommentThreads.List" and execute it;
        }
        save the array into a file;
        return the file path;
        end;
    }
}
```

Figure 3.8 YouTube Comment extraction Psudu-co

3.3.2 WEKA

Weka is a free Java-based tool developed at the University of Waikato, New Zealand consists of a collection of machine learning algorithms and data preprocessing techniques. It is licensed under the GNU General Public License (Frank et al., 2016).



Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Figure 3.9 Weka GUI

The functionality of Weka is available through its workbench (graphical user interface) shown in Figure 3.9, or the API. The problem with the workbench is that when a data set is loaded, it is kept in the main memory which makes Weka workbench not to be suitable for the large-size datasets (Frank et al., 2016).

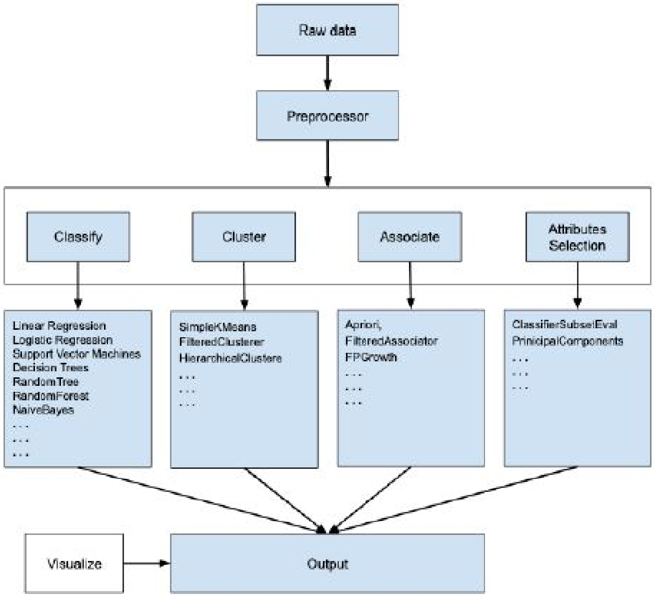


Figure 3.10 Main functional features of Weka (Wall et al., 2015)

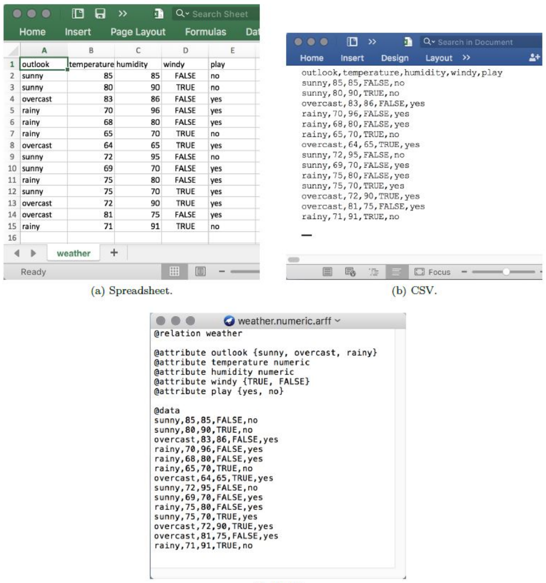
Weka covers all the necessary steps to perform data mining tasks as well as data pre-processing, training, and evaluation of learning algorithms and also a visual view of the results (Frank et al., 2016). Figure 3.10 displays the main functional features that Weka offers based on (Wall et al., 2015).

Input data can be a file or data from a database (Frank et al., 2016). Weka uses ARFF format by default. Attribute-Relation File Format(ARFF), is developed at the University of Waikato specifically for use with Weka.

ARFF represents a list of instances with their set of shared attributes (Bouckaert et al., 2018). It consists of two sections: Header and Data. Information such as the name of the relation and attributes and their types are stored in the header

(Bouckaert et al., 2018). The data section consists of a @data tag following the values of the attributes separated by a comma.

Spreadsheets and CSV are convertable to ARFF. Data from database can also be exported to CSV and then be converted from CSV to ARFF and then be used in Weka. CSV can also be directly imported via the Weka workbench. In Figure 3.11, an example of a Spreadsheet, CSV and ARFF file are displayed.



	A	B	C	D	E
1	outlook	temperature	humidity	windy	play
2	sunny	85	85	FALSE	no
3	sunny	80	90	TRUE	no
4	overcast	83	86	FALSE	yes
5	rainy	70	96	FALSE	yes
6	rainy	68	80	FALSE	yes
7	rainy	65	70	TRUE	no
8	overcast	64	65	TRUE	yes
9	sunny	72	95	FALSE	no
10	sunny	69	70	FALSE	yes
11	rainy	75	80	FALSE	yes
12	sunny	75	70	TRUE	yes
13	overcast	72	90	TRUE	yes
14	overcast	81	75	FALSE	yes
15	rainy	71	91	TRUE	no
16					

(a) Spreadsheet.

```
outlook,temperature,humidity,windy,play
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

(b) CSV.

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature numeric
@attribute humidity numeric
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

(c) ARFF.

Figure 3.11 Spreadsheet vs CSV vs ARFF (Frank et al., 2016)

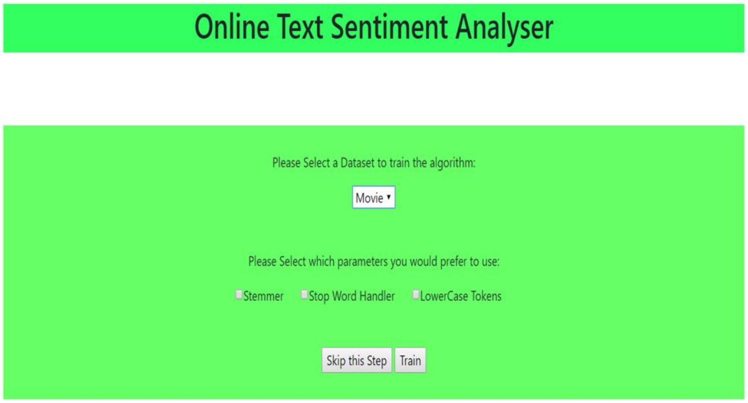
Spreadsheets and CSV are Convertable to ARFF. Data from the database can also be exported to CSV and then be converted from CSV to ARFF and then be used in Weka. CSV can also be directly imported via the Weka workbench. In Figure 3.11, an example of a Spreadsheet, CSV, and ARFF file are displayed.

After preparing the ARFF file, the next step is to load the dataset into Weka and apply the preprocessing techniques if needed, and then perform the data mining task on it.

3.3.3 ONLINE SENTIMENT ANALYSIS TOOL

The online sentiment analyzer tool is an online Java-based application that uses machine learning algorithms to perform the document sentiment classification task. It is designed based on a client-server architecture. On the client-side, in addition to the input, dataset and pre-processing techniques are selected. On the server-side, after the classifier is trained based on the selected dataset and selected pre-processing techniques are applied on the input, then the polarity of the document is determined by using machine learning techniques and in the end, results are sent to the client to be displayed.

The application is composed of two steps, the configuration step, and the analysis step. As shown in figure 3.12, in the configuration step, the domain of the document is defined by selecting the dataset.



Online Text Sentiment Analyser

Please Select a Dataset to train the algorithm:

Movie ▾

Please Select which parameters you would prefer to use:

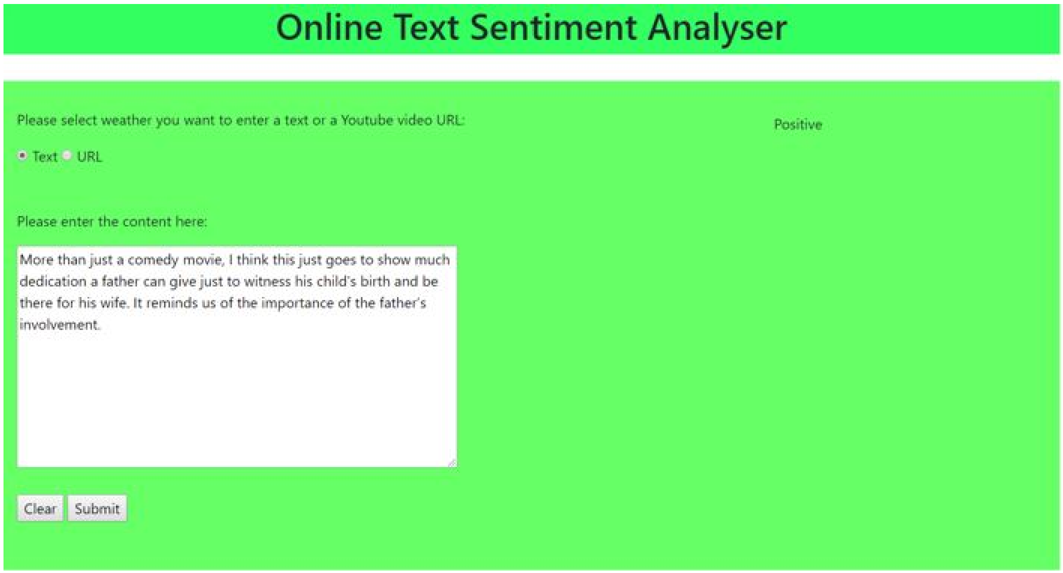
□ Stemmer □ Stop Word Handler □ LowerCase Tokens

Figure 3.12 Main page of the sentiment analysis tool

For this experimentation, only one dataset is added but any other dataset can be added later and the algorithm will be trained by the selected dataset. The dataset we used is "IMDB movie reviews for Sentiment Analysis", including 25,000 movie reviews.

To get more accurate results, the input would better be a movie review or URL of a movie or its trailer (KIMURA et al., 2001). Pre-processing techniques available within this tool are stemmer, stop word handler, and lowerCase tokens. These techniques were explained in more detail in the previous chapter.

In the analysis step, two types of input are considered, a text typed by the user or the URL of a YouTube video. In the first type, the input text is considered a single document to be classified. The results of an input type "text" are displayed in figure 3.13.



Online Text Sentiment Analyser

Please select weather you want to enter a text or a Youtube video URL:

● Text ○ URL

Please enter the content here:

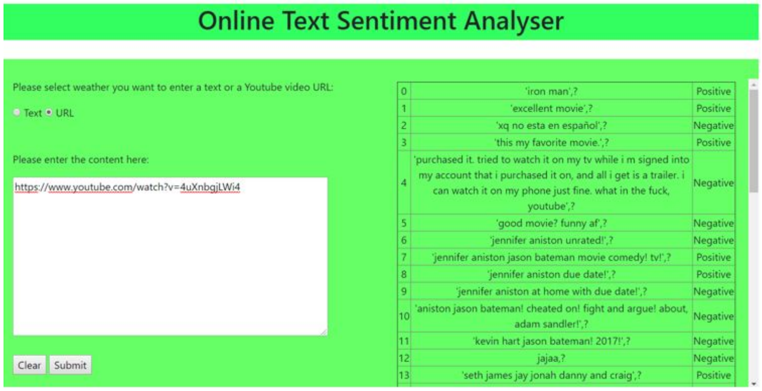
More than just a comedy movie, I think this just goes to show much dedication a father can give just to witness his child's birth and be there for his wife. It reminds us of the importance of the father's involvement.

Positive

Figure 3.13 Results of the sentiment analysis tool - Text Input

In the second type, the URL is used to find the video, then all the comments of that video are extracted and stored in a file, Each comment will be considered as a document and classified as negative or positive. The results of an input type "URL" is displayed in figure 3.14.

Weka is used to perform the pre-processing of data as well as the classification of the documents. Many machine learning algorithms are available in Weka. In this



Online Text Sentiment Analyser

Please select weather you want to enter a text or a Youtube video URL:

○ Text ● URL

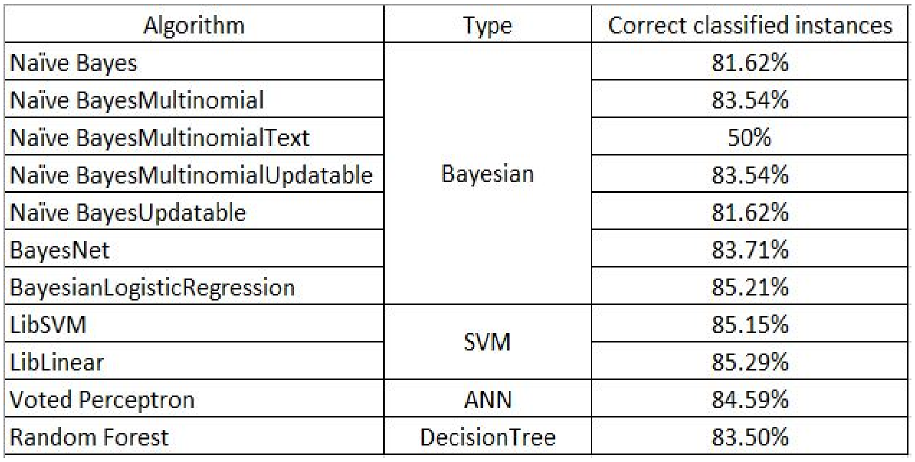
Please enter the content here:

https://www.youtube.com/watch?v=4uXnbgjLWi4

0	'iron man',?	Positive
1	'excellent movie',?	Positive
2	'xq no esta en español',?	Negative
3	'this my favorite movie.',?	Positive
4	'purchased it. tried to watch it on my tv while i m signed into my account that i purchased it on, and all i get is a trailer. i can watch it on my phone just fine. what in the fuck, youtube',?	Negative
5	'good movie? funny af',?	Negative
6	'jennifer aniston unrated!',?	Negative
7	'jennifer aniston jason bateman movie comedy! tv!',?	Positive
8	'jennifer aniston due date!',?	Positive
9	'jennifer aniston at home with due date!',?	Negative
10	'aniston jason bateman! cheated on! fight and argue! about, adam sandler!',?	Negative
11	'kevin hart jason bateman! 2017!',?	Negative
12	jajaa,?	Negative
13	'seth james jay jonah danny and craig',?	Positive

Figure 3.14 Results of the sentiment analysis tool - URL Input

study, we first used Weka Workbench to find the combination of machine learning algorithms and pre-processing data that gives the best results. Then using the same pre-processing techniques, the algorithm with the highest precision is used to implement the online sentiment analyzer tool. Table 3.1 displays the results of this experiment.



Algorithm	Type	Correct classified instances
Naïve Bayes	Bayesian	81.62%
Naïve BayesMultinomial		83.54%
Naïve BayesMultinomialText		50%
Naïve BayesMultinomialUpdatable		83.54%
Naïve BayesUpdatable		81.62%
BayesNet		83.71%
BayesianLogisticRegression		85.21%
LibSVM	SVM	85.15%
LibLinear		85.29%
Voted Perceptron	ANN	84.59%
Random Forest	DecisionTree	83.50%

Table 3.1 Results of algorithm comparison

As shown in the table 3.1, "LibLinear" has the highest accuracy and it was used in the experimentation.

3.4 CONCLUSION

In this chapter, we explained two main approaches to perform the sentiment analysis task. Because of a lot of limitations that the lexicon-based have, the machine learning-base approach was selected to be used in the base-line. Then, Weka workbench was used to apply pre-processing and machine learning algorithms on our data set to define the combination of these two features that provide the highest accuracy. The base-line was used to implement the online sentiment analysis tool.

CHAPTER IV

DISCUSSION OF RESULTS

In the previous chapter, we explained how we performed the sentiment analysis task on cultural product reviews, which is the objective of this thesis.

Two approaches were studied. The first one is the lexicon-based that outputs the polarity of a POS tagged text. It uses a bag of negative words and a bag of positive words to calculate the number of negative and the number of positive words to define the polarity of the text. We used Stanford POS Tagger to perform the part of speech tagging on the input text.

As explained earlier in chapter one, the phrases containing sentiment are mainly subjective. Subjective phrases usually contain adjectives and adverbs. So adjectives and adverbs are identifiers of sentiment. But they are not enough to perform a high accuracy sentiment analysis. For example, negation or sentiment diminishers and intensifiers can change the sentiment of an adjective. Sentiment diminishers and intensifiers are described in chapter one. As in this approach, sentiment analysis is done in the document level, sentiment targets are not considered. How complete the bag of negative words and positive words is, also affects the results of this task.

The second approach is the machine learning-based approach. In this approach, sentiment analysis is also done at the document level. Two types of input are accepted. A text or a URL of a YouTube video. Text input is considered as a document. For the URL, YouTube data API is used to extract the comments of that video. Then each comment is considered as a document to perform the task. We used the data set "IMDB movie reviews" to train the classifier.

Weka GUI is used to find the combination of data pre-processing and machine learning algorithms that gives the highest accuracy. Then using Weka API, the data pre-processing techniques are applied to the data and the algorithm is trained by the data set. The trained classifier is used to define the polarity of the document. In this approach, the results depend on the data set, algorithm, and data pre-processing selection.

Sentiment analysis is essentially an NLP task. Accordingly to get the highest accuracy it is better to consider the syntax and linguistic-based techniques with the selected approach, whether it is the lexicon-based or the machine learning-based. In data mining, the data pre-processing step plays an important role. In machine learning also it is important to choose an algorithm that gives the best results For sentiment analysis as a text classification problem, SVM and Bayes are frequently used in research. The data set which is used to train the classifier should also be in the same domain as the sentiment analysis task.

CONCLUSION

Nowadays companies and governments all using data mining techniques to extract information from the big amount of online data that is available in the form of reviews, comments, tweets, etc. This information can be used for example to make predictions in elections to find out the feedback of customers about a product or service. Sentiment analysis, or opinion mining, consists of extracting the opinions of individuals according to certain objects in a text in natural language.

Natural language processing is one of the most complicated and difficult fields. Sentiment analysis as a sub-domain of NLP is also a complex task. The level is complexity depends on several factors. One factor can be the type of the text For example tweets are short and they are focused on one sentiment target, they also have one sentiment holder by contrast forum discussions are more difficult as many opinion holders are talking about several sentiment targets and they also contain comparative opinions. Another factor can be the level of sentiment analysis. For example document level is easier than an aspect-based level.

One of the important factors is also the domain of the text. Based on the context, the sentiment analysis task might become more difficult. One of them is cultural products. On cultural product reviews, people usually don't express their feeling directly. They use figures of speech like sarcasm, language ambiguity, or allegories to express their sentiment. Extracting the sentiment from these types of text needs special techniques. Extracted sentiment can be used later to define if there is a correlation between the visibility of cultural products and the opinions expressed.

The objective of this thesis is to perform the sentiment analysis task on cultural product reviews. To reach this goal, two different approaches were studied and tested.

There are limitations to each approach. In the lexicon-based approach it is not enough to just rely on adjectives to define the polarity of the text. For example negation or sentiment diminishers and intensifiers are also vary important to define the sentiment but the are not considered. Sentiment diminishers and intensifiers are described in chapter one. In the machine learning-based approach, it is very important to use the training dataset in the same domain as the input text. But we cannot always find a labled dataset for the domain we want.

How successful the sentiment analysis task is, depends on many factors. To get the most accurate results, it is important to first define the domain and the level of sentiment analysis. Pre-processing also plays an important role in the quality of the results. Because of the nature of the problem which is an NLP problem, relying only on machine learning algorithms without considering the syntax and lexicon will not be enough to get the best results. For a complex domain like the cultural product reviews, as they contain special figures of speech, it is better to use a lexicon-based and linguistic-based approaches with a machine learning approach together to get the more accurate results.

BIBLIOGRAPHY

A., V. & Sonawane, S. (2016). Sentiment Analysis of Twitter Data: A Survey of Techniques. *International Journal of Computer Applications*, *139*(11), 5–15. `http://dx.doi.org/10.5120/ijca2016908625`

Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F. & Manicardi, S. (2016). A comparison between preprocessing techniques for sentiment analysis in Twitter. *CEUR Workshop Proceedings*, *1748*(December).

Aue, A. & Gamon, M. (2005). Customizing Sentiment Classifiers to New Domains: A Case Study. *Recent Advances in Natural Language Processing (RANLP)*, (January 2005).

Berwick, R. (2003). An Idiot's Guide to Support vector machines (SVMs): A New Generation of Learning Algorithms Key Ideas. pp. 1–28. Retrieved from `http://www.cs.ucf.edu/courses/cap6412/fall2009/papers/Berwick2003.pdf`

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A. & Scuse, D. (2018). WEKA Manual for Version 3-8-3. *The University of Waikato*, pp. 1–327. `http://dx.doi.org/10.2807/1560-7917.ES.2016.21.37.30341`

Brownlee, J. (2016). Master Machine Learning Algorithms Discover How They Work and Implement Them From Scratch. *Machine Learning Mastery With Python*, *1*(1), 11. Retrieved from `http://machinelearningmastery.com`

Brownlee, J. (2019). 14 Different Types of Learning in Machine Learning. Retrieved on 2020-08-27 from `https://machinelearningmastery.com/types-of-learning-in-machine-learning/`

Charalampous, A. & B, P. K. (2017). Research and Advanced Technology for Digital Libraries - 21st International Conference on Theory and Practice of Digital Libraries, {TPDL} 2017, Thessaloniki, Greece, September 18-21, 2017, Proceedings. *10450*(September), 181–192. `http://dx.doi.org/10.1007/978-3-319-67008-9`. Retrieved from `https://doi.org/10.1007/978-3-319-67008-9`

Das, O. & Chandra Balabantaray, R. (2014). Sentiment Analysis of Movie Reviews using POS tags and Term Frequencies. *International Journal of Computer Applications*, *96*(25), 36–41. http://dx.doi.org/10.5120/16952-7048

Dave, K., Lawrence, S. & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of the 12th International Conference on World Wide Web, WWW 2003*, pp. 519–528. http://dx.doi.org/10.1145/775152.775226

Davidov, D., Tsur, O. & Rappoport, A. (2010). Enhanced sentiment learning using twitter hashtags and smileys. *Coling 2010 - 23rd International Conference on Computational Linguistics, Proceedings of the Conference*, *2*, 241–249.

Du, J., Ling, C. X. & Zhou, Z. H. (2011). When does cotraining work in real data? *IEEE Transactions on Knowledge and Data Engineering*, *23*(5), 788–799. http://dx.doi.org/10.1109/TKDE.2010.158

Edosomwan, S., Prakasan, S., Kouame, D., Watson, J. & Seymour, T. (2011). The history of social media and its impact on business. *Journal of Applied Management and Entrepreneurship*, *16*, 79–91.

Etaiwi, W. & Naymat, G. (2017). The Impact of applying Different Preprocessing Steps on Review Spam Detection. *Procedia Computer Science*, *113*, 273–279. http://dx.doi.org/10.1016/j.procs.2017.08.368. Retrieved from http://dx.doi.org/10.1016/j.procs.2017.08.368

Frank, E., Hall, M. A. & Witten, I. H. (2016). The WEKA Workbench. Online Appendix. *Data Mining: Practical Machine Learning Tools and Techniques*, pp. 128. Retrieved from https://www.cs.waikato.ac.nz/ml/weka/citing.html

Hu, M. & Liu, B. (2004). Mining opinion features in customer reviews. *Proceedings of the National Conference on Artificial Intelligence*, pp. 755–760.

Jatav, V., Teja, R., Bharadwaj, S. & Srinivasan, V. (2017). Improving Part-of-Speech Tagging for NLP Pipelines. Retrieved from http://arxiv.org/abs/1708.00241

Javed, M. & Kamal, S. (2018). Normalization of unstructured and informal text in sentiment analysis. *International Journal of Advanced Computer Science and Applications*, *9*(10), 78–85.

http://dx.doi.org/10.14569/IJACSA.2018.091011

Jiang, L., Yu, M., Zhou, M., Liu, X. & Zhao, T. (2011). Target-dependent Twitter sentiment classification. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, *1*, 151–160.

Jurafsky, D. & Martin, J. (2019). Chapter 04: Naive bayes and sentiment classification. *Speech and Language Processing*, pp. 1024. Retrieved from https://web.stanford.edu/{~}jurafsky/slp3/6.pdf

Kamal, A., Abulaish, M. & Jahiruddin (2016). Sentiment Analysis and Ontology Engineering. *Studies in Computational Intelligence*, *639*(August 2016), 399–423. http://dx.doi.org/10.1007/978-3-319-30319-2. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84961589899{\&}partnerID=tZOtx3y1

Kennedy, A. & Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational Intelligence*, *22*(2), 110–125. http://dx.doi.org/10.1111/j.1467-8640.2006.00277.x

Khurana, D., Koli, A., Khatter, K. & Singh, S. (2017). Natural Language Processing: State of The Art, Current Trends and Challenges. (Figure 1). Retrieved from http://arxiv.org/abs/1708.05148

KIMURA, R., HAYASHI, H., TATSUKI, S. & TAMURA, K. (2001). Determinants and Timing of Housing Reconstruction Decisions by the Victims of the 1995 Hanshin-Awaji Earthquake Disaster : A 2001 Replication. *Journal of social safety science*, (3), 23–32. http://dx.doi.org/10.11314/jisss.3.23

Kok, J. N., Boers, E. J. W., Kosters, W. A., Putten, P. V. D. & Poel, M. (2010). Knowledge for sustainable development: an insight into the Encyclopedia of life support systems::::::Artificial Intelligence: Definition, Trends, Techniques and Cases. pp. 1096–1097.

Korovkinas, K. & Garšva, G. (2018). Selection of intelligent algorithms for sentiment classification method creation. *CEUR Workshop Proceedings*, *2145*, 152–157.

Langley, P. & Carbonell, J. G. (1984). *Approaches to machine learning*, volume 35. http://dx.doi.org/10.1002/asi.4630350509

Liu, B. (2015). *Sentiment Analysis.*

Liu, Q. & Wu, Y. (2012). Encyclopedia of the Sciences of Learning.

Encyclopedia of the Sciences of Learning, (January 2012). http://dx.doi.org/10.1007/978-1-4419-1428-6

M., Janyce Wiebet, Rebecca, T. (1998). P99-1032.pdf. pp. 246.

Mititelu, V. B. (2007). Treebanks, Building and Using Parsed Corpora. *Revue Roumaine de Linguistique*, *52*(1-2), 231–237. http://dx.doi.org/10.1007/978-94-010-0201-1

Mohammed, M., Khan, M. B. & Bashie, E. B. M. (2016). *Machine learning: Algorithms and applications*. No. July. http://dx.doi.org/10.1201/9781315371658

Nowak, A. Goals of this Course.

Palanisamy, P., Yadav, V. & Elchuri, H. (2013). Serendio: Simple and practical lexicon based approach to sentiment analysis. In **SEM 2013 - 2nd Joint Conference on Lexical and Computational Semantics*.

Pang, B., Lee, L. & Vaithyanathan, S. (1988). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), Philadelphia, July 2002, pp. 79-86. Association for Computational Linguistics.*

Shirsat, V. S., Jagdale, R. S. & Deshmukh, S. N. (2018). Document Level Sentiment Analysis from News Articles. *2017 International Conference on Computing, Communication, Control and Automation, ICCUBEA 2017*, pp. 1–4. http://dx.doi.org/10.1109/ICCUBEA.2017.8463638

Souza, G. M., Catuchi, T. A., Bertolli, S. C. & Soratto, R. P. (2013). Soybean under water deficit: physiological and Yield Responses. *A Comprehensive Survey of International Soybean Research - Genetics, Physiology, Agronomy and Nitrogen Relationships*, pp. 624. http://dx.doi.org/10.5772/711. Retrieved from www.intechopen.com

Taboada, M., Brooke, J., Tofiloski, M., Voll, K. & Stede, M. (2011). Lexicon-Based Methods for Sentiment Analysis DRAFT DRAFT DRAFT! *Computational Linguistics*, *37*(2), 267–307. http://dx.doi.org/10.1162/COLI_a_00049. Retrieved from http://www.sfu.ca/{~}mtaboada/docs/Taboada{_}etal{_}SO-CAL.pdf

Tsur, O., Davidov, D. & Rappoport, A. (2010). ICWSM - A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, (9), 162–169.

Verma, N. (2019). Study of Sentiment Classification Techniques. (April). http://dx.doi.org/10.13140/RG.2.2.19810.17603

Verspoor, K. M. & Cohen, K. B. (2013). Encyclopedia of Systems Biology. *Encyclopedia of Systems Biology*, (June 2018). http://dx.doi.org/10.1007/978-1-4419-9863-7

Wall, L., Extraction, P., Language, R., Os, M., Scripting, P., Shell, U., Point, T., Point, T. & Point, T. (2015). About the Tutorial Copyright & Disclaimer. pp. 2. http://dx.doi.org/10.1017/CBO9781107415324.004

Wan, X. (2008). Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. *EMNLP 2008 - 2008 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference: A Meeting of SIGDAT, a Special Interest Group of the ACL*, (October), 553–561. http://dx.doi.org/10.3115/1613715.1613783

Wan, X. (2009). Co-training for cross-lingual sentiment classification.

Yang, H., Si, L. & Callan, J. (2006). Knowledge transfer and opinion detection in the trec2006 blog track. *NIST Special Publication.*

YouTube (2020). YouTube Data API Overview | Google Developers. Retrieved on 2020-08-24 from https://developers.google.com/youtube/v3/getting-started