Predict User Mood According to Facebook Postings

Sabaragamu Ralalage Thilini Bhagya Samaraweera

169331R

Faculty of Information Technology

University of Moratuwa

2019

Predict User Mood According to Facebook Postings

Sabaragamu Ralalage Thilini Bhagya Samaraweera

169331R

Dissertation submitted to the Faculty of Information Technology, University of Moratuwa, Sri Lanka for partial fulfillment of the requirements of Master of Science in Information Technology

February 2019

Declaration

I hereby declare that this project report entitled “Predict User Mood According to Facebook postings” contains my own work and has not been submitted and will not be submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Name of Student: S.R.T.B.Samaraweera

Signature of Student:

Date:

Supervised by

Name of Supervisor:

Signature of Supervisor:

Date:

Dedication

I would like to dedicate my project “Predict User Mood According to Facebook Postings”,

To my project supervisor Mrs. GTI Karunarathne,

To the Instructors of Faculty of Information Technology at the University of Moratuwa who have supported me to make a success of this project.

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisor Ms.GTI Karunarathne for the continuous support of my research, for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of the research and writing of this thesis.

Besides my supervisor, I would like to thank Mr. SC Premarathne for his insightful comments, encouragement and for the support given me to widen my research from various perspectives.

Finally, my sincere thanks also go to Instructors of Faculty of Information Technology at the University of Moratuwa who has supported me to find resources and requirements for the project. Without their precious support, it would not be possible to conduct this research.

Abstract

We live in an era where communication is growing fast in the cyberspace. As a part of this people tend to be online 24 hours of a day and they write postings in social media. The interesting point is people may put a social mask to hide their feelings in the real world, but they reveal it on their post unknowingly. With this context discussion regarding opinion, mining has dominated research in recent years. Because of the ambiguity of human language, it is difficult to extract the sentiment precisely. Using appropriate machine learning approaches this paper explores to extract the polarity of the postings and predict the mood of a user accordingly. It could use to manage, communicate and collaborate with people more effectively and to manage own personal well-being and happiness. This study applies sentiment analysis for analyzing the hidden information present in the text on social media postings. It is an application of Natural Language Processing. In order to perform Sentiment analysis, need to identify the subjective and objective in the text. Because only the subjective text describes the sentimental information. Then the subjective text is preprocessing using various text preprocessing methods to extract the features. Text preprocessing may include stop word removal, stemming, tokenization, conjunction handling, and negation handling. After performing sentiment classification sentiment polarity can be extracted. To achieve this study uses the lexicon sentiment analysis process. Next, for sentiment classification, machine learning approaches can be used. It is an automatic classification technique and classification is performed using text features. The study uses supervised learning techniques. Using predefined emotion classes and sentiment polarity classifier is built accordingly.

Keywords - Natural language processing, Text preprocessing, Sentiment Analysis, Supervised learning.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Chapter Overview

This Chapter describes three sections. The first section mainly focuses on the background and the motivation for the study. In the second section Aim and the Objectives of the study is describing. Finally, the latter part of this chapter describes the solution for the study.

1.1 Background and Motivation

With the advancement of information technology, barriers to communication will no longer be an issue. Especially the internet is the most important invention that connects people to one place via social media. Which means now the entire world is treated as a global family. With this concept, people tend to share their personal life with social media such as Facebook, Twitter, and Google Plus+. Out of this Facebook is the most popular social media network site and it reaches 1.06 billion users globally [5].

Most of the social networking sits facilitates to feel like we are one step away from each other. Moreover, social networking postings come to picture as a new trend of being online. Most of the time people self-disclosure on social media using those postings. To put it another way, they reveal their thoughts, feelings, failures, fears, dreams, goals, aspirations openly or ambiguously [1]. If there is a way to identify the mood of a person by analyzing the social networking postings it could use to identify a person's behavior and helps to manage, communicate and collaborate with them more effectively. Get along better with people. As well as it could help to manage own personal well-being and happiness [2]. A methodology of indicating the sentiment or specific content in the e-mail will be a solution to this problem, The attempt of this research is to use Natural-Language-Processing (NLP) techniques to take out meaningful information from the body part of the e-mail and use Machine Learning Algorithms to ascertain the meaning of the text [1].

With this context, it is clear that social media sites are becoming popular day by day and it spread all over the world like a viral. Accordingly, researchers also pay more attention to doing researches to assess the impact of social media sites. Especially discussions on opinion mining have dominated research in recent years[2]. Researches

tend to study deeply on extracting hidden information (emotions/opinions) in the text that people post within an online mention. It could help to manage, communicate and collaborate with people more effectively. The study investigates to create a software model that predict the mood of a user according to their postings on Facebook. Moreover, the mood is predicted using eight emotions that identified by Robert Plutchik namely joy, sadness, anger, fear, trust, disgust, surprise and anticipation [4].

Consequently, identifying the mood will be benefited in so many ways. Family members, friends, partners can get an idea of how the person is going through when they do not talk much. Organizations can analyze the behavior of employees and can make decisions accordingly. Moreover, people can get along better with people.

There are several existing tools to categorize words as positive or negative opinion types such as General Inquirer, LIWC, WordNet, and Whissell Dictionary of Affective Language. But those tools never give 100 percent accurate results. They cannot extract the sentiment precisely. Because human language is more complex, and it is difficult to build a method to overcome all the grammatical tones, misspellings and cultural variations that occur in social media. It is even more difficult to train a method to identify the effect on tone.

1.3 Aim and Objectives

Aim: Aim is to perform feature extraction using social media text postings and predict the mood of a user.

Objectives:

- To Extract features from text postings by performing text preprocessing techniques.
- To find an appropriate sentiment classification approach to extract sentiment polarity.
- To develop a methodology to predict mood (predefined) using extracted sentiment polarity.
- To train the classifier to get highly accurate results.

1.4 Chapter Summary

In this chapter, it is discussed about the background and motivation. The background section discussed the existing data on the study and the problem statement that is identified as an issue with supporting data. The motivation section discussed the reason for this study. Finally, in the latter part with narrow details aim and objectives this study are discussed.

Chapter 2

Review of others' work

2.1 Chapter Introduction

In this section, it is discussed the previous researchers' findings regarding this study. And it mainly includes previous methods for text preprocessing and sentiment classification.

2.1 Text preprocessing techniques

To make predictions using online data, there are two things to be performed. In the beginning, the online text must be preprocessed. Most of the words in the online text have no use to extract the sentiment. It may contain noise and unformatted text. Use of the appropriate preprocessing technique is essential because it may lead to improving the performance and the accuracy of the classifier[3]. There are considerable techniques to perform text preprocessing and this section discussed prior researchers use of those techniques in their studies.

2.1.1 N-gram

N-gram model is a type of probabilistic language model. It is a basic level technique. The idea behind this concept is getting n number of consecutive words from a sentence to extract the idea of the text. It is to say getting a one word at a time is called as a unigram, getting two words at a time is called as bigram and likewise, word phrases can be extracted. The certain study indicated that it is impossible to capture the opinion and the relationship of words using this unigram concept[4]. But using a bigram, opinions can be extracted up to some level. This is a simple model that can be used for small experiments to scale up efficiency.

2.1.2 Tokenization

Tokenization or laxing is another method that is widely used. Here the given sentence will be divided into small pieces called tokens. In other words, tokens are sequences of words that grouped in to identify useful semantic. M. Aldarwishand and H.Ahmed used this method to predict depression levels using social media posts[5]. But here also a question arises is this a better way to preprocess a text. Finding the correct token is hard because of the ambiguity of languages. A prior study indicated some challenges in tokenization. According to that, some languages are space delimited and some are not.

And that study made another point by indicating the need for additional morphological and lexical information when performing tokenization on unsegmented languages[6].

2.1.3 Stemming

The process of removing various morphological forms of a word into a common representation referred to as stemming[6]. The basic level of an algorithm for stemming is Porter's algorithm. It is also called a suffix stripping algorithm and it can use to get the main morpheme out of the word. For the removal of other derivational affixes, the stemming algorithms such as Table Look up Approach, Successor Variety, N-Gram stemmers, and Affix Removal Stemmers can be used. A recent study has surveyed that stemming can reduce the index size by up to 50%. Also, it highlighted some errors in stemming as over stemming which leads to false positive and under stemming which leads to false negative[6].

2.1.4 Removing stop words

Removing stop words perform a higher accuracy in text preprocessing. Therefore, stop words of a text should be removed. When it comes to NLP stop words are referred to as the words that give no little information about a sentiment. These words are extremely common words such as pronouns (it/he/she), articles (a/an/the) and prepositions (in/at/by)[4].Some researchers indicated that sometimes removing stop words also may lead to inaccurate data[7].

It can be true when there is a need for getting the gender of the person that text is referring. Another study also indicated that removing the stop word is time-consuming. They confirmed that when it comes to paragraph sequence identification, removing stop words has an influence on the quantity and the quality of extracted rules whereas in case of sentence sequence identification there is no substantial influence on the quantity and the quality of extracted rules. According to their study, they illustrated the process by preprocessing the data on several levels such as paragraph level, sentence level, removing stop words and paragraph level, removing stop words and sentence level. By examining the results they argued that removal of stop words in sentence level may lead to decrease useful and trivial rules in the classification[8].

2.1.5 Negation handling

Another important task in prepossessing is the handling of negations. It referred to as handling the meaning of a word by confirming it is not get modified by the negation

modifiers, like the word “not”. When taking the words individually from the sentences like “I am not satisfied with your remarks”, the word “satisfied” gives a positive sentiment and the word “not” will convert it into a negative sentiment. Prior research has suggested a negation vector in their study. According to that negation, the vector is a vector having the same length as that of the tokens. When a text contains a negation word, then until the next punctuation mark all the tokens are considered as negation and for later purposes that negation vector assigned to 1[9]. Here this method seems correct but when to consider the sentences like “I am not falling apart anymore” gives a negative polarity though it has a positive polarity.

2.2 Sentiment classification approaches

Data classification is the process of organizing data into categories to get effective and efficient use of data. Moreover, it is a machine learning technique that can be used for data predictions. The process of classification can be divided as a two-step process namely model construction and model usage. According to a certain study, classification techniques can be divided into machine learning, lexicon based and hybrid approach[10]. Human annotations used in lexicon and machine learning is the most commonly used technique that carried automatic methods called ML algorithms. To acquire a better performance both ML and lexicon are combined as a hybrid approach[10].

2.2.1 Lexicon based approaches

According to a certain study, it is indicated that this approach used a small set of words called as “seeds” and in order to acquire a larger lexicon, online resources or synonym detection methods are used[10].

2.2.1.1 Dictionary-based approach

In this approach manually collected opinion sets are listed as a seed list and grown by using dictionaries and thesaurus for their synonyms and antonyms. The process continues until no new words are found and when new synonyms are found it is added to the seed list[10][4]. This is a rarely used approach since it is hard to find the context or domain-oriented opinion words.

2.2.1.2 Corpus-based approach

Here with the help of a corpus text, seed list is prepared and expanded[4]. In linguistics, a corpus or plural corpora is a large and structured set of text. Corpus can be monolingual or multilingual[11]. Thus this approach helps to solve the problem of

finding domain oriented text[10]. Some researchers have used this approach in their study to build a predictive model. They extracted corpus from Facebook status. Then 80% of the random corpus used as a training data set and 20% is used as the test data set. Finally, they have illustrated that the analysis of human languages can go beyond prediction by introducing a model of corpora with annotated well-being data. Moreover, they indicated that the model can demonstrate what leads to a good life[12].

2.2.2 Machine learning approaches

ML algorithms use syntactic or linguistic features to perform sentiment analysis. Supervised and unsupervised are two approaches to ML. When there is a large number of a labeled training document, supervised methods are used and when it is hard to find such documents, researchers can sustain using unsupervised methods[10].

2.2.2.1 Naïve Bayes Classifier (NB)

This is the most popular classifier among others. It uses Bayes theorem to calculate the most probable class label for the given text document by foreseeing the set of words in the document. In this model, it uses the BOWs feature extraction method to ignore the position of the word in the given document[10].

2.2.2.2 Maximum Entropy

This classifier uses vectors in the algorithms. Using encoding methods vectors are extracted by converting the labeled features of the document. Then for each feature weight is calculated using the encoded vector. Further that weight is combined and most likely label for a feature set is determined[10][4]. Prior research has proved that out of the above two approaches the Naïve Bayes classifier is the most suitable approach when comparing performance and accuracy[13].

2.2.3 Support Vector Machines Classifier

Several researchers indicated that this as the most popular classification algorithm. By constructing hyperplanes in a multidimensional space, it performs classification tasks. With multidimensional space, different class labels can be separated. Moreover, this classifier is considered as a non-probabilistic classifier.

2.3 Chapter Summary

This chapter discussed the work of other researcher's in previous studies. Most of the findings were a great contribution to the literature as well as to the new researchers. Also, in the related study, there are yet to find more, because of the ambiguity of various languages and that's why any research related to sentiments analysis is not outputted the 100% accurate results.

Chapter 3

Technology Adapted

3.1 Chapter Introduction

This chapter discussed the technologies adapted to implement the suggested model. Mainly pyCham software was used to implement the system. As the programming language python was used.

3.2 Python

I used the Python language as a programming language for the development of this project. Because python provides many standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces, and protocol and moreover most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python. Since this project is mainly based on natural language processing and those advantages over the Python language will make to do NLP tasks in a more efficient way. Python provides a powerful toolkit called NTLK which makes a platform for building Python programs to work with human language.

3.2 NLTK

NLTK is standing for natural language toolkit and it is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, semantic reasoning and stop words removing functionalities like so many various NLP tasks. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project [12].

3.3 TextBlob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, spelling correction and more [13].

3.4 PyCharm 2018 3.5

PyCharm used as the development environment for this project. Apart from IDLE, PyCharm is Python's most popular Integrated Development and Learning Environment.

3.5 Tkinter toolkit

Tkinter is a Python binding to the Tk GUI toolkit. The study used this toolkit along with python to build the model GUI [14].

3.6 WordNetLemmatizer

To remove "lemma" of a word I used WordNetLemmatizer which provided by nltk.

3.7 Chapter Summary

This chapter discussed the technologies used to implement the suggested model. Specially python language is used widely in this area and it includes a wide range of libraries that can be used to implement the suggested models.

Chapter 4

Methodology

4.1 Chapter Introduction

This chapter mainly focusses on the specific methods or procedures used to analyze the information regarding this study.

4.2 Statement of Research Problem

Rather than doing face to face communication now a day's people more likely to self-disclose on social media. Even though they put a social mask to hide their feelings in the real world unacquainted they reveal it in their post. This study explores to predict the mood/emotion of a person using those postings and so that people can know about themselves or others in a better way.

4.3 Process

In this study, two approaches are implemented in order to predict the mood. One method used a classifier model and it is shown in Figure 4.1. The other method used a lexicon and it is shown in Figure 4.2.

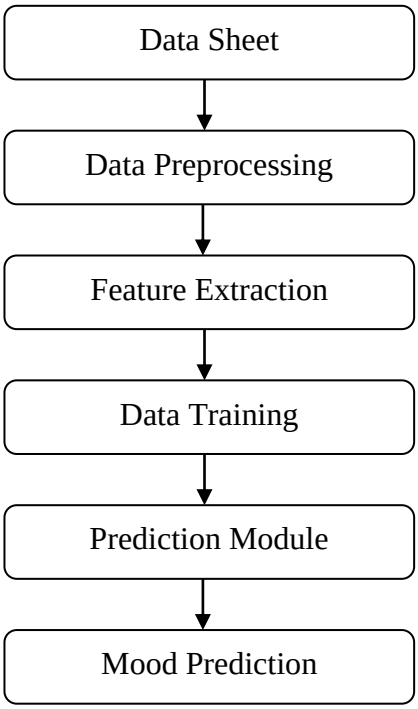


Figure 4.1: Classification Model

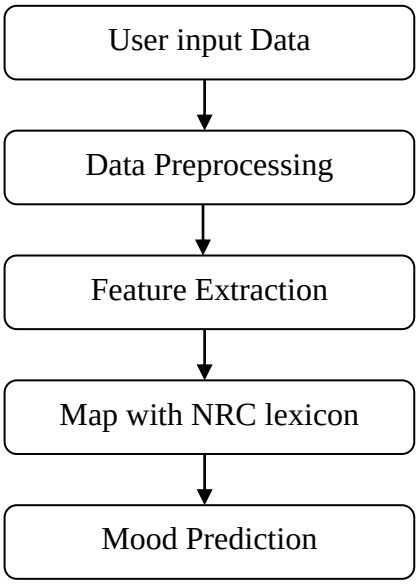


Figure 4.2: Lexicon Model

4.2 Expected outcome/alternative approaches

A software system was created to extract the sentiments by reading the text postings. And this system can predict the mood/emotion of a person. Emotions can be categorized as the following table (Table 4.3).

Emotions		
Positive	**Negative**	**Neutral**
Joy	**sadness**	**anticipation**
Trust	**anger**	
Surprise	**fear**	
	disgust	

Table 4.1: Types of Emotions

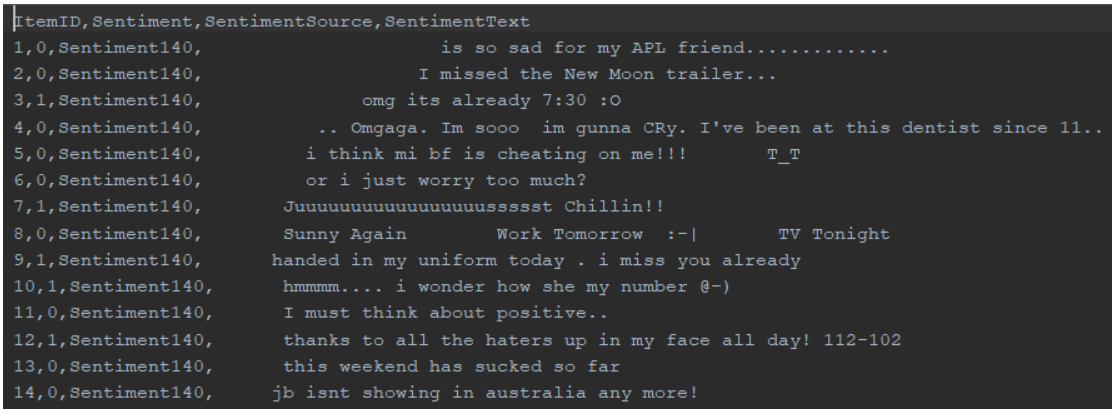
In this study, I used one posting from each user to predict the mood/emotion. Those emotions (Figure 4.1) are extracted using two approaches to compare the results. One approach is building a model using a supervised learning algorithm. In this approach, I used only 4 emotions among the eight. Those are joy, surprise, sadness, and anger. As the second approach, I used a lexicon to get the emotions. Here I used 4 emotions to get the output.

4.3 Data Set

I used an available online dataset from Kaggle for this study. Kaggle is a platform which provides public datasets for various research topics. This data set is provided by Sentiment140. Sentiment140 allows doing sentiment analysis on brand, product, or topic on Twitter. This dataset is already labeled to the polarity of each text.

4.4 Data Preprocessing

Most of the data that is collected from any source considered as unstructured data. Shown in figure 4.3. Using appropriate techniques data should convert them into structured data. The reason is when the unnecessary things of the text are removed, it is easier to train the data. And also, it will enhance the accuracy of the prediction module.



```
ItemID,Sentiment,SentimentSource,SentimentText
1,0,Sentiment140,                    is so sad for my APL friend.............
2,0,Sentiment140,                  I missed the New Moon trailer...
3,1,Sentiment140,              omg its already 7:30 :O
4,0,Sentiment140,          .. Omgaga. Im sooo  im gunna CRy. I've been at this dentist since 11..
5,0,Sentiment140,         i think mi bf is cheating on me!!!       T_T
6,0,Sentiment140,         or i just worry too much?
7,1,Sentiment140,       Juuuuuuuuuuuuuuuussssst Chillin!!
8,0,Sentiment140,       Sunny Again        Work Tomorrow  :-|       TV Tonight
9,1,Sentiment140,      handed in my uniform today . i miss you already
10,1,Sentiment140,      hmmmm.... i wonder how she my number @-)
11,0,Sentiment140,      I must think about positive..
12,1,Sentiment140,      thanks to all the haters up in my face all day! 112-102
13,0,Sentiment140,      this weekend has sucked so far
14,0,Sentiment140,     jb isnt showing in australia any more!
```

Figure 4.3: Example for unstructured data

There are various techniques to achieve this prepossessing process. Following mentioned are techniques that I have used in this study.

- Removing punctuations and other unnecessary characters
- Removing whitespaces
- Spelling correction
- Tokenization
- Part of Speech Tagging
- Stop word removal
- Lemmatization
- Negation Handling

4.4.1 Removing punctuations and other unnecessary characters

This system removes punctuations, other symbols such as ~#$@%^& and numbers form the given text. Figure 4.4 and Figure 4.5 are shown as examples.

I missed the New Moon trailer… #me @-) 123

Figure 4.4: Punctuations and other characters

i miss the new moon trailer

Figure 4.5: Punctuations and other characters are removed

4.4.2 Removing whitespaces

The system removes unnecessary white spaces that are placed in the text. Examples are shown in Figure 4.6 and Figure 4.7.

Sunny Again Work Tomorrow :-| TV Tonight

Figure 4.6: White spaces are included

sunny again work tomorrow tv tonight

Figure 4.7: White spaces are removed

4.4.3 Spelling correction

When the online text having a spelling mistake this system will correct the mistake and append the corrected word. Examples are shown in Figure 4.8 and Figure 4.9.

She is so pretty…

Figure 4.8: Spelling mistakes

she is so pretty

Figure 4.9: Spelling correction

4.4.4 Tokenization

Then after the cleaning process, this system tokenizes the text postings into a sentence and further it tokenized into words to identify the features correctly. After that sentences are stored to a python list.

Examples are shown in Figure 4.10, Figure 4.11 and Figure 4.12.

I just cut my beard off. It's only been growing for well over a year. I'm gonna start it over.

Figure 4.10: Facebook posting 1

```
[' I just cut my beard off.', "It's only been growing for well over a year.", "I'm gonna start it over."]
```

Figure 4.11: Sentence tokenization

```
['I', 'just', 'cut', 'my', 'beard', 'off', 'It', "'s", 'only', 'been',
'growing', 'for', 'well', 'over', 'a', 'year', 'I', "'m", 'gon', 'na',
'start', 'it', 'over']
```

Figure 4.12: Word tokenization

4.4.5 Part of Speech Tagging

POS tagging is done in order to achieve features of the text in the lexicon approach. The system needs to identify the adjectives of each sentence to map to the lexicon. Hence POS tagging is important to find these features. Examples are shown in Figure 4.13 and Figure 4.14.

I'm not happy and that’s makes him sad

Figure 4.13: Facebook posting 2

```
I PRP
'm VBP
not RB
happy JJ
and CC
thats NNS
makes VBZ
him PRP
sad JJ
```

Figure 4.14: POS tagging for each word

4.4.6 Stop word removal

Removal of stop words will enhance the accuracy of the system and it has no use in the process of sentiment analysis. Therefore, this system removes all the stop words from the text.

4.4.7 Lemmatization

Lemmatization is getting the base form of the word with the use of vocabulary and morphological analysis of words. Normally this removes only the inflectional endings of the word and returns the base or dictionary form of a word. This is which is known

as the lemma. When it comes to online text it can have various morphological forms. Therefore, to get the most accurate data it is important to extract the lemma of the word. And also, in the lexicon, the lexicon words are in their common base. Because of that getting lemma of the word is important in this study. Examples are shown in Figure 4.15 and Figure 4.16.

The **laughs** you two **heard were triggered** by **memories**

Figure 4.15: Facebook posting 3

```
laugh heard be trigger memory
```

Figure 4.16: Lemmatized words

4.4.8 Negation Handling

This is quite important preprocess because negation words like not, can't, isn't, never, nothing, etc. will change the sentiment of the text at once to the opposite sentiment. Considering this system summarizes all the negation words into the word "not". It will make the analysis more accurate and easier. Examples are shown in Figure 4.17 and 4.18.

```
['''I don't like him.we aren't close much''']
```

Figure 4.17: Facebook posting 4

```
['I do not like him.we are not close much']
```

Figure 4.18: convert into "not"

4.5 Supervised learning classification approach

After the data cleaning process, the study needs to identify the features of the text in order to perform the text classification. Since this study use supervised leaning there should be labeled data to train the classier model.

4.5.1 Feature Extraction

Furthermore, to perform sentiment analysis using text, it is important to identify the subjectivity/objectivity of the text. Any sentence is combined with subjectivity and

objectivity parts. But only subjective text holds the sentiments. The objective text contains only information[4].

Example-

1. Subjective: That was a superb movie.

This sentence has a sentiment(superb), thus it is subjective

2. Objective: James Cameron is the director of that movie.

This sentence has no sentiment, it is a fact, thus it is objective)

After identifying the subjectivity part of the sentence, then again polarity of that sentence also needs to be identified.

Moreover, subjective text can be classified into 3 categories as positive, negative and neutral based on the sentiments conveyed in the text. Known as the polarity of the text[14].

Example-

1. Positive – I love my new iPhone.
2. Negative – It was a horrible evening.
3. Neutral – I usually get up early in the morning.

When it comes to this study the dataset already labeled into whether the text is positive or negative but after examining it negation handling and other words handlings are not properly performed in the dataset. Therefore, some text postings give the wrong sentiment.

Therefore, the data set is again labeled into the polarity. Along with the polarity I manually labeled the text postings into selected moods/emotions. Selected emotions are joy, surprise, sadness, and anger. After labeling text is saved to a csv file. This data is getting to train and test the classifier.

4.5.2 Sentiment Classification

After setting the data, sentiment classification is done using an appropriate sentiment classifier. Purpose of this is when a text is given trained model should predict the polarity and the mood of the text. In this study, the Naive Bayes classifier is used as the model classifier. Because according to the literature the Naive Bayes classifier gives the highest accuracy results for text analyzing when compared to other supervised

classifiers[10]. After the creation of the model, the system will output the polarity or the selected mood.

4.6 Lexicon based approach

The lexicon used in this study is the NRC Word-Emotion Association Lexicon (NRC Emotion Lexicon) [15]. The lexicon has more than 3000 words and 8 emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) and polarity sentiment of each word is included.

As the first step in the lexicon-based approach, each sentence is tokenized into words in order to extract the adjective of the sentence. This is done using textblob pos tagging. Then, the extracted adjective is a map with the lexicon and get the count of each sentiment of each sentence. Highest counted sentiments will be output as the overall sentiment of the text posting.

4.7 Chapter Summary

This chapter discussed the research problem and the rationale for the application. Also discussed the specific procedures or techniques used to identify, select, process, and analyze information applied to understand the problem, thereby, this allows to critically evaluate a study's overall validity and reliability.

Chapter 05

Analysis and Design

5.1 Chapter Introduction

This chapter discussed the system by categorizing it into four main sections. They are Collecting Data, Data preprocessing, supervised learning approach and lexicon-based approach. Top Level Diagram of this study is shown in Figure 5.1.

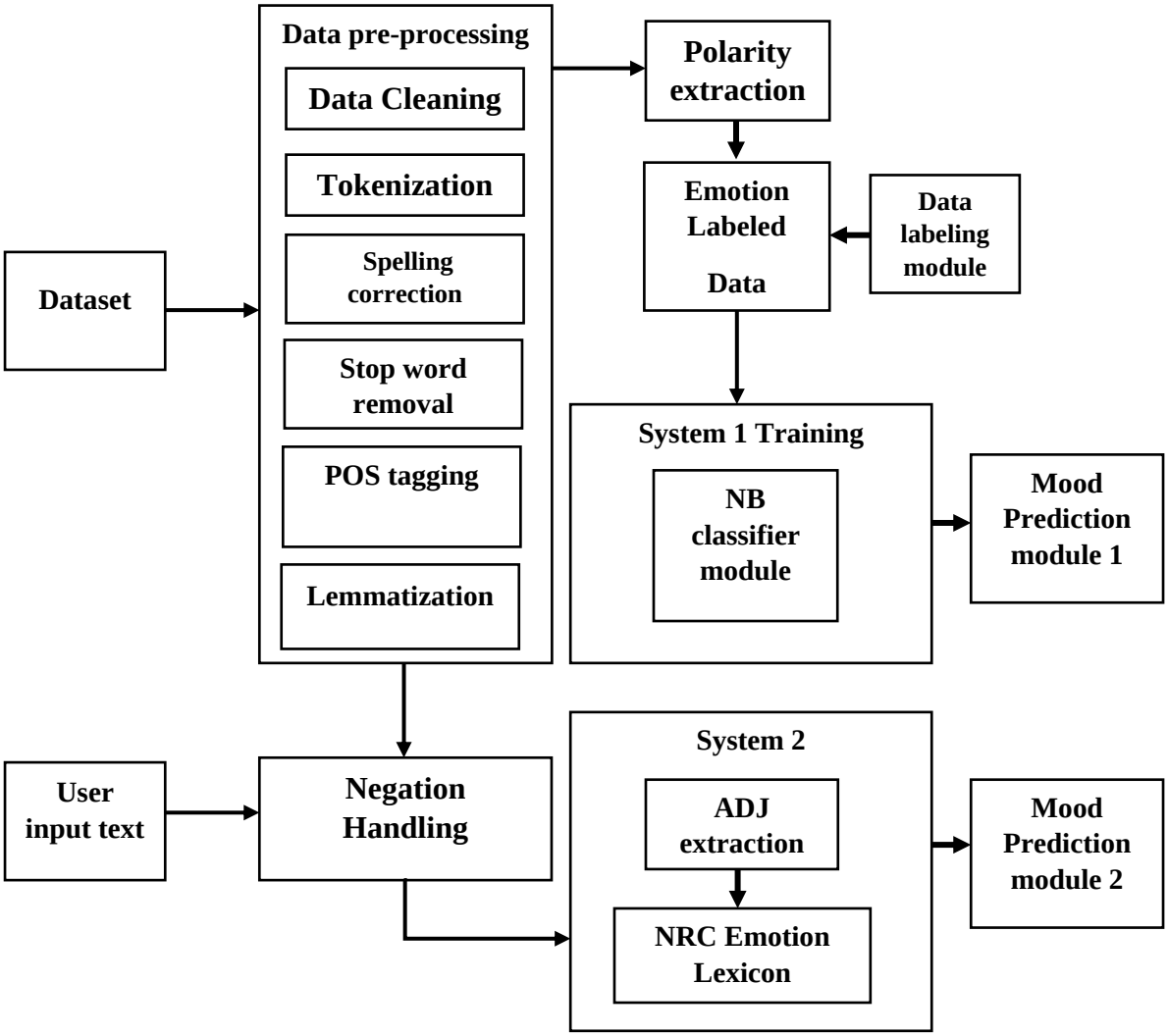


Figure 5.1: System Diagram

5.2 Collecting Data

This data set is provided by Sentiment140. Sentiment140 allows doing sentiment analysis on brand, product, or topic on Twitter. This dataset is already labeled to the polarity of each text.

5.3 Data Preprocessing

First posting of the user is read from the csv file. Using regular expressions punctuations, numbers and other unnecessary characters are removed. Then, whitespaces and unnecessary characters are removed. After that spelling is corrected using textblob library. Again, using NLTK stopwords corpus, stop words are removed. After this process, the text tokenized into sentences and further to words. And then POS tagging, Lemmatization and Negation handling is performed.

5.4 Supervised learning classification approach

In this approach polarity extraction of the text is done. And after the text is manually labeled to train and build the classifier model.

5.4.1 Feature Extraction

Textblob library is used to extract the polarity feature of the text. In textblob, the sentiment property returns a named tuple of the form Sentiment (polarity, subjectivity). The polarity score is a float within the range [-1.0, 1.0] where 1.0 will leads to the positive polarity and -1.0 will leads to negative polarity. The subjectivity is a float within the range [0.0, 1.0] where 0.0 is very objective and 1.0 is very subjective[16].

5.4.2 Sentiment Classification

Training data set is fed using the Naive Bayes classifier that provided by the textblob library. After feeding it model should be pickled otherwise the model will be trained again and again when the application is run. The pickle module is a powerful algorithm for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream [17]. This will reduce the processing time of the application.

5.5 Lexicon based approach

Instead of directly accessing the lexicon, BOW (bag of words) are created for each sentiment and saved as a separate list.

5.6 System training and evaluation

System training is performed by using previously labeled text in the text preprocessing stage. Once the system has produced outputs from the classifier model, the accuracy of outputs was determined by the previously labeled text.

5.7 Graphical User Interface (GUI) design

One Graphical user interface is designed in the user end. The user can input their thought and get the mood/emotion/polarity according to the entered text. This GUI is shown in Figure 5.2. Another GUI is created to data labeling process. It is shown in Figure 5.3.

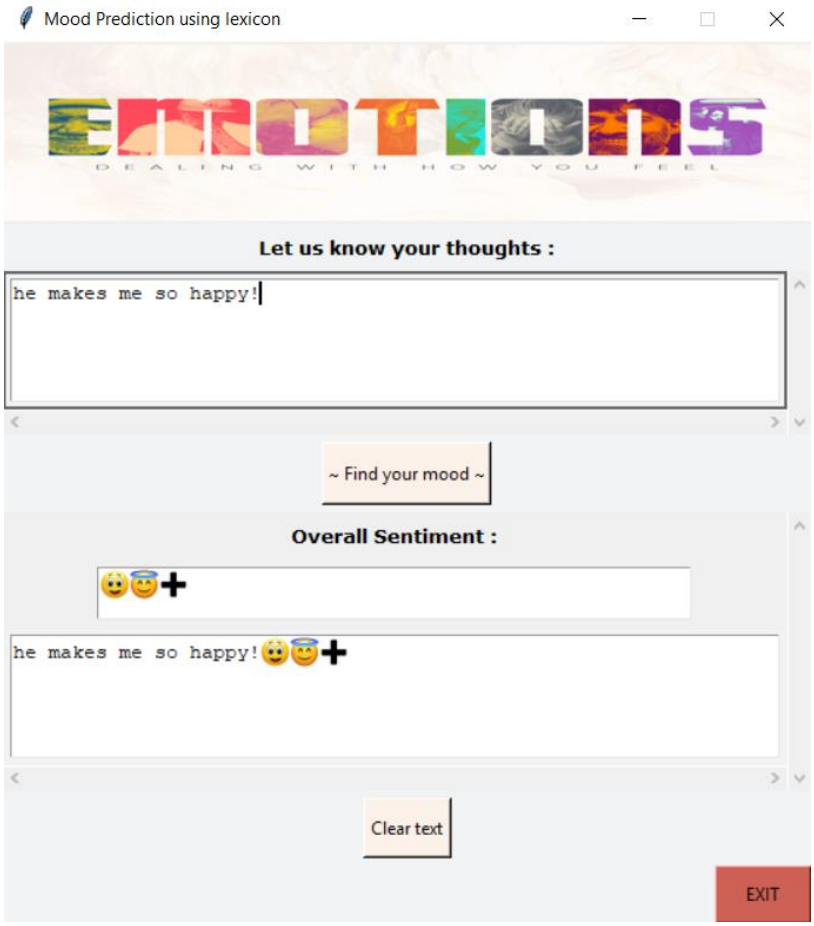


Figure: 5.2 User GUI

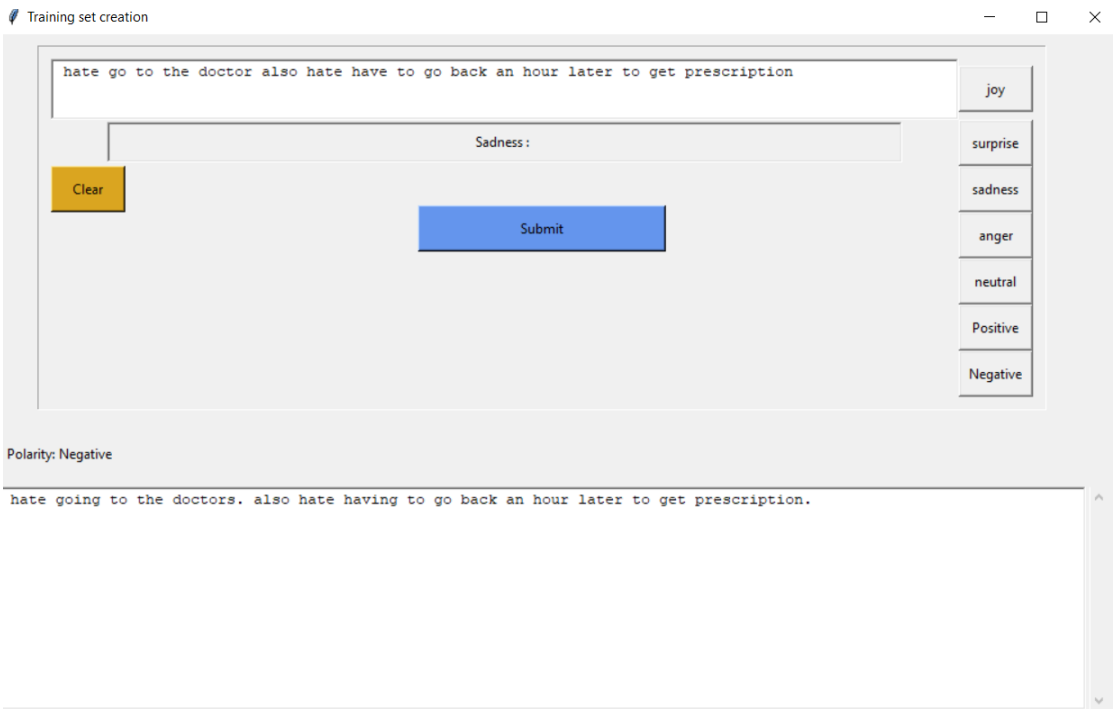


Figure: 5.3 Data labeling GUI

5.8 Chapter Summary

This chapter discussed the system by categorizing into four main sections. They were They are Collecting Data, Data preprocessing, supervised learning approach and lexicon-based approach.

Chapter 6

Implementation

6.1 Chapter Introduction

This chapter mainly focuses on the algorithms used to analyze and design the system. Since Python has a massive number of various libraries such as powerful NLTK library for natural language processing, Python programming language is used for the development of this project.

6.2 Data Preprocessing

First text preprocessing is performed using various text preprocessing techniques as mentioned in chapter 4.

6.2.1 Removing punctuations and other unnecessary characters

Punctuations and other unnecessary characters are removed using regular expressions. Removed punctuations and characters:

“!~#$@%^&*()_+=[]{}”’;:'"|<>,.?/1234567890\/”

6.2.2 Removing whitespaces

White spaces are removed using join and split functions in NLTK. The code segment for this task is shown in figure 6.1

```
def remove_white_spaces(self, text4):
    my_list = " ".join(text4.split())
    return my_list
```

Figure 6.1: Code segment for whitespace removing

6.2.3 Spelling correction

Spelling correction is performed using a method in the textblob library. The relevant code segment is shown in figure 6.2

```
def spelling_correction(self, record):
    self.sentence_record = record
    for line in self.sentence_record:
        b = TextBlob(line)
        self.spelling_corrected.append(str(b.correct()))
    return self.spelling_corrected
```

Figure 6.2: Code segment for spelling correction

6.2.4 Tokenization

Tokenization is performed in two stages. The first text is tokenized into sentences using the code as shown in Figure 6.3 and further sentences are tokenized into words as shown in Figure 6.4.

```
def tokenize_sentences(self, text):
    tokenized_sentences = sent_tokenize(text)
    return tokenized_sentences
```

Figure 6.3: sentence tokenization

```
def tokenize_words(self, text5):
    tokenized_words = word_tokenize(text5)
    return tokenized_words
```

Figure 6.4: word tokenization

6.2.5 Stop word removal

According to the study, there is no point of having stop words in the sentence in order to extract the feature set. Therefore, stop words are removed using the code given in Figure 6.5.

```
def stop_word_removal(self, text1):
    stop_words = set(stopwords.words('english'))
    stop_removed_sent = []
    for w in text1:
        if w.lower() not in stop_words:
            stop_removed_sent.append(w)
    return stop_removed_sent
```

Figure 6.5: stop word removal

6.2.6 Lemmatization

For the lemmatization process, this study used WordNetLemmatizer along with the wordnet pos tagging.

6.3 Supervised learning classification approach

To create a labeled data set separate module is created. In this module, the pre-pressed text is displayed, and emotions can be labeled, and it is saved to a separate file.

According to the textblob library if the value represented by the method polarity is a positive value that word is considered as positive text and if the value is negative it is considered as a negative text. In this study, I ignored the neutral polarity. Because of that, I did not consider the subjectivity of the text. Instead of that, I used the above code segment to get the polarity of the text.

6.4 Lexicon based approach

In this approach, the adjective word is extracted as the main feature of the text. Pos tagging is performed using textblob pos tag. Then the word “not” also extracted. It is needed to perform negation handling. The code segment for this task is shown in Figure 6.6

```
def textblob_adj(self, text):
    blobed = TextBlob(text)

    adj_list = []
    adv_list = []

    not_list = []
    adj_tag_list = ['JJ', 'JJR', 'JJS']
    adv_tag_list = ['RB', 'RBR', 'RBS']

    for (a, b) in blobed.tags:
        if b in adj_tag_list:
            adj_list.append(a)
        elif b in adv_tag_list:
            if a == 'not':
                not_list.append(a)
            else:
                adv_list.append(a)
        else:
            pass

    return adj_list, not_list
```

Figure 6.6: textblob pos tagging

6.5 Chapter Summary

This chapter mainly discussed the algorithms that are used to build the system. Futher more this chapter is given a clear picture of the research task and the process to reach the goal of the study.

Chapter 07

Evaluation and Results

7.1 Chapter Introduction

This chapter discussed the results of this study. Manly calculated the accuracy of both models using the confusion matrix.

7.2 Supervised learning classification approach

To train the classification model 6000 of data set is manually labeled and 2000 of data was taken to test the accuracy of the NB classifier. The system was indicated 50% accuracy for the classifier. Accuracy percentage and most informative features are shown in Figure 7.1

```
NaiveBayes Classifier accuracy percentage : 50.67621320604614
Most Informative Features
           contains(omg) = True           surpri : negati =    134.8 : 1.0
         contains(wonder) = True          surpri : joy    =    105.4 : 1.0
            contains(wow) = True          surpri : negati =     80.9 : 1.0
         contains(wouldn) = True          surpri : negati =     80.9 : 1.0
             contains(ya) = True          surpri : negati =     80.9 : 1.0
           contains(adam) = True          surpri : negati =     80.9 : 1.0
        contains(present) = True          surpri : negati =     80.9 : 1.0
          contains(local) = True          surpri : negati =     80.9 : 1.0
          contains(crown) = True          surpri : negati =     80.9 : 1.0
        contains(council) = True          surpri : negati =     80.9 : 1.0
```

Figure 7.1: NB model accuracy

7.2 Confusion Matrix

This is a terminology that is used to describe the performance of a classification model. In order to get the performance accuracy, Precision and recall functions are used. Equations are shown in Equation 01 as precision, Equation 02 as recall and Equation 03 as to accuracy. And Figure 7.4 shows the meanings of tp, fp, tn and fn.

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{(Equation 01)}$$

$$\text{Recall} = \frac{tp}{tp + fn} \qquad \text{(Equation 02)}$$

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn} \qquad \text{(Equation 03)}$$

tp: positive as positive	fp: positive as not positive
fn: not positive as positive	tn: not positive as not positive

Figure 7.4: Measures of Confusion Matrix

According to the confusion matrix, Data set and the accuracy for the classification model is shown in Figure 7.5 and Figure 7.6 respectively.

No of groups	**No of postings (for +ve)**	**No of postings (for -ve)**
tp	1143	1681
fp	905	426
fn	538	234
tn	414	659
Total Postings	3000	3000

Figure 7.5: confusion matrix for clasification model

For positive postings	**For negative postings**
Precision = (1143)/(1143+906) 0.558	Precision = (1681)/(1681+426) 0.797
Recall = (1143)/(1143+538) 0.678	Recall = (1681)/(1681+246) 0.872
Accuracy = (1143+414)/3000 51.9%	Accuracy = (1681+659)/3000 78%

Figure 7.6: Results of Classification model

7.3 Lexicon based approach

According to the confusion matrix, Data set and the accuracy for the lexicon model is shown in Figure 7.7 and Figure 7.8 respectively.

No of groups	No of postings (for +ve)	No of postings (for -ve)
tp	1124	1282
fp	963	827
fn	557	468
tn	356	423
Total Postings	3000	3000

Figure 7.7: confusion matrix for lexicon model

For positive postings	For negative postings
Precision = (1124)/(1124+963) 0.538	Precision = (1282)/(1282+827) 0.607
Recall = (1124)/(1124+557) 0.668	Recall = (1282)/(1282+468) 0.732
Accuracy = (1124+356)/3000 52.1%	Accuracy = (1282+423)/3000 56.3%

Figure 7.8: Results of lexicon model

7.4 Evaluation

According to the above results, it is clear, that the classification model gives high accuracy than lexicon model. For the classification model study used Document level sentiment analysis and for the lexicon model study used sentence-level sentiment analysis. Therefore, according to the results of this study, it can be concluded that the document level is given high accuracy than sentence level.

In the classification model, negative postings are given high accuracy than positive postings. That is because the dataset got for the study included more negative postings than positive postings. When the classifier model trained the dataset, it identified negative postings than positive postings.

Accuracies can be improved using various methods. In classification model accuracy can be improved by,

- Getting many datasets as training data

- Using more than one person for the manual labeling process and getting average labeled data
- Use of Random forest classification method
- Handling idioms

And in the lexicon model accuracy can be improved by,

- Using other features in the sentence such as bigram, n-gram
- Including more words for the lexicon
- Handling idioms

7.5 Chapter Summary

This chapter mainly focused on the results of the modules. A confusion matrix is used to calculate the accuracy. Imbalance of the data set caused the high accuracy in negative postings and low accuracy in positive postings. Also, the classifier model is far better than the lexicon model to get highly accurate results.

Chapter 08

Discussion

7.1 Chapter Introduction

This chapter briefly discussed the limitations on sentiment analysis and future trends on the regarding the area.

7.2 Future works and Limitations

Recently more attention has been paid to doing researches to assess the impact of social media sites. Especially discussions regarding opinion mining or the sentiment analysis have dominated research in recent years. For the development of this research area, some factors have been affected. As examples, in World Wide Web discussions on machine learning methods in natural language processing have increased to provide training data sets. One of the most recent applications of SA is Twitter1, Inc. It is an advanced incorporated tweet-searching function based on sentiment direction. Using this a user can search negative or positive tweets according to a specific topic. Another factor is all the web 2.0 applications such as social networks, review sites, wikis, and blogs are proving rich and diverse data. This will leads to getting different dimensions of data that can be used for SA[2].

Moreover, according to a recent study present SA can be grouped into four categories, namely; sentic computing, keyword spotting, lexical affinity, and statistical methods. Sentic computing is the most recent one and it makes use of ontologies and common-sense reasoning tools for a conceptual-level analysis of natural language text. In keyword spotting, get the most relevant unambiguous affect word and then text classification is done accordingly. When considering lexical affinity emotion or opinion polarity is calculated for random words. In statistical methods calculations are made according to the importance of keywords and co-occurrence of words[2].

In this context, it is worthwhile to consider the need for a more accurate text preprocessing technique. Still, there is no technique to handle all the ambiguities in various human languages. Consequently, anaphora resolution (the problem of resolving what a pronoun or noun phrase refers to) also presents a challenge when comes to text preprocessing.as an example, in this sentence “Alex helped Lily and he was kind”, what

“he” refers to. Another problem is abbreviations, which must be transformed into a standard form. Sometimes online text may exist in the form of a topographical structure. Handling those things also is a bit of a challenge. And some idiom phrases like “The ATM is not working. Brilliant!!” are still need to be resolved.
When it comes to classification algorithms above mentioned are considered as the most popular algorithms. Each has their own performances and accuracies. C. Jefferson, H. Liu, and M. Cocea have introduced a fuzzy based classification method for sentiment analysis. In their study, they compared the results with other algorithms such as Decision Trees and Naïve Bayes. They concluded their fuzzy-based approach performs better than above mention algorithms[17].

According to a recent website article, there can be remarkable milestones for sentiment analysis in the near future. SA will provide a service to every application and digital devices. There will be more complete profiles of individuals, under personal control like personal Health cloud. Also, the development of longitudinal analysis and root cause analysis. As example change of attitude of a person over a period and how daily activities of a person could affect to SA. Further, there will be scoring reflect new models of cognition such as how is the social gestures will intensify the patterns discovered through brain imaging and analysis of non-textual inputs like voice calls, typing speed/interval/error patterns in web browsers, facial expressions in online video chats and gestures in Google Glass. And, there can be aggregating SA for smaller, defined groups as micro public reporting. Moreover, there will be sentiment streaming. Which means sentiment as real-time presence, as mobile devices may emit a stream of moods according to the behavior of a person[18].

7.2 Chapter Summary

This chapter discussed about limitations and future works.

References

[1] M. A. Javed and C. Technology, “Numerical Optimisation of the Learning Process.”

[2] A. Kumar and T. M. Sebastian, “Sentiment Analysis: A Perspective on its Past, Present and Future,” *Int. J. Intell. Syst. Appl.*, vol. 4, no. 10, pp. 1–14, 2012.

[3] E. Haddi, X. Liu, and Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis,” *Procedia Comput. Sci.*, vol. 17, pp. 26–32, 2013.

[4] H. Kaur, V. Mangat, and Nidhi, “A Survey of Sentiment Analysis techniques,” *Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud) (I-SMAC 2017)*, pp. 921–925, 2017.

[5] M. M. Aldarwish and H. F. Ahmad, “Predicting Depression Levels Using Social Media Posts,” *Proc. - 2017 IEEE 13th Int. Symp. Auton. Decentralized Syst. ISADS 2017*, pp. 277–280, 2017.

[6] C. Paper, “Preprocessing Techniques for Text Mining Preprocessing Techniques for Text Mining,” no. October 2014, 2015.

[7] I. Technology, “Survey of Analysis of User Behavior in Online Social Network,” pp. 128–132, 2016.

[8] M. Munk, “Data Pre-Processing Evaluation for Text Mining : Transaction / Sequence Model,” vol. 18, pp. 1198–1207, 2013.

[9] M. Bouazizi and T. Ohtsuki, “A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter,” *IEEE Access*, vol. 3536, no. c, pp. 1–21, 2017.

[10] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[11] “Corpus linguistics.” [Online]. Available: https://en.wikipedia.org/wiki/Corpus_linguistics. [Accessed: 01-Jan-2017].

[12] H. A. Schwartz *et al.*, “Predicting individual well-being through the language of social media,” *Pac Symp Biocomput*, vol. 21, pp. 516–527, 2016.

[13] A. Kowcika, A. Gupta, K. Sondhi, N. Shivhre, and R. Kumar, “Sentiment analysis for social media,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, no. November 2013, 2013.

[14] K. Rafael -Michael student, “Sentiment Analysis For Social Media.”

[15] “NRC Word-Emotion Association Lexicon.” [Online]. Available: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm. [Accessed: 14-Dec-2018].

[16] S. Loria, “textblob Documentation,” 2014.

[17] C. Jefferson, H. Liu, and M. Cocea, “Fuzzy Approach for Sentiment Analysis.”

[18] P. Wolff, “What Sentiment Analysis milestones would you expect to see in ten years?” [Online]. Available: https://www.quora.com/What-Sentiment-Analysis-milestones-would-you-expect-to-see-in-ten-years. [Accessed: 09-Dec-2017].