

Sentiment Analysis on Twitter Data Using Machine Learning

by

Ravikumar Patel

A thesis submitted in partial fulfillment
of the requirements for the degree of
MSc Computational Sciences

The Faculty of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

© Ravikumar Patel, 2017

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Sentiment Analysis on Twitter Data Using Machine Learning	
Name of Candidate Nom du candidat	Patel, Ravikumar	
Degree Diplôme	Master of Science	
Department/Program Département/Programme	Computational Sciences	Date of Defence Date de la soutenance March 8, 2017

APPROVED/APPROUVÉ

Thesis Examiners/Examineurs de thèse:

Dr. Kalpdrum Passi
(Supervisor/Directeur(trice) de thèse)

Dr. Claude Vincent
(Committee member/Membre du comité)

Dr. Julia Johnson
(Committee member/Membre du comité)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Jinan Fiaidhi
(External Examiner/Examineur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Dr. David Lesbarrères
Monsieur David Lesbarrères
Dean, Faculty of Graduate Studies
Doyen, Faculté des études supérieures

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

I, **Ravikumar Patel**, hereby grant to Laurentian University and/or its agents the non-exclusive license to archive and make accessible my thesis, dissertation, or project report in whole or in part in all forms of media, now or for the duration of my copyright ownership. I retain all other ownership rights to the copyright of the thesis, dissertation or project report. I also reserve the right to use in future works (such as articles or books) all or part of this thesis, dissertation, or project report. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that this copy is being made available in this form by the authority of the copyright owner solely for the purpose of private study and research and may not be copied or reproduced except as permitted by the copyright laws without written authority from the copyright owner.

Abstract

In the world of social media people are more responsive towards product or certain events that are currently occurring. This response given by the user is in form of raw textual data (Semi Structured Data) in different languages and terms, which contains noise in data as well as critical information that encourage the analyst to discover knowledge and pattern from the dataset available. This is useful for decision making and taking strategic decision for the future market.

To discover this unknown information from the linguistic data Natural Language Processing (NLP) and Data Mining techniques are most focused research terms used for sentiment analysis. In the derived approach the analysis on Twitter data to detect sentiment of the people throughout the world using machine learning techniques. Here the data set available for research is from Twitter for world cup Soccer 2014, held in Brazil. During this period, many people had given their opinion, emotion and attitude about the game, promotion, players. By filtering and analyzing the data using natural language processing techniques, and sentiment polarity has been calculated based on the emotion word detected in the user tweets. The data set is normalized to be used by machine learning algorithm and prepared using natural language processing techniques like Word Tokenization, Stemming and lemmatization, POS (Part of speech) Tagger, NER (Name Entity recognition) and parser to extract emotions for the textual data from each tweet. This approach is

implemented using Python programming language and Natural Language Toolkit (NLTK), which is openly available for academic as well as for research purpose. Derived algorithm extracts emotional words using WordNet with its POS (Part-of-Speech) for the word in a sentence that has a meaning in current context, and is assigned sentiment polarity using ‘SentWordNet’ Dictionary or using lexicon based method. The resultant polarity assigned is further analyzed using Naïve Bayes and SVM (support vector Machine) machine learning algorithm and visualized data on WEKA platform. Finally, the goal is to compare both the results of implementation and prove the best approach for sentiment analysis on social media for semi structured data.

Keywords: Natural Language Processing (NLP), Data pre-processing, Word Tokenization, word stemming and lemmatizing, POS Tagging, NER, Machine learning, Naïve Bayes, SVM, Maximum Entropy, WEKA.

Acknowledgements

I shall begin by first acknowledging my thesis supervisor Dr. Kalpdrum Passi. He provided me with constant support and guidance whenever I ran into queries, and always welcomed any problems or questions that I had regarding the research and writing process of my thesis. Without his guided correct direction, my thesis would not have taken this shape. I would also like to thank the committee members that were involved in the validation process of my thesis: (Dr. Julia Johnson, Dr. Ratvinder Grewal). It is with their feedback, discussion and inputs that lead me to construct a well-developed thesis.

I would also like to thank Dr. Ann Pegoraro for providing me the access to World cup soccer data and Dr. Claude Vincent for his valuable suggestions.

I dedicate this thesis to my dad who was the greatest inspiration in my pursuit of higher education and I have been able to complete my thesis due to his blessings. I lost my father during my master's program, which was a personal loss and setback in my studies. However, I derived strength and inspiration from my father to continue with my studies. Finally, I would convey my heartfelt thankfulness for the constant support and encouragement that I received from my parents, brother and my best friend Kavina Shalin for her consistent support throughout the years of my study as well as when developing my thesis. The process of writing my thesis would not have been enjoyable without their unconditional support and love.

Table of Contents

Abstract	iii
Acknowledgements	v
List of Tables	viii
List of Figures	x
Abbreviations	xi
1 Introduction	1
1.1 Sentiment Analysis.....	1
1.2 Different approaches for Sentiment Analysis.....	3
1.2.1 Lexicon Based.....	3
1.2.2 Machine Learning	4
1.3 Methodology	5
1.4 Outline	6
2 Literature Review	7
2.1 Related Work.....	7
2.1.1 Sentiment Analysis on Twitter data	7
3 Data Set	11
3.1 About Twitter	11
3.2 Data format and characteristics	12
3.3 Data Set and variables	13
4 Data Preprocessing	16
4.1 Introduction	16
4.2 Python.....	17
4.3 Data Cleaning and Noise reduction	18
5. Linguistic Data Processing using Natural Language Processing (NLP)	29
5.1 Introduction	29
5.2 NLTK (Natural Language Toolkit)	32

5.3 Word Tokenization.....	33
5.4 Word Stemming	34
5.5 Word Lemmatization.....	37
5.6 Removing Stop Words	38
5.7 POS (Part-of-speech) Tagging	40
5.8 WordNet.....	45
5.9 SentiWordNet Dictionary.....	52
6. Machine Learning Techniques for Sentiment Analysis	59
6.1 Introduction	59
6.1.1 Implementation using WEKA.....	60
6.2 Analysis using Machine Learning.....	66
6.2.1 Naïve Bayes	67
6.2.2 Support Vector Machine (SVM).....	76
6.3 Results and comparison.....	80
7. Conclusions and Future Work	83
7.1 Conclusions	83
7.2 Future Work	85
References	86
Appendix A	91

List of Tables

Table 1: Statistics of Twitter platform	12
Table 2: Statistics of available data set.....	14
Table 3: Sample of Data Set.....	15
Table 4: Algorithm for pre-processing of Twitter linguistic data	21
Table 5: Example on removing URLs	22
Table 6: Example on replacing and filtering @username	23
Table 7: Example on removal of #Hashtags	24
Table 8: Example on removing repeated characters	26
Table 9: Analysis on File Size and Data processing time using derived algorithm	27
Table 10: Algorithm for Word Tokenization	34
Table 11: Example on Word Tokenization	34
Table 12: Algorithm for word Stemming and Lemmatizing	35
Table 13: Example on Word Stemming	36
Table 14: Example on Word Lemmatizing	38
Table 15: Example on Removing of Stop Word	39
Table 16: Example of Part-Of-Speech (POS) Tagging	41
Table 17: Algorithm for Marking Negation Words	42
Table 18: Example for Marking Negation word	44
Table 19: Algorithm extract emotional words from tweet	47

Table 20: Example of Word Sanitization	48
Table 21: Example on WordNet Synset	49
Table 22: Example on word lemmas	51
Table 23: Example of SentiWordNet Dictionary structure	54
Table 24: Assigning Sentiment Polarity to the Word	56
Table 25: Example of Output for Sentiment Analysis	58
Table 26: Confusion matrix for sentiment class (Naïve Bayes).....	70
Table 27: Accuracy of Sentiment Labeled dataset using Naïve Bayes.....	76
Table 28: Confusion matrix for sentiment class (SVM).....	79
Table 29: Accuracy of Sentiment Labeled dataset using SVM.....	80
Table 30: Comparison of Naïve Bayes and SVM.....	81

List of Figures

Figure 1: Overview on approach for Sentiment Analysis.....	5
Figure 2: Overview of Data pre-processing.....	18
Figure 3: Structure for pre-processing user tweets on Twitter	20
Figure 4: Reduction in file size after each pre-processing tasks.....	28
Figure 5: Derived architecture for Sentiment analysis using Natural Language Processing (NLP).....	31
Figure 6: Sentiment Analysis using machine learning.	60
Figure 7: Sentiment Classification of Tweets	62
Figure 8: Accuracy of overall positive sentiment tweets	63
Figure 9: Accuracy of overall negative sentiment tweets	65
Figure 10: Overview on Applying Machine Learning	67
Figure 11: Result of analysis using Naïve Bayes classifier.....	69
Figure 12: Result of analysis using SVM Algorithm.....	78
Figure 13: Comparison of Naïve Bayes and SVM Algorithm.....	82

Abbreviations

NLP	Natural Language Processing
NLTK	Natural Language Toolkit
POS	Part-Of Speech
NER	Name Entity Recognition
SVM	Support Vector Machine
IDE	Integrated Development Environment
CPU	Central Processing Units
BCPL	Basic Combined Programming Language
URL	Uniform Resource Locator
PosScore	Positive Score
NegScore	Negative Score
HDFS	Hadoop File System

Chapter 1

Introduction

1.1 Sentiment Analysis

In order to have a successful and a well-established business or an event, it is essential for the company or the event organizers to know the feedback and sentiments of the targeted customers or people that have reacted to it via social media. In this advancing world of technology, expressing emotions, feelings and views regarding any and every situation is much easier through social networking sites. The reaction of the customers and attendees on the social media is open ended and it may contain feedback from them in form of written text. Hence, what better way to monitor success of the products' promotion, famous personality, an event or an organization's achievement than through social media platform? Therefore, the public opinion regarding how popular the business is running, material is readily available in the form of social media blogs. These blogs contains valuable information that can allow analysts to extract decision making information through social media platforms.

Nevertheless, to assess this achievement, a standard process is required, and this is where Sentiment Analysis comes into play. Sentiment Analysis along with Opinion Mining are two processes that aid in classifying and investigating the behavior and approach of the customers in regards to the brand, product, events, company and their customer services (Neri *et al.* 2012). Also, the validation and evaluation done by sentiment analysis depends upon the syntactical tree that is formed during the analysis of the sentence and is not solely based upon the words or concepts that have a negative or positive meaning to it.

Sentiment analysis can be defined as the automatic process of extracting the emotions from the user's written text by processing unstructured information and preparing a model to extract the knowledge from it (Bird *et al.* 2009). Currently, many companies and organizations employ sentiment analysis to understand user's opinion for the product or the user's reaction to the event without being dependent on the surveys and other expensive and time consuming procedures. In this thesis, one such social networking site is taken into account, which is among the largest networking sites, Twitter. Looking at the statistics, users that are active monthly range at about 316 million, and on an average, about 500 million tweets are sent daily (Twitter, 2016). Due to the fact that these statistical values are extremely high, the content is restricted to a minimal level, and because the text has no uniform structures, social networking sites such as Twitter, and those similar to it put up challenges for the classifiers to analyze their data.

1.2 Different approaches for sentiment analysis

There are many approaches used for sentiment analysis on linguistic data, and which approach to be used depends on the nature of the data and the platform you are working on. Most research carried out in the field of sentiment analysis employs lexicon-based analysis or machine learning techniques. Machine learning techniques control the data processing by the use of machine learning algorithm and by classifying the linguistic data by representing them into vector form (Olsson *et al.* 2009). On the other side, Lexicon-based (also called Dictionary based) approach classifies the linguistic data using dictionary lookup database. During this classification, it computes sentence or document level sentiment polarity using lexicon databases for processing linguistic data like WordNet, SentiWordNet and treebanks. In this section, the brief discussion on lexicon-based and Machine Learning approaches has been outlined.

1.2.1 Lexicon-Based approach

The lexicon-based approach predicts the sentiments by using the lexical databases like SentiWordNet and WordNet. It obtains a score for each word in the sentence or document and annotates using the feature from the lexicon database that are present. It derives text polarity based on a set of words, each of which is annotated with the weight and extracts information that contributes to conclude overall sentiments to the text. Also, it is necessary

to pre-process data before assigning the weight to the words. The discussion on data preprocessing is explained in Chapter 3.

Moreover, Lexicon dictionary or database contains the opinionated words that are classified with positive and negative word type, and the description of the word that occurs in current context. For each word in the document, it is assigned with numeric score, and average score is computed by summing up all the numeric scores and sentiment polarity is assigned to the document. The detail discussion and implementation using lexicon-based approach is explained in Chapter 5.

1.2.2 Machine Learning approach

Machine Learning approach is widely seen in the literature on sentiment analysis. Using this approach the words in the sentence are considered in form of vectors, and analyzed using different machine learning algorithms like Naïve Bayes, SVM, and Maximum Entropy. The data is trained accordingly, which can be applied to machine learning algorithms. The detailed discussion on Machine learning approach is discussed in Chapter 6.

1.3 Methodology

In this thesis, both approaches have been combined, namely Lexicon-based and Machine learning for sentiment analysis on Twitter data. The algorithms were implemented for pre-processing of data set for filtering as well as reducing the noise from the data set. Therefore, the core linguistic data processing algorithm using Natural Language Processing (NLP) has been developed and implemented and discussed in Chapter 5, and assigned sentiment polarity to the tweets using lexicon-based approach. Finally, the data set is trained using machine learning algorithm: Naïve Bayes and SVM for measuring the accuracy of the training data set, and have compared results of both algorithms in Chapter 6. The most abstract view of derived approach that combines the lexicon-based and machine learning for sentiment analysis is shown in Figure 1.

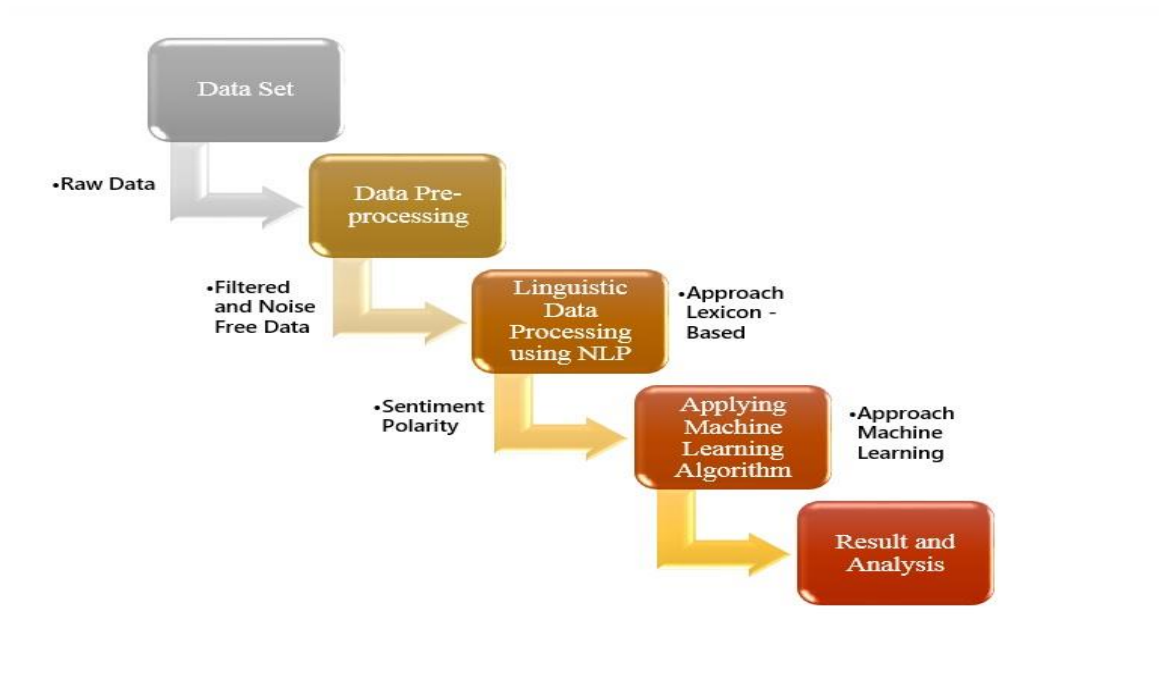


Figure 1: Overview on approach for Sentiment Analysis

1.4 Outline

I will discuss all the research steps performed while analyzing sentiments of the user tweets on world cup 2014 accordingly:

Chapter 2 discusses the approaches used by other researchers to perform sentiment analysis on linguistic data set.

Chapter 3 explains the data set available to us for performing sentiment analysis and how the data is structured for the Twitter platform.

Chapter 4 explains data pre-processing steps for filtering and reducing the noise from the data set.

Chapter 5 introduces with core functioning algorithm for processing linguistic data using Natural Language Processing (NLP) concept and data preparation for machine learning algorithm.

Chapter 6 Analysis and comparison of Result using Machine Learning and Data visualization using WEKA platform.

Chapter 7 is on conclusion and future work for performing sentiment analysis.

Chapter 2

Literature Review

2.1 Related Work

2.1.1 Sentiment Analysis on Twitter data

In this era, information sharing through social media has increased and most users actively share their personal ideas and information publically. This information for an analyst or researcher is a gold mine to dig out the valuable information for strategic decision-making (Younis *et al.* 2015). Now-a-days, most people review others' opinion, and openly convey their agreement or disagreement with the argument. For example, asking friends for their reviews about the new movie in theater, looking over public reviews of a product before buying it, voting in an election and taking into consideration the political party or the candidate who promises the best for the society based on public pole.

Twitter is an online social networking site and contains huge number of active users who enthusiastically share their thoughts and reviews on events, news, products, sports, elections. These reviews, written by the users, express their sentiments towards the topics

they tweeted. Fishing out sentiments embodied in the user's written text, in the world of social media is known as Sentiment analysis or opinion mining. Firmino Alves *et al.* (2013) states that from the beginning of the 21st century, sentiment analysis is one of the most interesting as well as active research topic in the domain of Natural Language Processing. It helps the decision maker to understand the responses of people towards a particular topic and aids in determining whether the event is positive, negative or neutral. Twitter has been considered a very important platform for data mining by many researchers. Hemalatha *et al.* (2014) discusses that the Twitter platform contains more relevant information on particular events with hashtags that has been followed and accepted by many popular personalities.

Neri *et al.* (2012) in their experiment, classified that negative or positive polarity is not the only concept of sentiment analysis. It is a data structure that analyzes the words from root node to parent node of the sentence structure. Further, it is a system for sentence structure that analyze the word meaning, its synonyms, expression, and changes the polarity in case of negation word. It also, changes and modifies the polarity of word based on adverbs, noun, and adjective. In their research, Isha *et al.* (2014) suggest that the aim of sentiment analysis is to detect and mine the sentiments, moods and attitudes of individuals and groups. Also, Sentiment detection from the natural language written by the user in social media environment is a challenging task. Moreover, emotions contained in the sentence possess the ability to distinguish the nature and feelings of humans with regards to the events they are watching. The **application** of sentiment analysis can be the review of customer towards the products, opinion of voters during election, individuals feelings after winning or losing

sports game, stock market opinion, as well as in many other business domain that rely on customer feedback and services.

The main **fundamental objective** of the sentiment analysis is to classify sentiment polarity from the text whether it is positive, negative or neutral. This classification can be done at the sentence level, document level or with the entity and aspect level. There are many approaches to classify the sentiment polarity from the user generated text. Firmino Alves *et al.* (2013) give an insight of the main approaches for classifying sentiment polarity which are: machine learning, statistical approach, semantic approach and approach based on lexical analysis or thesaurus. Augustyniak, Łukasz *et al.* (2015) describe that in the world of opinion mining predicting sentiment polarity from the text can be done by employing the specialists to manually classify the polarity, and can be done automatically or using both techniques.

Hemalatha *et al.* (2012) shows a very nice approach for pre-processing Twitter data following simple steps, and demonstrates how to prepare the data for training in machine learning technique. This approach eliminates unnecessary noise such as slang, abbreviation, URL, special characters from the linguistic data and also reduces the size of data set by eliminating noise. Extending the work in other literature; Hemalatha and her colleagues derived a combined pre-processing and classification approach executed parallel to achieve high performance, reduced data size and produced more accurate results by classifying the features from the sentiment words by adding polarity of it, and applied machine learning techniques to the derived data set. Bandgar and Kumar (2015) using their research methodology illustrated how to create a windows application for real-time Twitter data for

pre-processing of text data using available natural language processing resources like WordNet, SMS dictionary, Stanford dictionary.

Augustyniak, Łukasz, *et al.* (2015) proposed a new method called “**frequentiment**” that robotically evaluates sentiments (opinions) of the user from amazon reviews data set. Extending the work in this method, they developed dictionary of words by calculating probabilistic frequency of words present in the text and evaluated the influence of polarity scored by separating the features present in the text. They analyzed the outcome that was produced by unigram, bigram and trigram lexicon using lexicon based, supervised and unsupervised machine learning approaches, and compared 37 machine learning methods to evaluate results in analyzing the amazon dataset. Here, the authors claim that it is one of the most comprehensive domain of sentiment analysis in the literature. Isha *et al.* (2014) reported in their research paper, illustrate how they developed a reliable framework for sentiment analysis using machine learning and lexicon based approach. The case study compared sentiment analysis for three products and brands using Naïve Bayes algorithm as a baseline classifier, which shows significant results with accuracy for the product safety. The literature of Neri *et al.* (2012) shows comparison of sentiment analysis of 1000 Facebook post from newscasts by using knowledge based system. Neri *et al.* (2012) proposed semantic and linguistic approaches for classifying and analyzing the huge amount of distributed data, and assigned automatic polarity classification for sentiment analysis to use in the knowledge based system.

Chapter 3

Data Set

3.1 About Twitter

Twitter is a social networking or a blogging platform that was founded in 2006 by Jack Dorsey, Biz Stone, Noah Glass and Evan Williams (*Twitter*, 2016). The idea was to develop an SMS-based communication platform, where a group of people create their account, update the status and can text using the platform. This idea was initially proposed by Jack to his partners Biz and Evan during the brainstorming session at the podcasting company Odeo. Later, after going through more research the platform Twitter, referred as ‘twtr’, was founded, and Jack sent the first message on Twitter on March 21, 2006, 9:50pm by setting up the account on Twitter platform (MacArthur 2016). Twitter today, has become the most popular and successful social networking site. Twitter serves as a platform where people can freely express their thoughts, feelings, discuss issues, and also state beliefs and opinions (Hemalatha *et al.* 2012). Not only one’s ideas and beliefs, but also others’ philosophies and principles, and in order to do so, one has to just follow the other person on Twitter. Table 1 shows statistics about the Twitter as of June 30, 2016 (MacArthur, 2016).

Table 1: Statistics of Twitter platform

Monthly Active users	313 M
Unique visits monthly to the sites with embedded Tweets	1 Billion
Active users on Mobile	82 %
Employee around the world	3860
Offices around the world	35 +
Accounts outside U.S.	79 %
Languages supported	40 +
Employees in technical roles	40%

Source: "Company | About." *Twitter*. Twitter, 30 June 2016. Web. 04 Mar. 2017.

3.2 Characteristics of Twitter Data

Furthermore, the SMS-based platform for Twitter is developed to present one's idea in a concise and in effective manner. Therefore, tweets are formulated to be a maximum of 140 characters long in size, however; within the tweet, sharing videos, pictures and other tweets. always serves as an option (MacArthur, 2016). This short and precise description of one's thoughts and sentiment can be conveyed (Hemalatha *et al.* 2012). Also, Twitter data consist of '#Hashtags', which is the most important and meaningful symbol in the Twitter platform. This number sign, or pound sign, or hashtag is used to identify the topics, events, company or a keywords in every tweets on Twitter. For example, '#DonaldTrump' on Twitter showcases all the current or live information like news, photos and videos. about Donald Trump, the newly elected President of United States. This means # is a primary symbol to

identify the person, company, sports, or any public event occurs around the world and people react to it on Twitter platform. Another important attribute or symbol on Twitter is '@' followed by a word or name, represents the user id for the account on Twitter. For example, '@narendramodi' in Twitter comment is the username (narendramodi) the, Prime Minister of India. Moreover, one can see his/her followers, tweets, retweets, likes, and can also reply on one's account with the username i.e. '@username'. For Example, in the data set available '@username' represents the name of the user, which can be seen in the 'text' attribute of the data set, and in 'screen_name' attribute (as shown in Table 3). Furthermore, the single user tweet contains number of people following it, date and time when the user tweeted, retweet status, and the text blog where the user wrote the comment. Here, the text attribute of Twitter data set that contains user's opinion is taken into account for sentiment analysis using lexicon based and machine learning algorithm.

3.3 Data Set and Variables

The Twitter Data available is of World Cup Brazil 2014 with the hashtags '#brazil2014', '#worldcup2014', and games hashtags, as shown in Table 2. This data set distinguish the tweets based on the hashtags namely #brazil2014, #worldcup2014, #ALGRUS (Algeria vs Russia) as well as, other games and event. The hashtag #worldcup2014 contains all the tweets from the date 06-June to 14-July, 2014 (40 Days), which consist of 44,040,192 user tweets globally during the world cup. Similarly, the hashtag #brazil2014 comprises of all the user tweets on the promotion of the world cup, which started on 08-June to 15-June, 2014 (8 days). Moreover, it classifies the tweets by the game played between any two countries; for example, #ALGRUS (Algeria vs Russia)

only contains tweets representing this particular game or match. There was a very good response from people all around the world giving their views on World Cup events, promotion and the players. The data set available for the analysis contains a huge number of tweets for the game hashtags which has approximately 2 million tweets. The statistics on overall tweets in the data set can be shown in Table 2.

Table 2: Statistics of available data set

Hashtags #	Date	Number of Tweets	File Size (approximately)
#brazil2014	08-June to 15-June, 2014 (8 Days)	1,415,958	268 MB
#worldcup	06- June to 14-July,2014 (40 Days)	44,040,192	4 GB
Game Hashtags (e.g. #ALGRUS Algeria vs Russia)	June - July,2014	Approx. 2 Million Tweets	More than 2 GB

Table 3 shows a sample of the data set with its attributes and tweeted data by user. The data set contains six attributes namely id (user id), created_at (date and time), screen_name (@username), followers_cnt (Number of followers), retweet, Text (or the blog posted by user). A single tweet by the user contains all this information compact in data set.

Table 3: Sample of Data Set

id	created_at	screen_name	followers_cnt	retweet	Text
4760000000	Sun Jun 8 19:49:54 2014 CDT	ravi2talk28	4	TRUE	RT @MeetTheMazda: birthday From Waka Waka for South Africa to this for Brazil. LOVE Shakira _ÜÖÄ #Brazil2014 http://t.co/TJc2QL6K7b
4760000000	Mon Jun 9 23:59:58 2014 CDT	Franc*****	185	FALSE	Feel it, it's here I know how Brazilians r feeling, that feeling is special @robertmarawa @YesWeCrann @Soccer_Laduma @GoalcomSA
4760000002	Mon Jun 9 23:59:16 2014 CDT	B**Farlz	27	TRUE	RT @Socceros: NEWS Chile are likely to be without Arturo Vidal for our #Brazil2014 opener - http://t.co/yJ4ej6M6lS #GoSocceros #CHIAUS

In this particular Chapter, the data set and attributes are explained in detail. As per Younis *et al.* 2015, the text attribute is a gold mine to dig out valuable information for strategic decision making. In order to perform sentiment analysis from this data set, the data is to be pre-processed (Chapter 4), which cleans and remove unnecessary noise from the data set. In Chapter 5, the data set undergoes Linguistic Data Processing using Natural Language Processing (NLP) and POSITIVE, NEGATIVE or NEUTRAL polarity is assigned respectively. Furthermore, in Chapter 6 Machine Learning Techniques namely Naïve Bayes and SVM (Support Vector Machine) are applied to analyze the sentiment labeled data using WEKA platform for data mining.

Chapter 4

Data Preprocessing

4.1 Introduction

Twitter data made available to conduct this research is in semi-structured data set. The data set contains ‘text’ field where the user generated tweets are used for research, which may consist of noise as well as partial and unreliable linguistic data. Hence, in order to analyze linguistic data from Twitter, it is necessary that this irregular data be cleared and removed, so the true meaning and sentiments can be accounted for from the data (Hemalatha *et al.* 2012). This is where data preprocessing comes into play. To filter and remove the noise from the data, the algorithm implemented using Python Programming language and all the preprocessing tasks for filtering the noise from the data are discussed in the following document.

4.2 Python

Python is a powerful programming language. The data structure and object-oriented programming concepts helps a programmer with the effective and efficient way of programming with minimum lines of code (Van Rossum *et al.* 2007). As it is open source, its interactive interpreter permits one to directly code ones program as well as lets access to many standard libraries and resources that are freely available on the web to fulfill your requirement for application development. It is also most suitable language for scripting and application development with regards to its sophisticated syntax and dynamic tying, which is more interesting for processing linguistic data (Bird *et al.* 2009). Its important feature is dynamic name resolution (late binding), which allows methods and variable names binding during execution of program (Van Rossum *et al.* 2007).

During the course of this research, Python 3.4 version was used for processing linguistic data using nltk (Natural Language Toolkit), which is most compatible with this version of Python. For programming interface ‘JetBrains PyCharm Community Edition 2016.1.4’ has been used that facilities Integrated Development Environment (IDE) for implementation, code compilation, error checking and editing, as well as, navigating and refactoring of the code.

4.3 Data Cleaning and Noise Reduction

Data set available on world cup 2014 contains text field, in which user's comments or tweets information on particular event or game is available. These tweets are in unstructured form of data and are full of noise and unwanted information. This textual data is full of unwanted text, special and repeated characters, and may contain unwanted space in it.

Therefore, in order to perform sentiment analysis on this data set, the preliminary step is to pre-process this data and transform it so that the machine learning algorithm analysis can be performed to it. Hence, in order to properly analyze this data from tweets, it is necessary that this irregular data is cleared and removed, so the true meaning and sentiments can be accounted from the sentence (Hemalatha *et al.* 2012). Preprocessing of data normalizes the linguistic data, eliminates noise and simplifies the vocabulary used for analyzing sentiments from the tweets (Fernández-Gavilanes *et al.* 2016). The most generic view for preprocessing can be shown by following Figure 2.



Figure 2: Overview of Data pre-processing

There has been a lot of research accounting for pre-processing text or linguistic data by the same authors. In the article "Preprocessing the informal text for efficient sentiment analysis" by Hemalatha *et al.*(2012), they demonstrated a proper order for pre-processing informal text and showed how it can be better for performing data mining tasks. In the following year Hemalatha *et al.* (2012), have published their work by developing a tool for sentiment analysis using machine learning algorithm. Here, they illustrated within the framework for natural language processing to extract the qualified content from the text data that can result in better sentiment analysis using machine learning algorithm. Later, in the case study published by Hemalatha *et al.* (2014), they suggested the removal of words and expressions that have no meaning to it in order to achieve better performance and results. In this thesis, the development of an algorithm for pre-processing of Twitter text data is been discussed based on the idea of Hemalatha *et al.* (2012) (2014).

There are various steps to be performed in order to reassess the data correctly and determine the true meaning behind it, through data processing. It is also necessary to follow proper sequence to pre-process data to achieve accuracy as well as consistency in data set. Taking the reference into account of Hemalatha and her colleagues' work (2012), the algorithm has been implemented for pre-processing tweets by modifying some of the functions and steps they suggested. The abstract idea for data pre-processing is show in Figure 3 below.

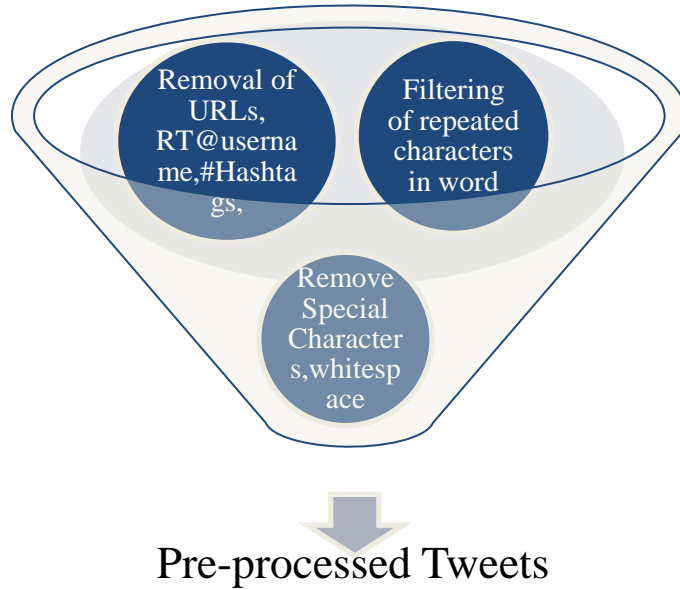


Figure 3: Structure for pre-processing user tweets on Twitter

From the above discussion, the basic idea about how to pre-process Twitter text data and steps need to perform to preprocess tweets is shown. Since, if the proper steps sequence is followed for eliminating noise from the data, to obtain more accuracy and consistency in the output from the pre-processing step. So, the developed algorithm that perform Natural Language Processing are shown in Table 4.

Table 4: Algorithm for pre-processing of Twitter linguistic data

<p>Input: Twitter comments or Text data</p> <p>Output: Pre-processed Text data for next step of Natural Language Pre-processing Task.</p>
<p>For each comment in Twitter Data File</p> <p>Initialize temporary empty string processedTweet to store result of output.</p> <ol style="list-style-type: none">1. Replace all URLs or https:// links with the word 'URL' using regular expression methods and store the result in processedTweet.2. Replace all '@username' with the word 'AT_USER' and store the result in processedTweet.3. Filter All #Hashtags and RT from the comment and store the result in processedTweet.4. Look for repetitions of two or more characters and replace with the character itself. Store result in processedTweet.5. Filter all additional special characters (: \ [] ; : { } - + () < > ? ! @ # % *,) from the comment. Store result in processedTweet.6. Remove the word 'URL' which was replaced in step 1 and store the result in processedTweet.7. Remove the word 'AT_USER' which was replaced in step 1 and store the result in processedTweet. <p>Return processedTweet.</p>

In the first step the algorithm clear out all the URLs present in the tweets. This step of pre-processing will eliminate all the 'URLs' from the Dataset and will result in reducing the noise as well as decreasing the size of dataset. However, the output generated will remain with the meaningful information in the tweet. Also, in the developed algorithm 'www. &' and 'https: //' is converted to the word 'URL' using regular expression function available in Python. This can be imported using regular expression (.re) module in Python, which gives programmer an embedded functionality inside Python language to operate textual or

string data set (Kuchling, A. M, 2014). This will eliminate all the URLs via matching regular expression and replacing it with generic word ‘URL’. For Example, as shown in figure below,

Table 5: Example on removing URLs

	Text Data
Input Tweet	The Best World Cup Song So Far READY FOR BRAZIL # RT @username World Cup Song http://t.co/O3wZGPsAxx #Worldcup2014 #Brazil2014
URL Processed Tweet	The Best World Cup Song So Far READY FOR BRAZIL # RT @username World Cup Song URL #Worldcup2014 #Brazil2014

The second step to perform preprocessing is to remove ‘@username’ from the tweet. ‘@username’ is the tag with ‘@’ followed by the user id or the user name on Twitter platform. The information can be found with ‘@username’ tag, and retweets have been abbreviated as ‘\RT’. Retweet is the process when any user re-posts the comment on others account, which describes the reaction of the user behavior to that particular post (Hemalatha *et al.* 2012).

In this process, two steps approach was used to eliminate RT ‘@username’ from the tweet. If a person likes the thoughts or opinions expressed in another tweet, he/she could retweet it (Hemalatha *et al.* 2012). This symbolizes “RT” in the tweet, which by itself does not stand for any meaning (Hemalatha *et al.* 2012). Hence, eliminating them would make the data free of complexity and useless characters. It can be done using regular expression function in Python, a pattern for ‘@’ followed by the ‘username’ and replace the whole word with ‘AT_USER’ can be discovered. This will replace all the ‘@username’ with static

word ‘AT_USER’. Secondly, to find a word ‘RT’ followed by the ‘AT_USER’ and replace ‘RT’ and ‘AT_USER’ by a blank character to remove from the tweet. The reason for this is, find ‘RT’ using regular expression without ‘AT_USER’ in the tweet, it can replace all the word that contains ‘RT’ or ‘rt’ in it. For example, let say a word ‘**Happy Birthday**’ in a tweet and it needs to eliminate letters ‘RT’ from the tweet. It will give a result like ‘**Happy Bihday**’, which gives incorrect meaning to the word when applied to the lexical resources. Therefore, to implement the idea for eliminating ‘RT’ just followed by ‘@username’ and if no such pattern found it will just replace ‘AT_USER’ to the blank space. Removing the retweets from the tweets would eliminate the username which has no meaning to it and would give the actually message needed (Hemalatha *et al.* 2012).

This preprocessing step reduces the size of data set as well as eliminates the information that doesn’t contain sentiment or emotional meaning to it. The example of a processed tweet is shown below in Table 6:

Table 6: Example on replacing and filtering @username

	Text Data
Input Tweet	The Best World Cup Song So Far READY FOR BRAZIL # RT @username World Cup Song URL #Worldcup2014 #Brazil2014
Output Tweet Step-1	The Best World Cup Song So Far READY FOR BRAZIL # RT AT_USER World Cup Song URL #Worldcup2014 #Brazil2014
Processed Output Tweet Step-2	The Best World Cup Song So Far READY FOR BRAZIL #World Cup Song #Worldcup2014 #Brazil2014

Here, one can observe that information followed by the ‘#Hashtags’ are the event tags. These may sometimes give emotional or sentimental meaning to the word. Therefore, in order to preserve the word followed by the hashtag, the third step is to remove only the ‘#’ or ‘hashtag’ symbol from the tweet. This is evident in table below.

Table 7: Example on removal of #Hashtags

	Text Data
Input Tweet	The Best World Cup Song So Far READY FOR BRAZIL #WorldCup Song #Worldcup2014 #Brazil2014
#Hashtag Processed Tweet (Output)	The Best World Cup Song So Far READY FOR BRAZIL World Cup Song Worldcup2014 Brazil2014

After eliminating URLs, retweet and username, and #Hashtag from the text data it becomes more meaningful and each word gives us some meaning. Again, this was only some basic steps to be performed to pre-process Twitter text data for analysis which not only removes noise from the data, but also, reduces the size of the dataset as well as increases performance for further data processing task (as shown in Figure 4).

Furthermore, the user generated information may also contain unnecessary whitespaces at the beginning, in between or at the end of the tweets, special characters like punctuation and repetition of characters. First, all extra white space was removed using the `strip` function available in Python. Secondly, all the meaningless and unnecessary special characters from the tweets were eliminated (Hemalatha *et al.* 2012). These characters include: `\ | [] ; : { } - + () < > ? ! @ # % *`, and a few more. Neither do these characters have specific and special meaning, nor do they explain if these characters are used for positivity or negativity, hence; removing them is the best option. Also, sometimes these

special characters are attached to the word like “sweetheart!” If you compare these words using the dictionary, it would not contain words with special characters (in this case, an exclamation mark (!)), and so, the dictionary would be unable to find the meaning associated with it (Hemalatha *et al.* 2012). So, if the comment was positive, and the dictionary does not recognize the word, it would decrease the polarity of the positive comment, making it a neutral comment, and giving inaccurate results.

In order to express their strong feelings, people often use word with multiple characters (Hemalatha *et al.* 2012). For example, “LOVEEEEE”. The number of ‘Es’ in this word are unnecessary and do not belong to lexical resources (e.g. SentiWordNet Dictionary), and are therefore, required to be eliminated (Hemalatha *et al.* 2012). However, there can also be words that might have a character repeating twice in them such as “Egg”, where it is necessary to have an extra ‘g’ in order to understand the true meaning of the word. Moreover, there are no words that have characters repeating more than twice. So, when programming, it is essential to state a rule that accounts for characters repeating twice, but not for those that repeat more than twice (Hemalatha *et al.* 2012). This would help eliminating extra and meaningless characters from tweets (stated earlier “LOVEEEEE” would become “LOVEE”), making the information more relevant. For example, the word “GOOOOOD” in input text has multiple sets of ‘O’, which will not give us polarity score when lexical resources is used (Table 8). Therefore, using developed algorithm in Table 4 it primarily removes special characters from the sentences, and have also taken out repeated characters in words to achieve sentimental meaning from words in the sentence.

Table 8: Example on removing repeated characters

	Text Data
Input Tweet	The Best World Cup Song So Far READY FOR BRAZIL Very GOOOOOOD!! username World Cup Song Worldcup2014 Brazil2014
Remove repeated and special characters from tweet (Output)	The Best World Cup Song So Far READY FOR BRAZIL Very GOOD World Cup Song Worldcup2014 Brazil2014

Preprocessing the tweets concludes with decrease in noise and reduction in size of dataset (as shown in Figure 4). This further achieves high performance analyzing data when applied to machine learning algorithm. Here, in this research the size of dataset has been matched after each data processing steps. The result obtained after preprocessing, reduced the size of dataset; hence, other Natural Language Processing tasks can be performed on the dataset. In the example of analysis, the raw tweet data from period 2014-06-08 to 2014-06-09 is taken into account, which has 24,336 raw tweets and the size of ‘.csv’ file is 4,308 KB. The raw tweets contains 100% noise in data and it is still not processed. After removing URLs from the data, there was a significant drop in the size of dataset to 3,695 (85.77%). It shows us that how much Twitter information consist of raw junk of URLs in tweets which may result in collecting inaccurate meaning from data. The processing time for filtering URLs from the dataset took 1.15 seconds. Further, to process sentence analysis and eliminated all the words that contained tag ‘RT’ followed by ‘@username’. In derived approach, to ,eliminate ‘RT@username’ using two steps by renaming ‘@username’ to AT_USER and finding a pattern ‘RT + AT_USER’ in the sentence and replacing it with the blank character. This reduces the size of dataset to 3,518 KB (81.66%) in comparison to original dataset size.

Table 9: Analysis on File Size and Data processing time using derived algorithm

Pre-Processing Tasks	FILE SIZE (KB)	FILE SIZE in %	PROCESSING TIME(sec)
Before Preprocessing	4,308	100 %	NA
After removing URLs	3,695	85.77%	1.15
Rename and removing of 'RT@username' from the tweets	3,518	81.66%	1.32
Filtering #Hashtags from tweets	3,442	79.90%	2.06
Removing repeated character	3,431	79.64%	2.42
Removing special character	3,420	79.39%	2.70

Also, by having aware of the symbol '#' (called Hashtag or number sign) followed by word gives no meaning to the word when applied to lexical resources. Therefore, by filtering '#Hashtags' from the dataset followed by the word which may give us sentiment meaning to the word(s). Once applied, the size is reduced to 3,442 (79.90%), and it took 2.60 seconds when filtering the '#Hashtags' from the dataset. The repeated as well as special characters are also filtered from that dataset which is not going to specify any positive or negative sentiment when applied to lexical resources like WordNet or SentiWordNet dictionary. The reduced size of the dataset was 3,420 KB (79.39 %) after preprocessing and filtering, which means approximately 7MB (Megabit) of noise was eliminated from the raw text. As shown in Figure 4, the file size declines when we filter the data from the dataset has been shown.

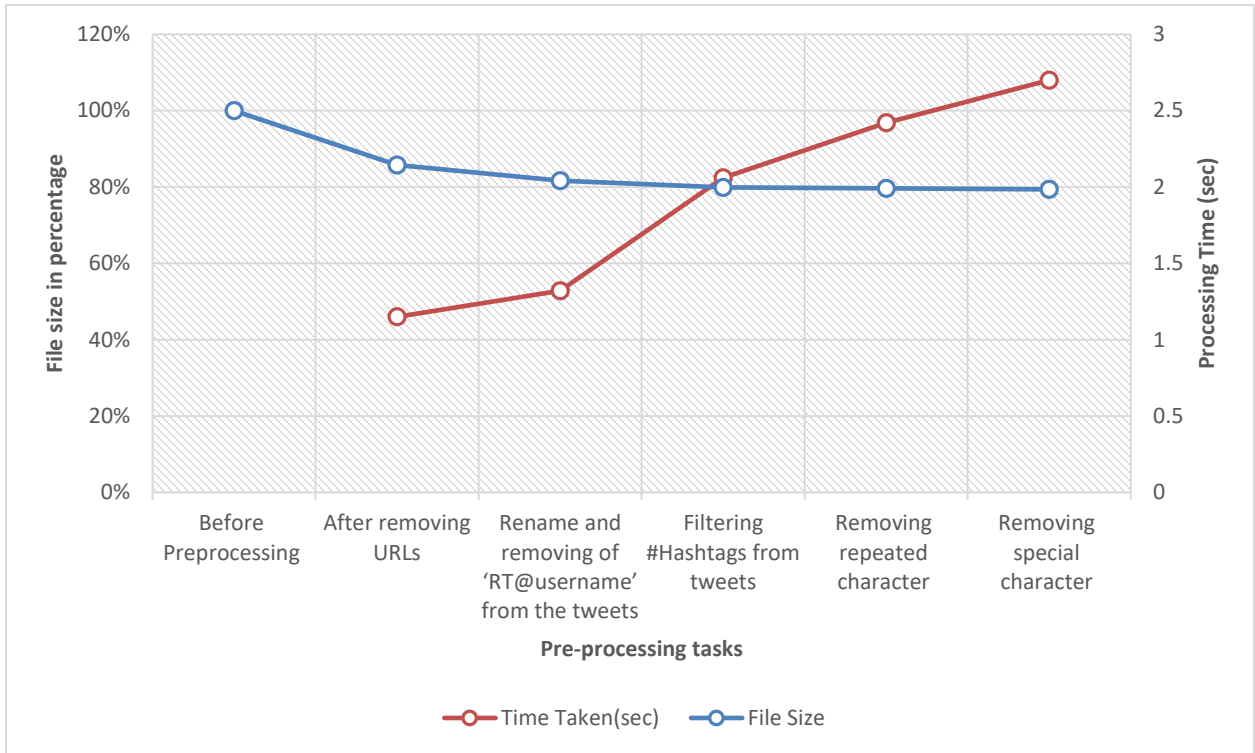


Figure 4: Reduction in file size after each pre-processing tasks

From Figure 4, the maximum time took is 2.70 seconds when filtering special characters and 2.42 seconds when filtering repeated characters. The reason for increase in time can be due to huge amount of unwanted text deployed to the tweeter platform or the text in other languages considered as a special character. Another thing can be pattern matching task and huge amount of these kind of characters in datasets may take longer time to process and as a result it will increase the load on the processor. Also, one can observe from Figure 4 that there was a dramatic drop in size to 30 % for dataset by removing URLs, RT@username and #Hashtags from the tweets. Moreover, it gives us clearer results after filtering tweets using this steps in developed algorithm and allows us for analysis sentiment from the tweets by applying core analysis by applying Natural Language Processing (NLP) concepts, which will be discussed in next Chapter.

Chapter 5

Linguistic Data Processing Using Natural Language Processing (NLP)

5.1 Introduction

The communication medium, which people use to interact with each other for some purposes is known as Natural Language, which can be English, French, Hindi or any other language. Communication by means of Language may be referred to as **linguistic communication** (Bird *et al.* 2009). This communication can be either written or verbal. Some forms of written communication are emails, social media blogs, letters, books or any other written form, which is either typed or hand written. Verbal forms of communication include voice over phone, lecture presentation or anything auditory. Moreover, every form of communication, whether written or verbal, has its own vocabulary, its structure, grammar, part-of-speech or all as a system. Therefore, processing of natural language can be categorized into two ways: firstly, by logically thinking and writing, and secondly, logically thinking and verbally communicating. Moreover, the term 'logically thinking or understanding' is defined as 'Natural Language processing', which we process in our mind

logically as a human, and a computer performs it using instructional steps written in programming language through Central Processing Units (CPU).

In the field of computer science, Natural language processing is a research field under artificial intelligence or computational linguistic, which focuses on the interaction between man-made natural language(s) and computers (Chowdhury, 2003). It is an active research area from the beginning of 21st century, and out of which the most common area is sentiment analysis using natural language processing, and the research domain influences the new areas for Machine Learning, Cognitive Science, and Computational Statistics (Firmino Alves *et al.* 2014). Machine learning techniques control the data processing by the use machine learning algorithm and classify the linguistic data by representing them into to vector form (Olsson *et al.* 2009).

It also affects the programming languages of computers, which allows programmer to interact with real world entity, and permits to process natural language by humans. These artificial languages (e.g. Python, C, C++, Java.) have their own structure, rules, words, notations. Therefore, processing human language(s) using the artificial languages can be referred to as Natural Language Processing (NLP) or Computational Linguistic (Bird *et al.* 2009). Therefore, the term Natural Language Processing involves a comprehensive set of techniques that allows automatic generation, manipulation and analysis of natural human languages.

Using Natural Language Processing (NLP) steps, one can process large amount of non-structured data by analyzing sentence structure, and can compute sentence or document level sentiment polarity with the use of famous linguistic database or lexical resources like

WordNet, SentiWordNet, and treebanks. (Bird *et al.* 2009). The techniques involved in processing natural language are POS (Part of Speech) tagging, parsing, named entity recognition, information extraction, word sense disambiguation, Word Stemming and lemmatization, stop word analysis, word tokenization, and many other depending upon research objective. During the evaluation process, punctuations between the lines are noted carefully and the expressions that are either idiomatic or colloquial are recognized, which helps in clarifying and understanding the “negations” that revises the word’s polarization depending upon the various types of parts of speech (nouns, prepositions adverbs, pronouns, adjectives, interjections, conjunctions and verbs) by taking into consideration the particular “functional-logic” statements (Neri *et al.* 2012). And this approach used for analyzing sentiment from linguistic data is known as Lexicon Based or Dictionary Based approach. The derived architecture for sentiment analysis using Natural Language Processing (NLP) shown in below Figure 5.

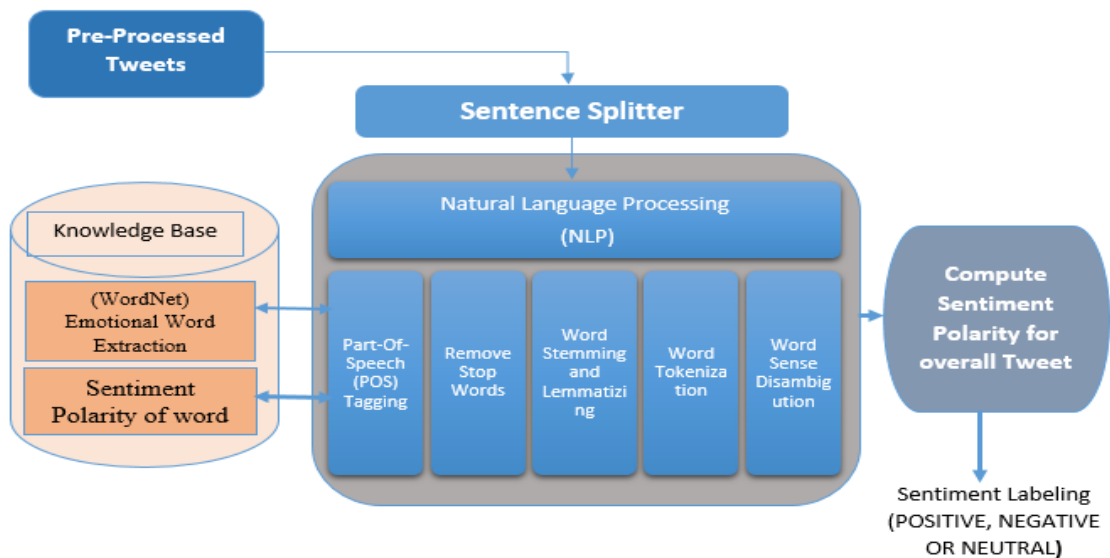


Figure 5: Derived architecture for Sentiment analysis using Natural Language Processing (NLP)

Here, the knowledge based tool is shown in Figure 5, is used to develop algorithm that analyzes each word in the context, and finds the related synsets (synonyms of word) and lemmas (in the domain it occurs) to achieve accuracy in the sentiment score obtained from the tweets. This derived architecture shown in above Figure 5 will be discussed in detail throughout this Chapter.

5.2 Natural Language Toolkit (NLTK)

Using Python for performing operation on strings involves very simple functions for language processing tasks. To achieve an advanced functionality for processing linguistic data, Natural Language Toolkit (NLTK) available for Python is used. NLTK is a collection of modules and corpora, released under GPL open-source license, which permits student to learn and to perform research in NLP (Bird *et al.* 2006). It has over 50 corpora and lexical resources like WordNet in combination with language processing libraries like work tokenization, classification and stemming, tagging, parsing and semantic rules for the analysis of text document, which will be discussed in detail (Bird *et al.* 2006). The key benefit of NLTK is that it is exclusively self-contained and has been praised by academic community (Bird *et al.* 2009). Also, it not only gives access to methods and packages for common NLP tasks, but also provides the pre-processed and raw versions of standard corpora used in NLP literature and courses (Bird *et al.* 2009).

5.3 Word Tokenization

After filtering the noise from that dataset, all that was left were raw words in the sentences. These words individually have some meaning and may consist of emotion or sentiment expressed by the user in the tweet. In Natural Language processing, the process or steps for breaking down sentences into words and punctuations is known as **Tokenization** (Bird *et al.* 2009). The goal for generating the list of words by separating the string is called Tokenizing sentence into words (Perkins 2010). Here, to tokenize the words Natural Language Toolkit (NLTK) tokenize package is used. The choice for selecting tokenizer depends on the characteristic of data you are working on and the language. Here, to create a tokenizing method to tokenize the words using Tweet Tokenizer module for processing English language terms. The algorithm for word Tokenization using Tweet Tokenizer is shown in below Table 10.

Table 10: Algorithm for Word Tokenization

Input: Filtered Tweets Output: Tokenize words
For all words in Processed Tweets Tokenize the word passing to Tweet Tokenizer Method and append Tokenize Sentence Return Tokenize Sentence

The result obtained after tokenizing word is shown below in Table 11:

Table 11: Example on Word Tokenization

	Text Data
Processed Tweet	The Best World Cup Song So Far READY FOR BRAZIL Very GOOD World Cup Song Worldcup2014 Brazil2014
Word Tokenization	['The', 'Best', 'World', 'Cup', 'Song', 'So', 'Far', 'READY', 'FOR', 'BRAZIL', 'Very', 'GOOD', 'World', 'Cup', 'Song', 'World cup', '2014', 'Brazil', '2014']

5.4 Word Stemming

The stemming and lemmatizing of words are the approaches that produces the normalized form of a word (Toman *et al.* 2006) in the text. According to (Younis *et al.*2015) word stemming is a technique that gets the root (base) of the word in the text. It normalize the word by removing the suffix from the word, which gives root meaning for the word. There are many stemming algorithms available openly to perform word stemming. In this approach of data pre-processing, the Porter Stemmer algorithm is used for stripping suffix from the word to retrieve proper meaning from the text.

Porter Stemmer algorithm originally was published by M.F. Porter (1980), and the algorithm was developed using BCPL (Basic Combined Programming Language) for removing suffix from the word automatically (Porter 1980). It gives the root meaning to the word in text by removing various suffix like –ED, -ING,-ION, IONS by stemming methodology and gives more abstract meaning to the word (Porter 1980). To implement this functionality for stemming the words, in this research, the Porter Stemmer algorithm available in Python NLTK stem package is used and developed a function that returns the word after stemming all the characters in a word. Porter Stemmer stems the word, character by character, and removes suffix and gives the base meaning to the word. Here, during the stemming process the word will stemmed and return the root meaning of the word. To achieve accuracy in sentiment analysis, only stemming the word whose length is greater than two, as the word like ‘a’, ‘is’, ‘OH’, are not taken into consideration when applied to sentiment dictionary for getting polarity of the word. The algorithm for performing word stemming is demonstrated in below Table 12.

Table 12: Algorithm for word Stemming and Lemmatizing

Input: Tokenize words
Output: stemmed and lemmatized words
For word in word Tokens
Initialize StemmedSentence variable to empty list
If length of word greater than 2
Method call for stemming the word using PorterStemmer object.
Method call for Lemmatizing the word using WordNetLemmatizer object
Append StemmedSentence list
Return Stemmed Sentence List

In this algorithm, the stemming of the word whose length is greater than two and append the word in the variable type list to avoid the returning of single character. And the returned stemmed word will be in more generic form and can be used in the further steps of natural language processing task. Furthermore, word stemming and lemmatizing gives common base form of word by removing the suffix from the word which gives the dictionary meaning to the word. In example given in Table 13, the word “it’s” becomes “it” by stemming the letter “s”. Also, the words like ‘connected, connecting, connects’ stemmed to the single word ‘connect’ which is the more generic form of the word. This will give us more generic sentiment score when applied to lexical dictionary and helps us to evaluate accurate sentiment polarity for textual tweets. The complete result obtained by using this algorithm is shown in Table 13.

Table 13: Example on Word Stemming

Example: Word Stemming	Text Data
Word Tokenization	[I, 'am', ' connected ', 'with', 'world', 'cup', 'and', "it's", 'GOOD', ' Connecting ', 'each', 'other', 'with', 'team', 'World', 'Cup', 'Song', ' connects ', 'Worldcup', '2014', 'Brazil', '2014']
Stemmed and tokenized Tweet	[' connect ', 'with', 'world', 'cup', 'and', "it'", 'good', ' connect ', 'each', 'other', 'with', 'team', 'world', 'cup', 'song', ' connect ', 'worldcup', '2014', 'brazil', '2014']

5.5 Word Lemmatization

Another important natural language processing task is word lemmatization. It is a technique that transforms the structural form of a word to the base or dictionary form of word by filtering the affixation or by changing the vowel from the word. The outcome achieved from the word is known as **lemma** (Liu *et al.* 2012). Lemmas are the word that has a root meaning of the term requested and the lemmatized word are the gateway to the WordNet (Bhattacharyya *et al.* 2014). Therefore, lemmatizing the word using algorithm will create a lemma which will further pass to WordNet dictionary that pulls out the sense of the word and its sense number, which is the objective for getting better sentiment score for the word. Here, for lemmatizing words by matching character by character using “WordNetLemmatizer” class available through ‘wordnet’ class of stem package in Python NLTK. It is a good choice to use for getting effective lemmas and generating vocabulary from the text (Bird *et al.* 2009). To achieve effective lemma or root meaning of the word using “WordNetLemmatizer”, it is really important that input word must be passed in lower case to the “WordNetLemmatizer” algorithm to achieve accuracy. Therefore, the lowercase word is passed to the function, which are greater than two to retrieve effective lemmas word from “WordNetLemmatizer” class. An example is shown in below Table 14. The word ‘women’ is lemmatized to form as ‘woman’, which is a root meaning in the ‘wordnet’ dictionary.

Table 14: Example on Word Lemmatizing

Example: Word Lemmatizing	Text Data
Word Tokenization	['I', 'am', 'children', ' women ', 'swimmer', 'and', 'I', 'like', 'swimming']
Lemmatized Tweet	['child', ' woman ', 'swimmer', 'and', 'like', 'swim']

5.6 Removing Stop words

While processing natural language, some of the words which have high occurrence in the document are stop words (e.g. ‘and’, ‘the’, ‘am’, ‘is’), which only have little emotional meaning and it do not affect the sentiment score when applied to lexical resources. Therefore, it is common practice by many researchers to filter stop words in the domain of analyzing sentiment from the document. According to (Saif *et al.* 2012), in their experiment, they compared both the results of keeping the stop words in the text as well as, by filtering stop words from the text, and the result obtained has high accuracy in sentiment classification for keeping stop words as is in the document. In the literature (Firmino Alves *et al.* 2014) and (Carvalho *et al.* 2014) it is shown that an approach for classification of text by removing stop words from the text and achieved accuracy in the calculation of sentiment polarity and obtained interesting results. In the book by (Bird *et al.* 2009), they explain that stop words may contain little vocabulary content and they also suggested that filtering stop words is necessary before performing another processing tasks. Therefore, keeping the idea of (Saif *et al.* 2012), both the approaches to compare sentiment classification, with and without stop words in the Tweets is taken into consideration. The result obtained after

removal of stop words from the Tweets are shown in Table 15. However, the result obtain from keeping stop words in the document has more accuracy in the result.

Table 15: Example on Removing of Stop Word

Example: Filtering Stop words	Text Data
Word Tokens without filtering stop words	['I', 'am', 'connected', 'with', 'world', 'cup', 'and', "it's", 'GOOD', 'Connecting', 'each', 'other', 'with', 'team', 'World', 'Cup', 'Song', 'connects', 'Worldcup', '2014', 'Brazil', '2014']
Word Tokens with filtering Stop Words	['connect', 'world', 'cup', 'GOOD', 'Connect', 'team', 'World', 'Cup', 'Song', 'connect', 'Worldcup', '2014', 'Brazil', '2014']

In the result above one can see how the stop words like 'I', 'am', 'with', 'and', 'it's', 'each', 'other' and 'with' are filtered from the Tweet sentence. The filtered words may contain sentiment meaning of the word when applied to lexical resources to retrieve sentiment polarity from it. Therefore, to test the objective the stop words will be kept for now, and at the end it will be discarded from Natural Language Processing task. The results obtained will be measured with accuracy and consistency in the analysis in presence of stop words in the data set.

5.7 Part-of-Speech (POS) Tagging

This Technique annotate the part-of-speech (e.g. Noun, Adverb, Adjective, Subjects, Objects) to the words analyzing the sentence structure, and creates the raw form of word sense disambiguation (Pang *et al.* 2008). According to (Kouloumpis *et al.* 2011), it is the last step in natural language processing for analyzing sentiment from the sentence. By performing this step, one can obtain featured words that represents the sentence structure and the meaning of the words in the domain it belongs to in the sentence. To achieve annotated part-of-speech in the approach used, POS tagger class available in the NLTK package has been used to develop algorithm to obtain word sense for only English language tags from the sentence. It analyzes the lowest level of syntactic structure of the sentence and tags them with their related part-of-speech, which categorizes the word lexically with its POS label and gloss together for further classification. The Table 16 below shows the annotated words with its POS tags (e.g. 'NN-NOUN', 'IN-Proposition or subordinating conjunction', 'NNP-Proper Noun singular') to each word in the sentence. The abbreviation for the Part-of-Speech (POS) tags has been described in Appendix A.

Table 16: Example of Part-Of-Speech (POS) Tagging

Example: POS Tagging	Text Data
Word Tokens	['connect', 'with', 'world', 'cup', 'and', 'it', 'GOOD', 'Connect', 'each', 'other', 'with', 'team', 'World', 'Cup', 'Song', 'connect', 'Worldcup', '2014', 'Brazil', '2014']
POS(Part-of-Speech)-Tagged sentence	[('connect', 'NN'), ('with', 'IN'), ('world', 'NN'), ('cup', 'NN'), ('and', 'CC'), ('it', 'VB'), ('GOOD', 'JJ'), ('Connect', 'NNP'), ('each', 'DT'), ('other', 'JJ'), ('with', 'IN'), ('team', 'NN'), ('World', 'NNP'), ('Cup', 'NNP'), ('Song', 'NNP'), ('connect', 'NN'), ('Worldcup', 'NNP'), ('2014', 'CD'), ('Brazil', 'NNP'), ('2014', 'CD')]

Here, in Table 16, the word with its part-of-speech *connect* is NN (a Noun), *with* is IN (a preposition), *and* is CC (a coordinating conjunction), *good* is JJ (an adjective), *song* is NNP (a proper Noun singular) and *2014* is CD (a cardinal number). Nouns are generally refer to the people, place, things or the concepts, verbs are words describing events and actions, Adjective and Adverbs are the two important classes, where adjective describe the nouns and can be used as a modifier. Adverbs modify verbs to specify the time, manner, place or direction of the event describe by the verb.

After tagging part of speech to each word in the sentence, it is necessary to structure the sentence in order to achieve accuracy in sentiment polarity for the sentence. Supposedly, what if negative word falls inside the sentence and gives positive sentiment polarity to the sentence? For example, the sentence “I do not like or enjoy this movie.”, where positive sentiment is assigned because of occurrence of the words “like” and “enjoy”, which give high positive sentiment score when applied to lexical resources (like SentiWordNet or WordNet Affect). In fact, it is negative sentence due to occurrence of word “not” in the sentence. In other words, the sense of the word that occurs after negation word changes the

meaning and sentiment score of the particular word, the overall polarity is taken into account (Kumar *et al.* 2015). As a result, it will be effect the accuracy in assigning the sentiment polarity to the sentence; i.e. positive, negative or neutral. For example, in the sentence ‘*I do not like to watch this game it is not interesting*’, the word ‘*do not*’, ‘*not*’ is the negation word, which change the meaning of the sentence. To solve this problem, a way is to change the meaning of the word to opposite (antonym of word) if the word is followed by a negation word. In the sentence ‘*I do not like to watch this game , it is not interesting*’ the word ‘*like*’ will replace by ‘*dislike*’ and the word ‘*interesting*’ will replace by ‘*uninteresting*’ . Hence, to analyze this sentence using lexical resources, it will provide higher total negative sentiment score for the sentence. Therefore, to achieve accuracy in sentiment analysis, we developed an algorithm derived in Table 17 that reverses the sentiment score, if the word(s) sense in the sentence refers to negative meaning (for example: do not, not, did not, cannot) and occurrence of this words in sentence.

Table 17: Algorithm for Marking Negation Words

<p>Input: stemmed and lemmatized words</p> <p>Output: negation tagged word ‘1’ for negative reference word and ‘0’ for positive reference word</p>
<p>List <i>mark_negation</i> by modifying the word with tag ‘_NEG’ using mark negation method</p> <p>Initialize <i>Total_Mark_List</i></p> <p>For <i>neg_mark</i> in <i>mark_negation</i></p> <p> Parse last 4 character in the <i>neg_mark</i> is ‘_NEG’</p> <p> If parsed word contain the tag ‘_NEG’</p> <p> Partition ‘_’ from the tag ‘_NEG’ to tail word</p> <p> If tail word contains ‘NEG’</p> <p> Append <i>Total_Mark_List</i> to 1</p> <p> Else</p> <p> Continue</p> <p> Else</p> <p> <i>neg_mark</i></p> <p> Append <i>Total_Mark_List</i> to 0</p> <p>Return <i>Total_Mark_List</i></p>

The algorithm above for taking negation words into account for the analysis, which will refer to mark negation module available in sentiment utility under NTLK package. This method assign the ‘_NEG’ tag for the words which are followed by the negation words. In algorithm, first the word tag ‘_NEG’ to the word is assigned that falls after the negation word in the sentence. During the sentence analysis, if the word like ‘not’, didn’t, do not., are appeared in the sentence all the word until last word of sentence are classified with tag ‘_NEG’. The result of first step is stored in the list for further analysis, that will further provide negation score (1 or 0) to the word. In the second step, by iterating the results obtained in a first step and parse the word with the tag ‘_NEG’ and assign the score 1 to the negation tagged word and 0 to the word without tag. The word with score ‘1’ will return the list of lemmas that contains antonym to the original word and will reverse the meaning as well the polarity of the sentence when applied to the lexical resources to achieve accuracy in sentiment analysis from the Twitter data.

is '1' which has negative meaning sense in the sentence. Therefore, this algorithm works perfectly fine for analyzing the negation words and sense of words followed by it.

The vector form '1' or '0' obtained in last step represents the sense of the words in the sentence, which will refer to the meaning of the word in current context when negation words comes in. And the result obtained will be utilized in combination with the result of POS tagging step to achieve a unique flavor or the objective for achieving the accuracy in result for assigning the polarity to the sentence by further developing steps in overall algorithm.

5.8 WordNet

WordNet databases are complex and functional that allow retrieving information in the field of linguistic data processing (Lam *et al.* 2014). One of most popular and well-mannered resources made for processing natural language contains emotional words as well as the “sematic relationships” among words (Ohana *et al.* 2009). This interconnection of semantic and lexical relationship for the words and its meaning are known as **Synsets or Synonyms set or group of synonyms**. According to (Wawer *et al.* 2010) the WordNet database consist of 150,000 words, which is organized in over 115,000 synsets having a pair of word-sense are 207,000 in the year 2006. In the book (Bird *et al.* 2009) says that the WordNet lexical resource contains 155, 287 words and 117,659 synset (similar meaning word) records by year 2009.

WordNet form of lexical databases are commonly used by Dictionary (Lexicon) based approach, which automatically generates the dictionary of the words and its relationship in proper size. To generate Dictionary, one approach suggested by (Augustyniak *et al.* 2015) is to produce a set of professional nominated emotions from the text and group those emotions by using vocabularies or the lexical resources like WordNet. In the literature by (Pang *et al.* 2008) this approach relates to the “data-driven” approach for generating dictionary like WordNet. In data-driven technique, the words are shared form of information, and frequency of words are grouped together with seed words that iterate through the synonyms and antonyms using WordNet lexical resources (Pang *et al.* 2008). Much of the work cited above focuses on identifying the *prior polarity* of terms or phrases, to use before assigning the sentiment polarity to the word using WordNet lexical resource. Moreover, due to absence of sentiment knowledge in the WordNet database it is not likely to be used directly to compute sentiment polarity (Wawer *et al.* 2010). WordNet lexicon assign the expressions, positioning the semantic meaning to the word and prepare the information into the context for further identifying the accurate sentiment polarity to word which convey its specific emotional content.

As an objective, an algorithm was developed to classify the correct synset word and its part-of-speech was further used to obtain a most accurate sentiment score when applied to lexical resources. The designed algorithm in Table 19 gives accurate synsets (all synonyms), lemmas (head word), the antonyms as well as part-of-speech (POS) tag which most accurately relates to the term. The discussion on this algorithm is discussed in detail throughout this Chapter.

Table 19: Algorithm extract emotional words from tweet

<p>Input: POS (Part-of-speech) tagged word, negation marks ('1' for Negative or '0' for Positive)</p> <p>Output: A unique synset word with its part of speech and close meaning to the word.</p>
<p>Method GetSynset by passing POS tag word and Negation mark</p> <p>Method to Sanitize part-of-speech (POS) tag to WordNet accepted POS</p> <p>For synset in WordNet Synsets (word, POS tag):</p> <p>Returns list of synsets for the words</p> <p>For lemma in synset list:</p> <p>If word equals to lemma name</p> <p>Append Synonyms(word with the same meaning) list</p> <p>If word has its Antonyms</p> <p>Append Antonyms(word with opposite meaning) list</p> <p>If negation mark is '0' and it is not NULL</p> <p>Return first synonym of word and POS tag from Synonyms list</p> <p>Else</p> <p>Return the same word and POS requested</p> <p>Else IF negation mark is '1' and it is not NULL</p> <p>Return first antonyms of word and POS tag from Synonyms list</p> <p>Else</p> <p>Return the same word and POS requested</p>

In order to explore the words from the tweets and to evaluate emotional words and its relationship using the WordNet object by importing the 'wordnet' class from 'nltk corpus' module, NLTK is available. The WordNet dictionary returns synonyms (Synsets) or the antonyms for the word, part-of-speech and its sense number for the requested corpus. Firstly, it sanitizes the word's part-of-speech to standardize the POS tags for WordNet. To do so, a method that sanitizes the part-of-speech has been developed that tags for all POS

tags starting with letter ‘V’ to ‘WordNet Verb’, ‘N’ to ‘WordNet NOUN’, ‘J’ to ‘WordNet ADJECTIVE’, ‘R’ to ‘WordNet ADVERB’ and for others, they were tagged to ‘NONE’ and modified the string pair to the word string and newly annotated part-of-speech (POS). The result for the sanitize method after sanitizing the POS tags is shown in Table 20.

Table 20: Example of Word Sanitization

Example: Sanitize	Word	Text Data
POS sentence	Tagged	[(I, 'PRP'), ('am', 'VBP'), ('connect', 'JJ'), ('with', 'IN'), ('world', 'NN'), ('cup', 'NN'), ('and', 'CC'), ('it', 'VB'), ('GOOD', 'JJ'), ('Connect', 'NNP'), ('each', 'DT'), ('other', 'JJ'), ('with', 'IN'), ('team', 'NN'), ('World', 'NNP'), ('Cup', 'NNP'), ('Song', 'NNP'), ('connect', 'NN'), ('Worldcup', 'NNP'), ('2014', 'CD'), ('Brazil', 'NNP'), ('2014', 'CD')]
Sanitized POS tags with word		(I , None) (am , v) (connect , a) (with , None) (world , n) (cup , n) (and , None) (it , v) (GOOD , a) (Connect , n) (each , None) (other , a) (with , None) (team , n) (World , n) (Cup , n) (Song , n) (connect , n) (Worldcup , n) (2014 , None) (Brazil , n) (2014 , None)

One can observe from the above example that all the words are tagged with the more generic POS tags and the words which are not in wordnet tags are set to ‘NONE’. The words which are tagged ‘NONE’ in the example, it does not return any sentiment or emotional characteristics and therefore, during further analysis it will be neglected if it does not contain any sentiment score. Further in the next step, the possible synset terms were obtained for the given word and analyze the synsets for the given words by iterating through the loop and find a correct lemma for the given word in the synsets. After performing the processing of the word and POS tag to obtain Synset, the list of analyzed synset term obtained is shown by the example in Table 21.

Table 21: Example on WordNet Synset

Example: Synsets for word	Text Data
Sanitized POS tags with word	(I , None) (am , v) (connect , a) (with , None) (world , n) (cup , n) (and , None) (it' , v) (GOOD , a) (Connect , n) (each , None) (other , a) (with , None) (team , n) (World , n) (Cup , n) (Song , n) (connect , n) (Worldcup , n) (2014 , None) (Brazil , n) (2014 , None)
Synsets obtained for each word followed by POS tag and sense number #	<p>[Synset('iodine.n.01'), Synset('one.n.01'), Synset('i.n.03'), Synset('one.s.01')]</p> <p>[Synset('be.v.01'), Synset('be.v.02'), Synset('be.v.03'), Synset('exist.v.01'), Synset('be.v.05'), Synset('equal.v.01'), Synset('constitute.v.01'), Synset('be.v.08'), Synset('embody.v.02'), Synset('be.v.10'), Synset('be.v.11'), Synset('be.v.12'), Synset('cost.v.01')]</p> <p>[Synset('universe.n.01'), Synset('world.n.02'), Synset('world.n.03'), Synset('earth.n.01'), Synset('populace.n.01'), Synset('world.n.06'), Synset('worldly_concern.n.01'), Synset('world.n.08')]</p> <p>[Synset('cup.n.01'), Synset('cup.n.02'), Synset('cup.n.03'), Synset('cup.n.04'), Synset('cup.n.05'), Synset('cup.n.06'), Synset('cup.n.07'), Synset('cup.n.08')]</p> <p>[Synset('good.a.01'), Synset('full.s.06'), Synset('good.a.03'), Synset('estimable.s.02'), Synset('beneficial.s.01'), Synset('good.s.06'), Synset('good.s.07'), Synset('adept.s.01'), Synset('good.s.09'), Synset('dear.s.02'), Synset('dependable.s.04'), Synset('good.s.12'), Synset('good.s.13'), Synset('effective.s.04'), Synset('good.s.15'), Synset('good.s.16'), Synset('good.s.17'), Synset('good.s.18'), Synset('good.s.19'), Synset('good.s.20'), Synset('good.s.21')]</p> <p>[Synset('each.s.01'), Synset('each.r.01')]</p> <p>[Synset('other.a.01'), Synset('other.s.02'), Synset('early.s.03'), Synset('other.s.04')]</p> <p>[Synset('team.n.01'), Synset('team.n.02')]</p> <p>[Synset('universe.n.01'), Synset('world.n.02'), Synset('world.n.03'), Synset('earth.n.01'), Synset('populace.n.01'), Synset('world.n.06'), Synset('worldly_concern.n.01'), Synset('world.n.08')]</p> <p>[Synset('cup.n.01'), Synset('cup.n.02'), Synset('cup.n.03'), Synset('cup.n.04'), Synset('cup.n.05'), Synset('cup.n.06'), Synset('cup.n.07'), Synset('cup.n.08')]</p> <p>[Synset('song.n.01'), Synset('song.n.02'), Synset('song.n.03'), Synset('birdcall.n.01'), Synset('song.n.05'), Synset('sung.n.01')]</p> <p>[Synset('brazil.n.01'), Synset('brazil_nut.n.02')]</p>

Therefore, synsets obtained for the given word from the WordNet dictionary are attached with its POS tag and the word sense number as shown in Table 21 above. The word sense number is the WordNet sense index, for which the most related synset for the word can be fetched from the WordNet database. Further from all the synsets obtained, it was analyzed using each synset to obtain lemmas by parsing the synset. The lemmas are the head word or the domain of the word from which it belongs to as well as it contains additional information like part of speech and sense definition (Bird *et al.* 2009). In the example shown below in Table 22, shows the list of lemmas for the synset of the word ‘good’ that contains the lemmas from all the domain it belongs to. Here the last expression or the term in the lemmas are the lemmas name, which will compare the lemma name to the input word. Once the lemmas name matches to the requested (input) word, it will be appended to all possible synonyms or antonyms for the matched cases and further classify for the negation marks to obtain correct lemma word to obtain sentiment score.

Table 22: Example on word lemmas

<p>Example: Obtaining Lemmas (Head word) from the synsets</p>	<p>Text Data</p> <p>The example shown for the synset term ‘good’.</p>
<p>Synsets obtained for each word followed by POS tag and sense number #</p>	<p>[Synset('good.a.01'), Synset('full.s.06'), Synset('good.a.03'), Synset('estimable.s.02'), Synset('beneficial.s.01'), Synset('good.s.06'), Synset('good.s.07'), Synset('adept.s.01'), Synset('good.s.09'), Synset('dear.s.02'), Synset('dependable.s.04'), Synset('good.s.12'), Synset('good.s.13'), Synset('effective.s.04'), Synset('good.s.15'), Synset('good.s.16'), Synset('good.s.17'), Synset('good.s.18'), Synset('good.s.19'), Synset('good.s.20'), Synset('good.s.21')]</p>
<p>Lemmas for the Synsets</p> <p>Here the last or end word are known as 'lemmas name'</p>	<p>Lemma('good.a.01.good') Lemma('full.s.06.full') Lemma('full.s.06.good') Lemma('good.a.03.good') Lemma('estimable.s.02.estimable') Lemma('estimable.s.02.good') Lemma('estimable.s.02.honorable') Lemma('estimable.s.02.respectable') Lemma('beneficial.s.01.beneficial') Lemma('beneficial.s.01.good') Lemma('good.s.06.good') Lemma('good.s.07.good') Lemma('good.s.07.just') Lemma('good.s.07.upright') Lemma('adept.s.01.adept') Lemma('adept.s.01.expert') Lemma('adept.s.01.good') Lemma('adept.s.01.practiced') Lemma('adept.s.01.proficient') Lemma('adept.s.01.skillful') Lemma('adept.s.01.skilful') Lemma('good.s.09.good') Lemma('dear.s.02.dear') Lemma('dear.s.02.good') Lemma('dear.s.02.near') Lemma('dependable.s.04.dependable') Lemma('dependable.s.04.good') Lemma('dependable.s.04.safe') Lemma('dependable.s.04.secure') Lemma('good.s.12.good') Lemma('good.s.12.right') Lemma('good.s.12.ripe') Lemma('good.s.13.good') Lemma('good.s.13.well') Lemma('effective.s.04.effective') Lemma('effective.s.04.good') Lemma('effective.s.04.in_effect') Lemma('effective.s.04.in_force') Lemma('good.s.15.good') Lemma('good.s.16.good') Lemma('good.s.16.serious') Lemma('good.s.17.good') Lemma('good.s.17.sound') Lemma('good.s.18.good') Lemma('good.s.18.salutary') Lemma('good.s.19 .good') Lemma('good.s.19.honest') Lemma('good.s.20.good') Lemma('good.s.20.undecomposed') Lemma('good.s.20.unspoiled') Lemma('good.s.20.unspoilt') Lemma('good.s.21.good')</p>

Once all the synonyms and the antonyms for the lemmas is obtained the selected lemma based on the negation mark. If the negation mark is '1', will return antonyms for the given lemma in synset with its POS and if negation mark is '0' will return most accurate synonyms for the words and POS tag. Therefore, to obtain synset term based on the negation mark and the most accurate lemmas name with it POS tag that gives an accurate sentiment score when applied to SentiWordNet database.

5.9 SentiWordNet Dictionary

SentiWordNet is a resource that consists of opinion information for the word extracted from the WordNet database where each term is assigned with its numerical scores that contain sentiment value for the word and the gloss (information) associated with the word (Ohana *et al.* 2009). It has been constructed with information attached to the synset term, which is built on quantitative analysis concept and it denotes the vector representation via semi-supervised synset classification methods (Esuli *et al.* 2006). Also according to (Ohana *et al.* 2009) formed based on semi-automated process which can be easily upgraded for the later version of WordNet and also for the language whose lexicons are available. SentiWordNet Dictionary is publically available for the research or academic purpose which permits access to sentiment information for the English language and can be used to develop an automated sentiment evaluation as well as it is mostly rely on the knowledge obtained from the WordNet (Taboada *et al.* 2011). This Dictionary provides the cluster of

synonymous words to be used for analyzing the sentiment for the given word and POS tag attached to it.

The extracted opinionated term from the WordNet database in section 5.8 will be assigned with the numerical score using SentiWordNet dictionary. Each set of terms distribution to the similar meaning in SentiWordNet (*synsets*) is associated with two numerical scores ranging from 0 to 1, each value indicates the synsets positive and negative bias. The scores return the agreement amongst the classifier group on the positive or negative label for a term, thus one distinct aspect of SentiWordNet is that it is possible for a term to have non-zero values for both positive and negative scores. Sample entries in the SentiWordNet dictionary can be found in Table 23.

Table 23: Example of SentiWordNet Dictionary structure

POS	ID	PosScore	NegScore	SynsetTerms	Gloss
a	02343110	1	0	splendid#2 first-class#1 fantabulous#1 excellent#1	very good; of the highest quality; "made an excellent speech"; "the school has excellent teachers"; "a first-class mind"
a	01251128	0	0.75	cold#1	having a low or inadequate temperature or feeling a sensation of coldness or having been made cold by e.g. ice or refrigeration; "a cold climate"; "a cold room"; "dinner has gotten cold"; "cold fingers"; "if you are cold, turn up the heat"; "a cold beer"
n	05015117	0	0.125	low_temperatur e#1 frigidness#2 frigidity#2 coldness#3 cold#2	the absence of heat; "the coldness made our breath visible"; "come in out of the cold"; "cold is a vasoconstrictor"
n	05142180	0.625	0	goodness#1 good#3	that which is pleasing or valuable or useful; "weigh the good against the bad"; "among the highest goods of all are happiness and self-realization"
n	05159725	0.5	0	good#1	Benefit; "for your own good"; "what's the good of worrying?"
r	00011093	0.375	0	well#1 good#1	(often used as a combining form) in a good or proper or satisfactory manner or to a high standard ('good' is a nonstandard dialectal variant for 'well'); "the children behaved well"; "a task well done"; "the party went well"; "he slept well"; "a well-argued thesis"; "a well-seasoned dish"; "a well-planned party"; "the baby can walk pretty good"
r	00013626	0.125	0.25	well#12 comfortably#3	in financial comfort; "They live well"; "she has been able to live comfortably since her husband died"

In SentiWordNet database shown in Table 23, all the WordNet synsets are classified in a way that it consists of two numerical score that defines the positivity as well as the negativity of the terms in combination with POS tag and sense number contained in the WordNet synset term. This will add real value sentiment score for each synset from WordNet database and allow us to label the sentiment polarity (positive, negative or neutral) for the requested word. The advantage of SentiWordNet is that it uses semantic resources to enhance the structure of the lexicon and for assignment of positive and negative scores for a single word attached with sense number. In this research, to fetch the Sentiment score from SentiWordNet using the sentence level sentiment classification or lexicon based approach.

SentiWordNet is a lexical resource for the English language. In this dictionary, each entry refers to a group of words of the same Part-of-Speech (POS) and with the same sense (meaning). Each group is associated to three sentiment numerical scores, which describe how positive, negative, or Neutral the words contained in it are. Such scores range from 0.0 to 1.0, and their sum is 1.0 for each group. The word “excellent”, e.g., is only categorized as adjective, and has a positive score of 1.0 and negative 0 as shown in Table 23. The word “cold”, in turn, has a negative score of 0.75, in the sense of “having a low or inadequate temperature” (adjective), and a negative score of 0.125, in the sense of “a mild viral infection” (noun). Some words may also have both positive and negative scores, such as example in Table 23 the adverb “well”, in the sense of “in financial comfort”, with 0.125 and 0.25 as positive and negative scores, respectively. The example of the sentiment scored for the tweet is used throughout the discussion and is illustrated in below Table 24.

Table 24: Assigning Sentiment Polarity to the Word

Example: Assigning Polarity using SentiWordNet	Text Data
Input: Sanitized POS tags with word	(I , None) (am , v) (connect , a) (with , None) (world , n) (cup , n) (and , None) (it' , v) (GOOD , a) (Connect , n) (each , None) (other , a) (with , None) (team , n) (World , n) (Cup , n) (Song , n) (connect , n) (Worldcup , n) (2014 , None) (Brazil , n) (2014 , None)
Output: Sentiment Score for the synset term obtained from the SentiWordNet Database	<i.n.01: PosScore=0.0 NegScore=0.0> <be.v.01: PosScore=0.25 NegScore=0.125> <universe.n.01: PosScore=0.0 NegScore=0.0> <cup.n.01: PosScore=0.0 NegScore=0.0> <good.a.01: PosScore=0.75 NegScore=0.0> <each.s.01: PosScore=0.0 NegScore=0.0> <other.a.01: PosScore=0.0 NegScore=0.625> <team.n.01: PosScore=0.0 NegScore=0.0> <universe.n.01: PosScore=0.0 NegScore=0.0> <cup.n.01: PosScore=0.0 NegScore=0.0> <song.n.01: PosScore=0.0 NegScore=0.0> <brazil.n.01: PosScore=0.0 NegScore=0.0>

The sentiment score obtained for all the related terms or word with its POS tag followed by sense number has its PosScore and NegScore for each word associated with every synset term. The evaluated positive and negative term score from SentiWordNet to determine sentiment orientation for each term in the sentence or tweet. Here, in this approach, firstly, list the sentiment score for the first synset in synsets list. The score obtained for the each synset term is based on the context or the occurrence of the in the given sentence. Here, the word ‘good’ defines the positive opinion and has a score 0.25 and the word ‘be’ in this context has 0.25 positive and 0.125 negative sentiment score. Whereas, the word ‘other’ define the negative opinion with score 0.625. Therefore, it aggregates the sentiment score for all the terms or words together, which identifies overall sentiment polarity of the

sentence. The equation (1) and (2) calculates total positive score (TotalPosScore) and total negative score (TotalNegScore) where s is the sentiment $PosScore_s$ and $NegScore_s$ for the single term, and t is the sum up in each iteration for all the words in the tweet.

$$TotalPosScore_t = \sum_{s=1}^n TotalPosScore + PosScore_s \quad (1)$$

$$TotalNegScore_t = \sum_{s=1}^n TotalNegScore + NegScore_s \quad (2)$$

Further, for each tweet that sum up the total positive score and total negative score is than compared for labeling the sentiment whether it is 'POSITIVE', 'NEGATIVE' or 'NEUTRAL'. The equation (3) shows how the overall sentiment polarity $Polarity_{swn}(t)$ for the tweet t is predicted:

$$Polarity_{swn}(t) = \begin{cases} \text{POSITIVE or } 1, & \text{if } TotalPosScore(t) > TotalNegScore(t) \\ \text{NEGATIVE or } -1, & \text{if } TotalPosScore(t) < TotalNegScore(t) \\ \text{NEUTRAL or } 0, & \text{otherwise} \end{cases} \quad (3)$$

Here, the sentiment score $Polarity_{swn}(t)$ obtained for the tweet t using SentiWordNet database provides three measures that determine sentiment of the user tweet t . The tweet t is 'POSITIVE' or '1' if the total positive score is greater than total negative score, if total negative score is greater than the overall sentiment is 'NEGATIVE' or '-1' for tweet t else it is 'NEUTRAL' or '0' opinionated tweet t . Finally, the data set was generated for the sentiment polarity for the Twitter data and the results obtained appended to the data set looks like:

Table 25: Example of Output for Sentiment Analysis

Text Data	Total POS Score	Total NEG Score	Sentiment Polarity
['I', 'am', 'connect', 'with', 'world', 'cup', 'and', 'it', 'GOOD', 'Connect', 'each', 'other', 'with', 'team', 'World', 'Cup', 'Song', 'connect', 'Worldcup', '2014', 'Brazil', '2014']	1.0	0.75	POSITIVE
['MATCHDAY', 'arg', 'v', 'bel', 'argbel', 'WorldCup', '2014', 'TousEnsembl']	0.0	0.0	NEUTRAL
['I', 'am', 'child', 'woman', 'swimmer', 'and', 'I', 'like', 'swim']	0.375	0.125	POSITIVE
['I', 'don't', 'enjoy', 'thi', 'game', 'it', 'wa', 'disgust', 'and', 'all', 'the', 'audienc', 'wa', 'upset']	0.375	1.125	NEGATIVE

Finally, the analyzed data file is appended with the three additional attribute and values are positive score, negative score and the Sentiment label. The sentiment label can be either 'POSITIVE', 'NEGATIVE' or 'NEUTRAL' that define the sentiment prediction of the user tweet. A machine learning classifier was then trained based on the label indicating positive and negative sentiment, and classification performance is measured using the training set obtained from this natural language processing task.

Chapter 6

Machine Learning Techniques for Sentiment Analysis

6.1 Introduction

So far, the discussion on the machine learning concepts, which are more accurate for performing linguistic data analysis has been discussed as well as the WEKA platform for analyzing and training data using different machine learning algorithm is covered . The data output obtained by the proposed algorithm in Chapter 4 and 5 which filters data and perform linguistic data analysis using Natural Language Processing techniques (NLP). This data has been appended with the total positive score, negative score in the tweets and sentiment labeling ('POSITIVE', 'NEGATIVE' and 'NEUTRAL') has been assigned to each tweet in the dataset. These data sets which are labeled with sentiment of the tweets are further trained using machine learning algorithm to measure its accuracy, performance and reliability of the result obtained from lexicon based sentiment analysis. During the analysis, overall 9 attributes has been used from the data set, out of which mainly 3 attributes will be taken into account namely: PosScore (Positive score), NegScore

(Negative score) and sentiment labeling of the tweets evaluated, so far. The most abstract view that performs sentiment analysis using machine learning can be shown as in below Figure 6.



Figure 6: Sentiment Analysis using machine learning.

The sentiment labeled data with the total positive and total negative score for the words in the tweet has been computed and the training data set is prepared to perform sentiment analysis using machine learning algorithms like Naïve Bayes, SVM (Support Vector Machine) and Maximum Entropy. Here, to discuss the evaluation and performance of single data set applying machine learning algorithm using WEKA platform and the result obtained was interesting and satisfied results has been concluded.

6.1.1 Implementation using WEKA

Here, WEKA v3.6.11 is been used for analyzing and training data set using machine learning. **Waikato Environment for Knowledge Analysis (Weka)** is a collection of machine learning algorithms for data mining tasks written in Java and developed at

University of Waikato, New Zealand (Weka). It is free software licensed under the GNU General Public License (Weka). According to (Aksenova 2004) Weka can be used for real world data analysis and developing Machine Learning (ML) techniques that allow to access for training data in that environment. It also contains various tools for data pre-processing, classification, regression, clustering, association rules, and visualization.

Weka consists of various options which can be used to perform various operations on data. These options are as follow (Weka 3):

1. **Simple CLI** provides a simple command-line interface and allows direct executions of Weka commands.
2. **Explorer** is an environment for exploring data.
3. **Experimenter** is an environment for performing experiments and conduction statistical tests between learning schemes.
4. **Knowledge Flow** is Java-Beans-based interface for setting up and running machine learning experiment.

Therefore, by using the functionality of Weka to train the data set and analyze sentiment from data to measure its accuracy, building classifiers, clustering techniques, performing experiments and data visualization.

Here for analyzing data using Weka using test data for hashtag '#Brazil2014' dated from 'Sun June 8' to 'Mon June 9, 2014' (2 days) and time between '19:49:54' (7:45) to 23:59:58 (approx. 12:00), which is 5 hours and 15 minutes in total. The total number of tweets available to us from the data collector for this period were **24,335** tweets. This data has been collected during promotion period of the World Cup, the starting date of the World Cup was June 12, 2014. During the analysis, the result obtained is really interesting and the concepts for achieving this is explained in this literature.

Using the overall data set obtained from derived algorithm that appends the PosScore, NegScore and sentiment label attribute and therefore, as a result total nine attributes in the output data set. The experiment on sentiment analysis using WEKA for the data set is shown in below Figure 7.

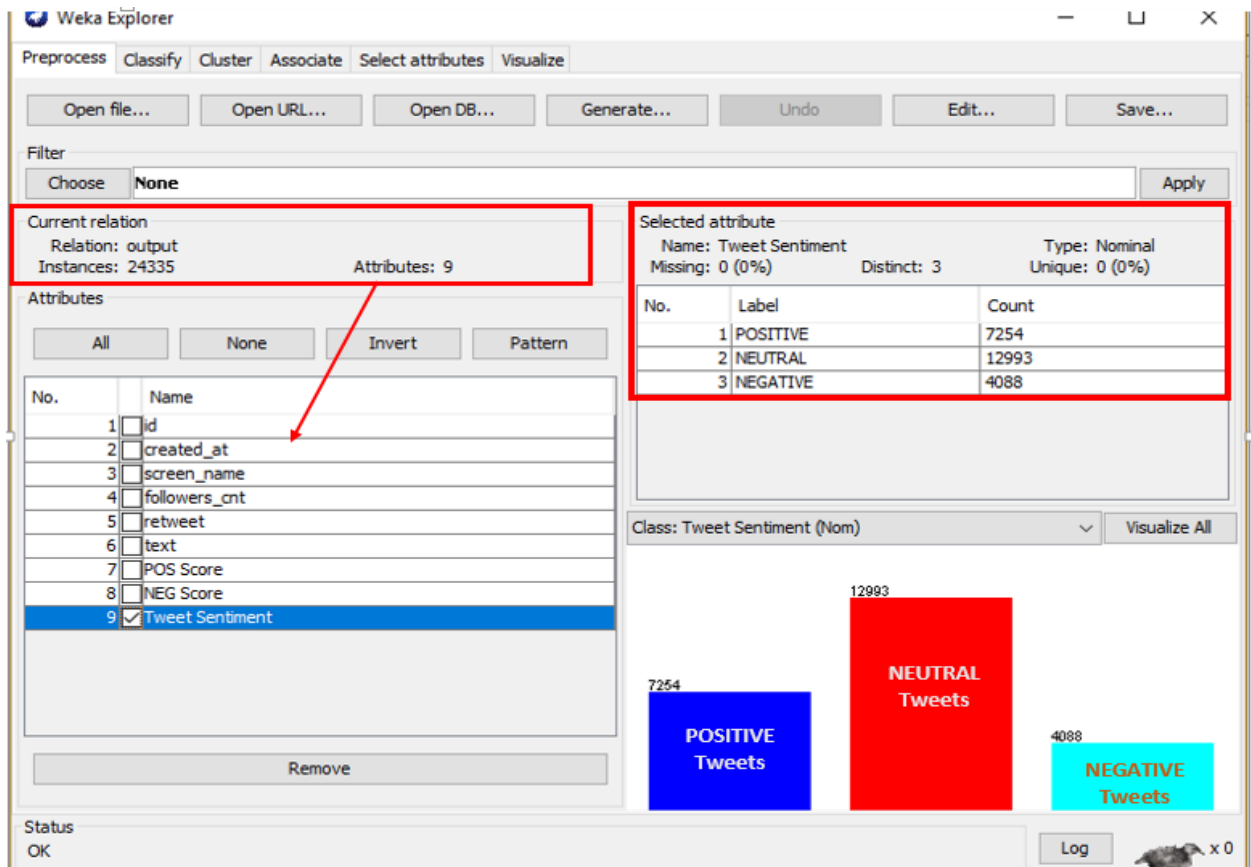


Figure 7: Sentiment Classification of Tweets

From above figure, 24335 tweets are classified into 7254 POSITIVE, 12993 NEUTRAL, and 4088 NEGATIVE tweets, which is 3 distinct classification of sentiment labels. The classification of NEUTRAL tweets is comparatively higher than positive and negative

tweets. The reason for higher score for “NEUTRAL” labeled tweets is because only English language tweets are analyzed and other languages during implementation of sentiment analysis algorithm has been ignored. However, the response toward the promotion of the World Cup 2014 was more positive in compare to negative. This response analysis for the sentiment of the people towards the event can give a strategic idea to the investors or the organizer to take a valuable decision for upcoming events based people’s feedback, likes and dislikes.

Further, the positive sentiment of the people in this case for World Cup 2014 promotion, now by comparing the overall all sentiment analysis with the total PosScore (Positive Score) attribute. The statistics and the results of analysis are shown in Figure 8.

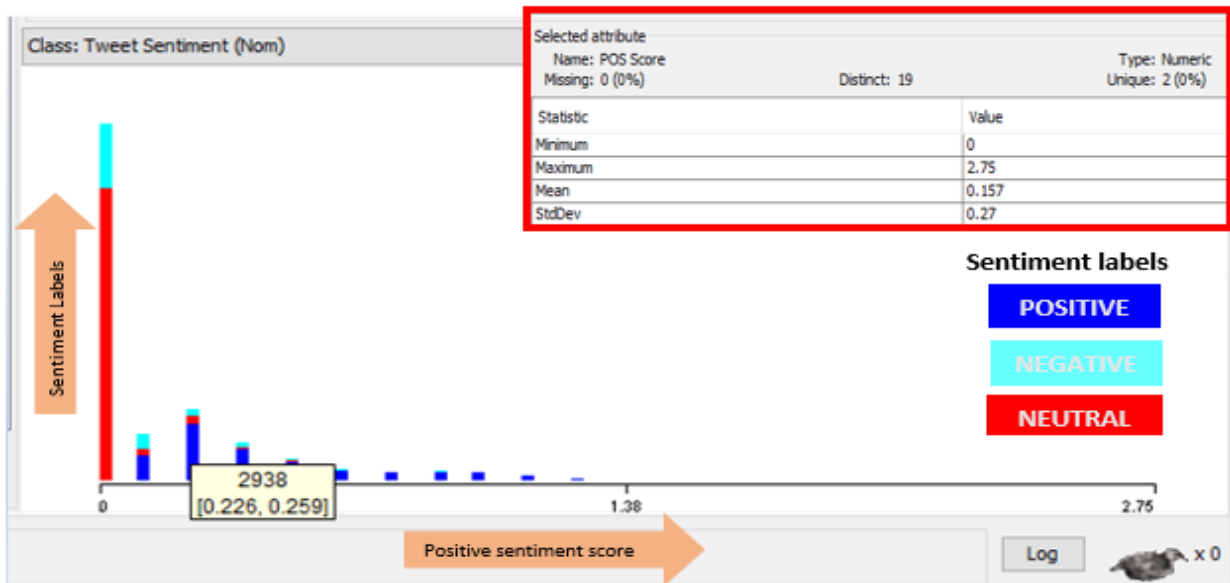


Figure 8: Accuracy of overall positive sentiment tweets

From Figure 8, summarize that most positive sentiment tweets has been scaled between 0 and 1.38. In the example above near second bar, there are 2938 Positive classified tweets

that fall under positive score from 0.226 to 0.259. Similarly, the maximum positive sentiment score for the tweet evaluated is 2.75, which is the most positive sentimental tweet in the data set of World cup 2014 promotion. The tweet which has been given a maximum positive score is “*Contact with nature is good for health. Come to the #EarthPortal and discover the wonders of the South of #Brazil! #Brazil2014*” tweeted by the username ‘portaldaterra’. Here, the presence of positive emotional words like ‘good’, ‘discover’, ‘wonders’ gives more contrast towards positivity of words with no single negative word in tweet, which conclude that tweet has positive sentiment. Moreover, the variation in positive sentiment score computed is 0.27 which is comparatively smaller, which shows that the classification of positive sentiment is consistency, predictability and quality in the resulted data set.

Moreover, the most negative sentimental tweet from the data set is obtained by comparing the overall sentiment score with the total negative sentiment score from the result data set. The result obtained can provide the organization with the worst feedback for not liking the event, which may allow them to improve using this single user feedback. The analysis of the most negative sentiment tweet can be shown in Figure 9.

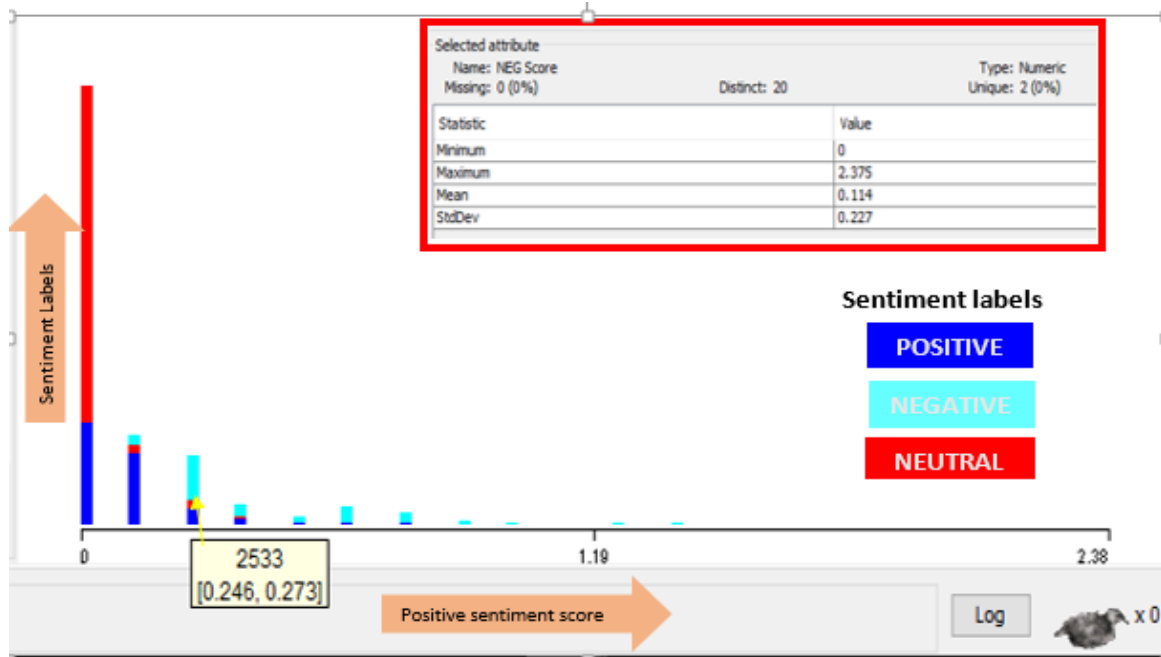


Figure 9: Accuracy of overall negative sentiment tweets

The most negative sentiment labeling can be found from 0.246 to 0.273 sentiment score, which is 2533 negative classified tweets. Also, the observation shows that the maximum negative sentiment score assigned is 2.375. This shows the most unhappy reaction blogged for the event, and there are two different tweets assigned with the same score are as:

- (1) Tweet# 4680 ‘3 daze to go. A few **problems: unfinished** stadiums, subway **strike** SP, Qatar bid **scandal**, **upset** sponsors. Ready or **not**. #Brazil2014 #WorldCup’,
- (2) Tweet# 14862 ‘It’s **hard** to hear but if #Rooney **isn’t** effective for #England he has to start on the bench. Too much faith/**pressure** on him #Brazil2014’H

Here, in above example the words in the tweet like ‘**problems**’, ‘**unfinished**’, ‘**strike**’, ‘**scandal**’, ‘**upset**’, ‘**not**’, ‘**hard**’, ‘**isn’t**’, ‘**pressure**’ classifies the tweets to maximum sentiment score and assign polarity to negative.

This shows that the derived classification algorithm using Natural Language Processing is valid and the result obtained is interesting. Now, to measure the accuracy for the resulted training data set machine learning algorithm comes into account which trains the data using various learning schemas and interpret received result. For training data set, the Naïve Bayes, and SVM (Support Vector Machine) algorithms has been applied to measure the accuracy and performance for sentiment labeled data. The detailed explanation about applying machine learning algorithm is discussed Section 6.2.

6.2 Analysis using Machine Learning

Machine learning can be defined as the process of inferring pattern and structure from the data by providing manually instruction to the machine to accomplished task. According to (Mohri *et al.* 2012), it is computational techniques that uses available information and predict accurate results using different algorithms. The information can be in the form of pre-processed data or electronic data collected and prepared for analysis. In this research, I have analyzed the Twitter linguistic data using Natural Language Processing (NLP) techniques as prepared data by assigning sentiment score and polarity to the data set. Now in this section the result of lexicon based method (derived in Chapter 5) and analyze training data set using machine learning classification algorithms. And lastly, by comparing results of machine learning classification algorithm and conclude the work. The most abstract view on applying machine learning techniques to the training Data set and analysis is shown in Figure 10.

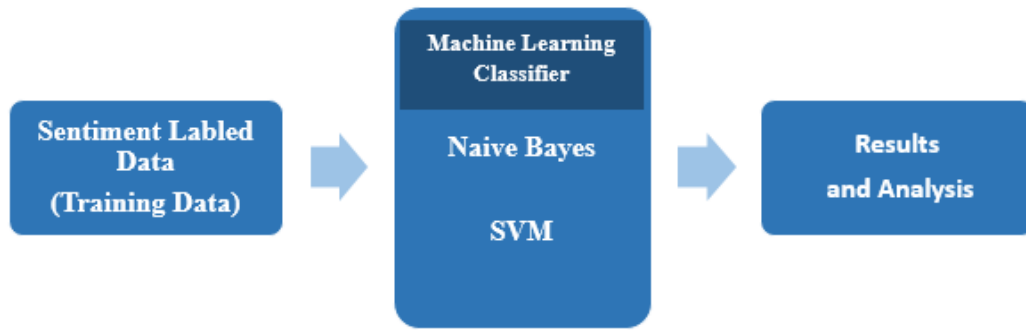


Figure 10: Overview on Applying Machine Learning

6.2.1 Naïve Bayes

Naïve Bayes is most commonly used classifier in Natural Language Processing (NLP). Pang *et al.* (2008) compared Naïve Bayes algorithm with other machine learning algorithm for sentiment analysis and achieved 90 per accuracy in classifying the data set. The main advantage of this classifier is its simplicity as well as prediction of the correct class for a new instance (Murphy 2006). It simply multiple all feature values which has been extracted from each instance in the class (for e.g. POSITIVE, NEGATIVE and NEUTRAL) are classified as a class. Every labeled sentiment tags (instance) contributes for the final classification result and given equally importance with respect to each other tokens in the data set. In machine learning, the sentiment labeled will be classified using this classifier and other attributes will be not considered anymore. Therefore, by considering only one attribute that is 'Tweet Sentiment' for classification using Naïve Bayes classifier and analyze result.

The Naïve Bayes classifiers assumes that every feature or the attribute of an instance (in this case ‘Tweet_Sentiment’ attribute) is considered independently from all other feature in the given class. And as a result, it will multiply all the members of feature vector in given class to compute Bayesian probability. Here for the available data set a given class is **Y** (POSITIVE, NEGATIVE and NEUTRAL), where **X** is the instance defined by a feature vector $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n\}$ with n being the number of features (sentiment labels) in the Data set. Therefore, Bayesian probability of the given class **Y** with an instance **X** can be computed $P(\mathbf{Y}|\mathbf{X})$ using following equation (4) (Murphy *et al.* 2006):

$$P(\mathbf{Y}|\mathbf{X}) = \frac{P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y})}{\sum_{\mathbf{Y}'=1}^c P(\mathbf{X}|\mathbf{Y}')P(\mathbf{Y}')} \quad (4)$$

- **X**: Is instance of class (sentiment labels for each tweet)
- **Y**: Is sentiment class (POSITIVE, NEGATIVE or NEUTRAL)
- $P(\mathbf{X}|\mathbf{Y})$: instance occurred in particular class for each value of **Y** (class-conditional density)
- $P(\mathbf{Y})$: prior probability of class

Using equation 4, we have $P(\mathbf{Y}|\mathbf{X})$ the Bayesian probability classification for the class **Y** to the instance **X**, which is equal to the probability $P(\mathbf{X}|\mathbf{Y})$ for the particular instance being seen under specific class. In this case, probability of each instance (sentiment labels) belong under specific class (POSITIVE, NEGATIVE or NEUTRAL). Further, it is multiplied with

the prior probability of the class $P(Y)$. At last, the result obtained is normalized so that the final probability for the given class with its instances will sum up to 1.

Weka is used for training the classifier for all instance of sentiment labeled tweets for each class (POSITIVE, NEGATIVE or NEUTRAL) to measure its accuracy, sensitivity, time cost and correctly classified instance for the training data set. The analyzed data of hashtag “#Worldcup” tweets is used to discuss the analysis throughout the literature to show the result of objective for sentiment analysis using machine learning. Figure 11 Shows the analysis using Naïve Bayes classifier and the result is been discussed.

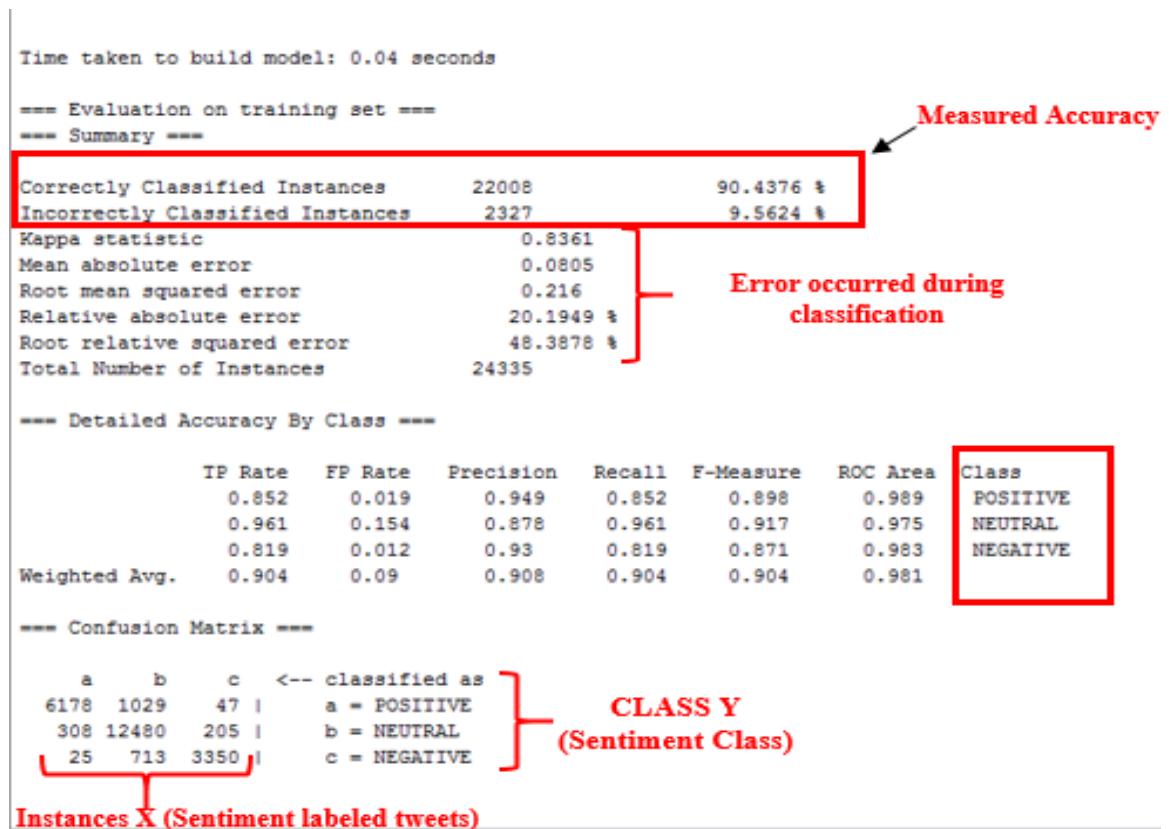


Figure 11: Result of analysis using Naïve Bayes classifier

The above analysis in Figure 11 gives us the estimation of predictive performance generated by WEKA’s evaluation module, one can observe accuracy 90.44 % (22,008

sentiment labeled tweets) is correctly classified with just 9.56 % (2327 sentiment labeled tweets) is incorrectly classified data. Therefore, the prediction of sentiment classes (POSITIVE,NEGATIVE,NEUTRAL) using Naïve Bayes classifier has accuracy rate of 90.44 %.The time taken for the classifier to train 24,335 instances just in 0.04 seconds, shows that the performance evaluation is very fast using Naïve Bayes classifier for the data set. The most important things to observe from the analysis is “ROC Area” column in Figure 11, the first row (i.e. 0.989) in Detailed Accuracy by class section and the “confusion matrix “section. Moreover, the accuracy of a classifier on a given data set is the percentage of the data set tuples that are correctly classified by the classifier. And the confusion matrix is a useful tool for analyzing how well your classifier can identify tuples of the different class. The confusion matrix generated by Naïve Bayes classifier is interpreted as well as the calculation for the detailed accuracy class is shown in table 26.

Table 26: Confusion matrix for sentiment class (Naïve Bayes)

	Predicted class				
Actual class		A (positive)	B (neutral)	C (negative)	Total
A (positive)		6178 (TP)	1029 (FN)	47 (FN)	7254
B (neutral)		308 (FP)	12480	205	12,993
C (negative)		25 (FP)	713	3350	4088
Total		6511	14,222	3602	24,335

Here we have 3x3 confusion matrix. The number of correctly classified instances is the sum of diagonal element in the confusion matrix; all other are incorrectly classified. For the computation purpose, let us assume TP_A be the number of true positive of the class A

(positive sentiment), TP_B be the true positive of the class B (Neutral sentiment) and TP_C be the number of true positives of class C.

- TP_A : refers to the positive tuples which are correctly labeled (POSITIVE) by the classifier in the first row- first column i.e. 6178.
- TP_B : refers to the positive tuples classified correctly labeled (NEUTRAL) by the classifier in second row – second column i.e. 12480.
- TP_C : refers to the positive tuples classified correctly labeled (NEGATIVE) by the classifier in third row- third column i.e. 3350.

Therefore, the Accuracy of the correctly classified instances can be calculated by the equation (5).

$$Accuracy_{Naive Bayes} = \frac{(TP_A + TP_B + TP_C)}{Total\ instance\ classified\ for\ class} \quad (5)$$

$$i.e\ Accuracy_{Naive Bayes} = \frac{(6178 + 12480 + 3350)}{24335} = 0.9043 \approx 90\%$$

Similarly, the total incorrectly classified instance are the instances except the highlighted in the confusion matrix in table 26. The total sum of that instances divided by the total number of classified instance give you incorrect classified instances is $0.0956 \approx 9.56\%$.

TPRate (True Positive rate), Sensitivity, and Recall: Number of sentiment labels predicted ‘positive’ that are actually ‘positive’ data set. Here, in Figure 11 one can see that the TPRate for POSITIVE, NEUTRAL, NEGATIVE are 0.852, 0.961, and 0.819 respectively. This observation shows that the data set classified is sensitive, which belongs

to the actual class. It is calculated for the all classes from the above confusion matrix using Equation (6):

- **Equation:** $TPRate, Sensitivity, Recall = \frac{\sum True\ Positive\ (TP)}{\sum Condition\ Positive}$ (6)

- **Example:** $TPRate_A = (6178) / (6178 + 1029 + 47) = 0.852$

$$TPRate_B = (12480) / (308 + 12480 + 205) = 0.961$$

$$TPRate_C = (3350) / (25 + 713 + 3350) = 0.819$$

Weighted Average for TPRate can be calculated by multiplying TPRate of each class with the TOTAL number of instances classified for that class and dividing by total number of instances.

$$\begin{aligned} \text{Weighted Avg} &= (0.852 * 7254) + (0.961 * 12993) + (0.819 * 4088) / 24335 \\ &= 22014.753 / 24335 = \mathbf{0.904} \end{aligned}$$

FPRate: False Positive: Number of sentiment labels predicted ‘positive’ that are actually ‘negative’ in the data set. Here the FPRate in Figure 11 for the POSITIVE, NEUTRAL, NEGATIVE are 0.019, 0.154, and 0.012 respectively. From the data one can conclude that the classification of the sentiment label has minimum number of tweet that are incorrectly classified. FPRate from the confusion matrix in table 26 can be calculated as in Equation (7).

$$FPRate = \frac{\sum False\ Positive\ (FP)}{\sum Condition\ Negative} \quad (7)$$

Precision: Precision is the proportion of the predicted positive cases that were correct. The precision can be calculated using the Equation (8)

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

Example of confusion matrix from Table 26:

$$\text{Precision}_A = 6178 / (6178 + 308 + 25) = 0.9488 \approx 0.949$$

$$\text{Precision}_B = 12480 / (12480 + 1029 + 713) = 0.8775 \approx 0.878$$

$$\text{Precision}_C = 3350 / (3350 + 47 + 205) = 0.9300 \approx 0.93$$

The weighted Average for Precision for the class POITIVE, NEUTRAL and NEGATIVE can be given as:

$$\begin{aligned} \text{Weighted Avg} &= (0.949 * 7254) + (0.878 * 12993) + (0.93 * 4088) / 24335 \\ &= 22093.74 / 24335 = 0.9078 \approx \mathbf{0.908} \end{aligned}$$

The result of precision shows that the data set has correctly classified the positive cases for the instances in the data set. Therefore, one can say that 94 % of POSITIVE, 87% of NEUTRAL and 93 % Negative labeled data set is correctly or positively classified using Naïve Bayes classifier.

F-Measure: The F-measure score is the harmonic mean of the precision and recall. This evaluates the equivalency between the sensitivity (recall) and the precision (correctness) of the data. This give us the interpretation about how the measure recall and precision

values behaves for the data set. The F-measure can be calculated from the confusion matrix in Table 26 using Equation (9).

$$F - measure = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (9)$$

The F-Score measure for the class POSITIVE (A), NEUTRAL (B) and NEGATIVE (C) can be calculated using Equation 9 as:

$$F\text{-measure}_A = 2 * (0.949 * 0.852) / (0.949 + 0.852) = 0.898.$$

Similarly, $F\text{-measure}_B = 0.917$, $F\text{-measure}_C = 0.871$ and the weighted Average for the F-Score for the class A, B and C can be given as:

$$\begin{aligned} \text{Weighted Avg} &= (0.898 * 7254) + (0.917 * 12993) + (0.871 * 4088) / 24335 \\ &= 21989.321 / 24335 = 0.9036 \approx \mathbf{0.904} \end{aligned}$$

ROC Area: The Receiver Operating Characteristics (ROC) graphs is techniques for organizing classifier and visualizing the performance of the trained data using algorithm. The ROC for the POSITIVE, NEUTRAL and NEGATIVE class is 0.989, 0.975 and 0.983. From which one can say that the performance evaluation for the POSITIVE class in better in compare to other classed in the data set. Therefore, one can say the overall sentiment score for the given data set is positive based on the highest ROC Area computation. Also, the weighted value for ROC area is 0.981 (98 %) of the data classified using Naïve Bayes is correctly classified with higher accuracy and performance.

The error occurred during classification can be interpreted using parameters namely: Kappa statistic, Mean absolute error, root mean squared error, relative absolute error, and root relative squared error.

- **Kappa statistic:** “Kappa Statistic” is analog of correlation coefficient. It derives statistical relation between the class label and attribute of instances. It is 0 if there is lack of relation and approaches. Here in figure 11, value of Kappa statistic is 0.83 means that the statistical significance of the Naïve Bayes model is rather high statistical dependence.
- **Mean absolute and root mean squared error:** Both this errors simply look for the “average difference” of true value and estimated value obtained using algorithm. Root means squared error is the root of mean absolute error. Here the mean absolute error is 0.08 and root mean squared error is 0.2. Which is comparatively negligible to the result obtained.

The overall weighted average or the accuracy for the sentiment labeled classes as well as the result from analysis using Naïve Bayes can be shown in Table 27 and the detail discussion on computing this parameter is discussed.

Table 27: Accuracy of Sentiment Labeled dataset using Naïve Bayes

Number	Parameters	Naïve Bayes
1	TPRate	0.904
2	FPRate	0.09
3	Precision	0.908
4	Recall	0.904
5	F-Measure	0.904
6	ROC Area	0.981

6.2.2 SVM (Support Vector Machine)

It is a supervised learning method in which produces a mapping function from the available training data set (Wang *et al.* 2005). Support Vector Machine (SVMs) is widely applied for classification problem and nonlinear regression, which classifies both linear and nonlinear (Wang *et al.* 2005). The mapping function can be the classification function which classify the labeled data in the data set. According to (Joachims *et al.* 1998), SVMs are universal learners, which can be used to learn polynomial classifiers and it has ability to learn independent of the dimensionality of the feature space. SVM is very useful in dealing with questions related to classification of texts by linearly separating them as suggested by (Joachims *et al.* 1998). One of the disadvantages of using SVM is that it is incapable of differentiating between words that have different senses in different sentences and so, particular “domain-based lexicons” cannot be generated (Joachims *et al.* 1998). While, in this approach the generated sentiment for the Twitter data using lexicon based

approach and the machine learning techniques has been applied to measure the accuracy by combining both the approaches.

Here, in the section SVM algorithm is applied for classifying sentiment labels from Twitter data and measure accuracy of the classified data using WEKA platform. In which, the labeled tweets from the data set trained using SVM classifier and the classification obtained shows the accuracy of the data set. It is observed that performing classification using SVM algorithm consumes a huge amount of computer memory, using the computer with 8GB RAM for processing 24,335 tweets was not possible with SVM classifier. In this case, the Naïve Bayes algorithm classification in comparison to SVM consumes less internal computer memory. Here, comparatively more accuracy and lower performance for data classification is achieved using SVM algorithm. There result for the analysis of the data set using SVM algorithm is in fig 12.

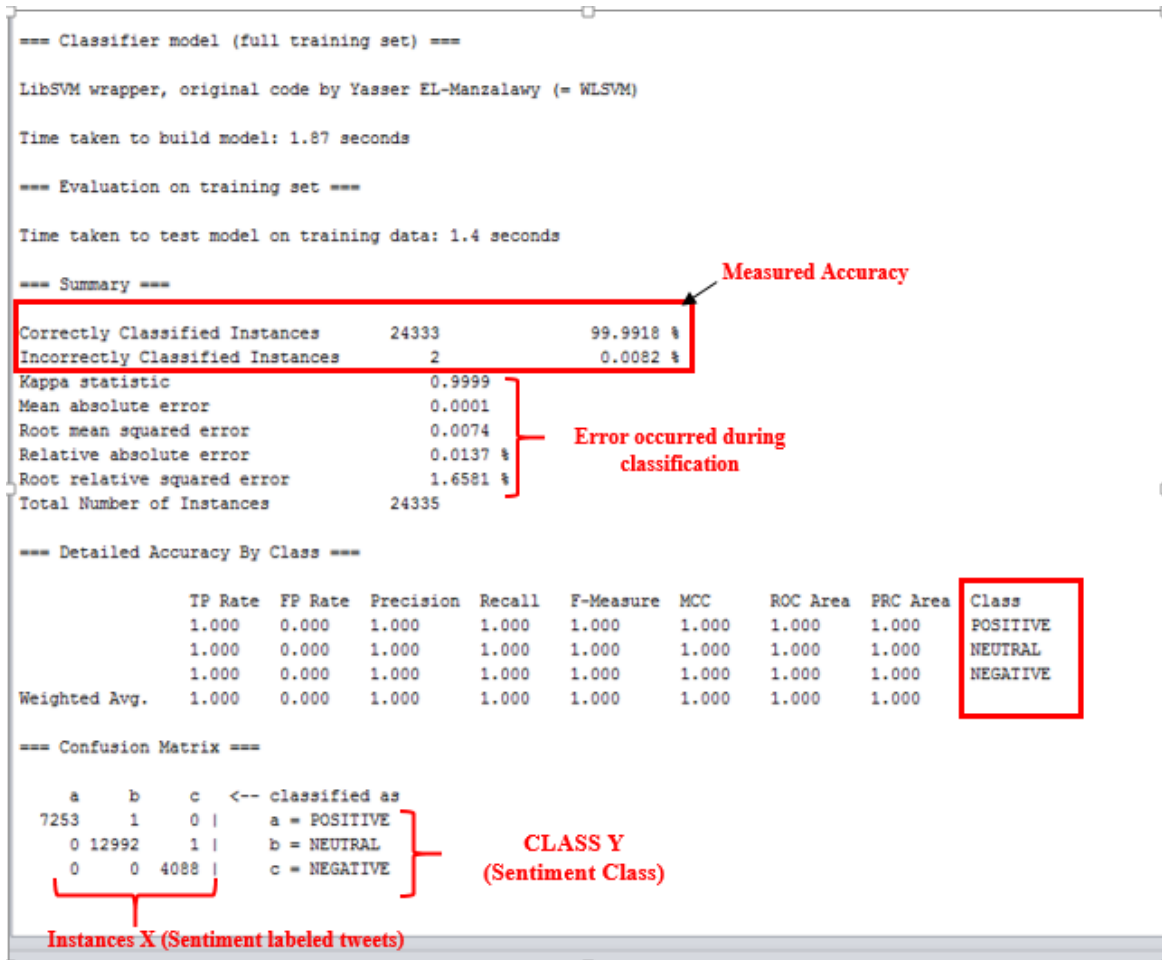


Figure 12: Result of analysis using SVM Algorithm

From the above analysis in Figure 12 gives us the estimation of predictive performance generated by WEKA's evaluation module for SVM classifier, one can observe accuracy 99.99 % (24,333 sentiment labeled tweets) is correctly classified with just 0.0082 (2 sentiment labeled tweets) is incorrectly classified data. Therefore, the prediction of sentiment classes (POSITIVE, NEGATIVE, And NEUTRAL) using SVM classifier has accuracy rate of 99.99 %. The time took for the classifier to train 24,335 instances just in 1.87 seconds, shows that the performance evaluation is slower in compare to Naïve Bayes. The most important things to observe from the analysis is "ROC Area" column in the first

row (i.e. 1) in Detailed Accuracy by class in Figure 12. The confusion matrix generated by Naïve Bayes classifier is interpreted as well as the calculation for the detailed accuracy class is shown in table 28.

Table 28: Confusion matrix for sentiment class (SVM)

	Predicted class				
Actual class		A (positive)	B (neutral)	C (negative)	Total
A (positive)		7253 (TP)	1 (FN)	0 (FN)	7254
B (neutral)		0 (FP)	12992	1	12,993
C (negative)		0 (FP)	0	4088	4088
Total		6511	14,222	3602	24,335

From the above confusion matrix one can compute Accuracy, TPRate, FPRate, Precision, F-Measure and ROC Area can be calculated using the Equation (5), (6), (7), (8) and (9). And the obtained for all of the parameters can be shown in Table 29. In which, all the instances are classified correctly and the performance evaluation is 100 % for the sentiment labeled data set.

Table 29: Accuracy of Sentiment Labeled dataset using SVM

Number	Parameters	SVM
1	TPRate	1
2	FPRate	0.0
3	Precision	1
4	Recall	1
5	F-Measure	1
6	ROC Area	1

6.3 Results and Comparison.

Two different machine learning techniques has been used for training sentiment labeled data set and the result obtained using both the classifier are accurate. The performance evaluation, accuracy, sensitivity and classification result obtained using Naïve Bayes and SVM supports the objective for processing linguistic data set using Natural language processing (NLP) techniques and measure the accuracy of the sentiment labeled data set. In comparison of using Naïve Bayes and SVM for measuring the accuracy for sentiment labeled data set, SVM algorithm stand ahead given high accuracy in data classification. Although, Naïve Bayes gives better performance and throughput for data classification. For training SVM in using 8 GB RAM machine does not allow to train the data, as Naïve Bayes train the data set with the huge file size with greater speed. The statistics of the performance is show in table 30.

Table 30: Comparison of Naïve Bayes and SVM

Number	Parameters	Naïve Bayes	SVM
1	TPRate	0.904	1
2	FPRate	0.09	0
3	Precision	0.908	1
4	Recall	0.904	1
5	F-Measure	0.904	1
6	ROC Area	0.981	1
7	Performance Time	0.04 sec	1.87 sec

It is clearly seen from the result in the table 30, that the SVM algorithm stands ahead in comparison to Naïve Bayes classification algorithm except for the time taken to train the data and the memory consumption during data classification.

Below Figure 13.

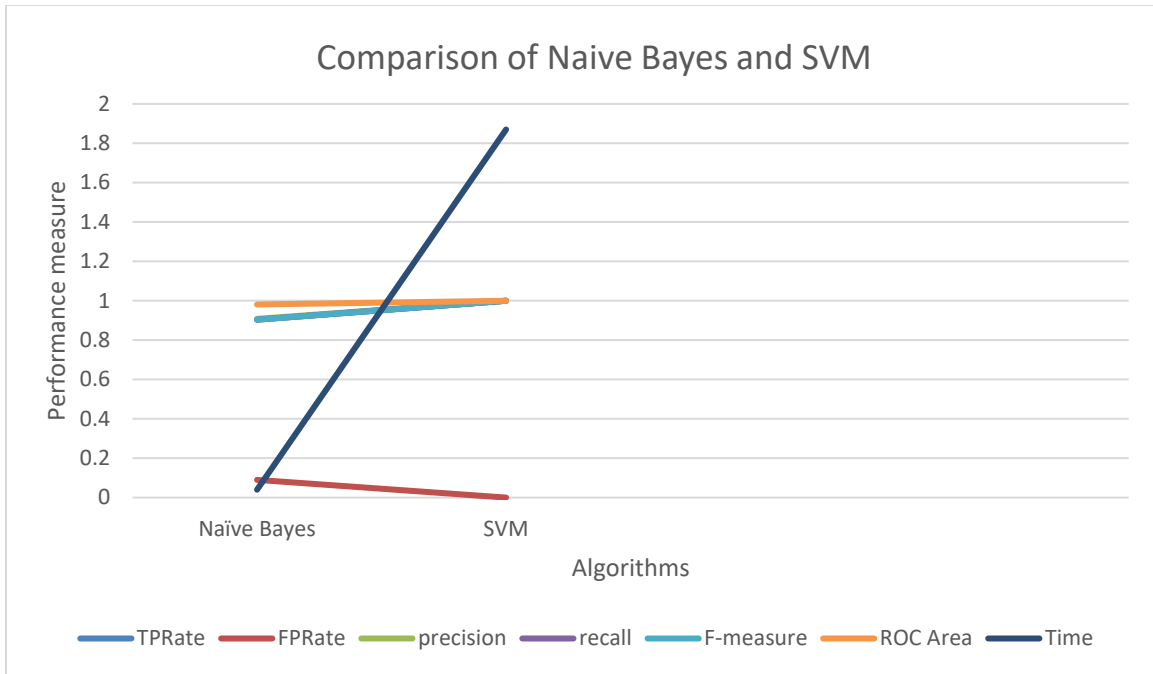


Figure 13: comparison of Naïve Bayes and SVM performance

The above Figure 13, shows that the time for training the same data set using SVM takes too long in compare to Naïve Bayes. Also, the value for ROC area is 0.981 (98 %) for Naïve Bayes and 1 for SVM which is negligible measure for correctly classified instances for sentiment labeled tweets and has higher accuracy and performance. Therefore, one can conclude from this that both the classifier used for analyzing sentiment data set stands ahead. Objective for combining lexicon based and machine learning method for sentiment analysis for Twitter data proves that using Natural language processing (NLP) techniques gives accurate classification for linguistic data set and machine learning techniques classifies the instances for measuring accuracy.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, a unique approach to perform sentiment analysis on linguistic data set has been introduced using machine learning algorithm. The developed algorithms for removing noise or data filtering and pre-processing linguistic data using Natural Language Processing (NLP) techniques is demonstrated. Also, In order to pre-process or filter the noise from the textual Twitter data, it is necessary to perform sequence of pre-processing steps. During this process the input tweets are filtered and processed to give more accurate data as well as reduce the size of dataset. In these steps of pre-processing by renaming the links 'URL' and in the final steps removed the word 'URL' from the tweets to gain accurate data. Likewise, the same method was applied in the renaming and removing usernames from the tweets. Furthermore, by filtering #Hashtags, characters that are repeated more

than two times in the word, any special characters (for e.g.: \ | [] ; { } - + () < > ? ! @ # % *) from the tweets. Hence, using derived algorithm the satisfied result is achieved that reduces the size of the dataset thereby, filtering unnecessary noise from the tweets and prepared tweets in the order perform further processing tasks.

To perform core natural language processing for the tweeter data, by analyzing the data using NLTK toolkit that has different function that allow us the process natural language. It is initialized with the word tokenization method that allow us to tokenize each words in the tweet, which allowed to perform unigram analysis of the word. Then in the next step to perform stemming and lemmatizing of the word, the words obtained in this step are the base form of word which contains the root meaning for the given term. Later, by assigning part-of-speech (POS) tags to all the term in the sentence context and obtained the synsets term by analyzing the WordNet in combination with the negation marks assigned to each word in the sentence. The negation word mark is evaluated by allocating '1' to negative term and '0' to all positive occurrence of word followed by negation word in the sentence. Moreover, in derived approach based on the negation marked for each word by changing the meaning of the word to the antonyms of word if the negation mark is '1' and left with synonyms synset if the word occurs in negation mark '0'. This is the key step of analyzing sentiment polarity and to obtain accurate sentiment score for the given synset when applied to the lexical resources. Further, computed positive and negative sentiment score for the each word in the tweet using SentiWordNet lexical resources that assigns sentiment score to each term. Finally, by aggregating the total positive and total negative sentiment score for all the occurrence of the word in the tweet and compared them to label the overall sentiment score for the given tweet. Since, the analyses assigned

sentiment label to each tweet that analysis the sentiment of the user when reacting on the Twitter platform, which derives not only the opinion from the user but allow the business to know the feedback about the event, game, promotion. Further, analysis of data using machine learning concepts like Naïve Bayes, SVM and Maximum Entropy algorithm for measuring consistency, accuracy and reliability of classified sentiment analysis data. The visualization of the sentiment analysis result using WEKA platform and compared the result using machine learning algorithm.

7.2 Future Work

For the future work on sentiment analysis it is necessary to perform real time sentiment polarity assigning to the Twitter data. To do so, by preparing an outline to implement same data processing algorithm on cloud that increase the performance for sentiment analysis using Natural Language Processing (NLP) techniques. This can be done by creating nodes on cloud data platform like Hadoop that allow us to store the data on cloud using HDFS (Hadoop File System) and Map-reduce concept to distribute the data processing algorithm on cloud to load and process large size data set and real time sentiment analysis for the linguistic data. This will be contribution towards real time sentiment analysis in a cloud environment and will allow the business user to fetch real time sentiment analysis for the linguistic data.

References

- Aksenova, Svetlana S. "WEKA Explorer Tutorial.", 2004.
- Augustyniak, Łukasz, et al. "Comprehensive Study on Lexicon-based Ensemble Classification Sentiment Analysis." *Entropy* 18.1 (2015): 4.
- Bandgar, B. M., and Binod Kumar. "Real time extraction and processing of social tweets." *International Journal of Computer Science and Engineering, E-ISSN 2347-2693* (2015): 1-6.
- Bhattacharyya, Pushpak, et al. "Facilitating multi-lingual sense annotation: Human mediated lemmatizer." *Global WordNet Conference*. 2014.
- Bird, Steven. "NLTK: the natural language toolkit." *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006.
- Bird, Steven, Ewan Klein, and Edward Loper. *Natural language processing with Python*. "O'Reilly Media, Inc.", 2009.
- Cambria, Erik. "Affective computing and sentiment analysis." *IEEE Intelligent Systems* 31.2 (2016): 102-107.
- Carrillo de Albornoz, Jorge, Laura Plaza, and Pablo Gervás. "A hybrid approach to emotional sentence polarity and intensity classification." *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2010.
- Carvalho, Jonnathan, Adriana Prado, and Alexandre Plastino. "A Statistical and Evolutionary Approach to Sentiment Analysis." *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 02*. IEEE Computer Society, 2014.
- Chowdhury, Gobinda G. "Natural language processing." *Annual review of information science and technology* 37.1 (2003): 51-89.
- "Company | About." *Twitter*. Twitter, 30 June 2016. Web. 04 Mar. 2017.

- Esuli, Andrea, and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." *Proceedings of LREC*. Vol. 6. 2006.
- Fernández-Gavilanes, Milagros, et al. "Unsupervised method for sentiment analysis in online texts." *Expert Systems with Applications* 58 (2016): 57-75.
- Firmino Alves, André Luiz, et al. "A Comparison of SVM versus naive-bayes techniques for sentiment analysis in tweets: a case study with the 2013 FIFA confederations cup." *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. ACM, 2014.
- Gastelum, Zoe N., and Kevin M. Whattam. "State-of-the-Art of Social Media Analytics Research." *Pacific Northwest National Laboratory* (2013).
- Gonçalo Oliveira, Hugo, António Paulo Santos, and Paulo Gomes. "Assigning Polarity Automatically to the Synsets of a Wordnet-like Resource." *OASICS-OpenAccess Series in Informatics*. Vol. 38. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.
- González, Cristóbal Barba, et al. "A Fine Grain Sentiment Analysis with Semantics in Tweets." *International Journal of Interactive Multimedia and Artificial Intelligence* 3.Special Issue on Big Data and AI (2016).
- Hemalatha, I., Dr GP Saradhi Varma, and A. Govardhan. "Case Study on Online Reviews Sentiment Analysis Using Machine Learning Algorithms." *International Journal of Innovative Research in Computer and Communication Engineering* 2.2 (2014):3182-3188.
- Hemalatha, I., Dr GP Saradhi Varma, and A. Govardhan. "Preprocessing the informal text for efficient sentiment analysis." *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 1.2 (2012): 58-61.
- Hemalatha, I., Dr GP Saradhi Varma, and A. Govardhan. "Sentiment analysis tool using machine learning algorithms." *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)* 2.2 (2013): 105-109.
- Hull, David A. "Stemming algorithms: A case study for detailed evaluation." *JASIS* 47.1 (1996): 70-84.
- Isah, Haruna, Paul Trundle, and Daniel Neagu. "Social media analysis for product safety using text mining and sentiment analysis." *Computational Intelligence (UKCI), 2014 14th UK Workshop on*. IEEE, 2014.

Joachims, Thorsten. "Text categorization with support vector machines: Learning with many relevant features." *European conference on machine learning*. Springer Berlin Heidelberg, 1998.

Kaufmann, Max, and Jugal Kalita. "Syntactic normalization of Twitter messages." *International conference on natural language processing, Kharagpur, India*. 2010.

Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore. "Twitter sentiment analysis: The good the bad and the omg!." *Icwsn* 11 (2011): 538-541.

Kurian, Merin K., et al. "Big Data Sentiment Analysis using Hadoop." *International Journal for Innovative Research in Science and Technology* 1.11 (2015): 92-96.

Kuchling, A. M. "Regular Expression HOWTO." *Regular Expression HOWTO—Python* 2.10 (2014).

Kumar, KM Anil, et al. "Analysis of users' Sentiments from Kannada Web Documents." *Procedia Computer Science* 54 (2015): 247-256.

Lam, Khang Nhut, Feras Al Tarouti, and Jugal Kalita. "Automatically constructing Wordnet Synsets." *ACL* (2). 2014.

Liu, Haibin, et al. "BioLemmatizer: a lemmatization tool for morphological processing of biomedical text." *Journal of biomedical semantics* 3.1 (2012): 1.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Murphy, Kevin P. "Naive bayes classifiers." *University of British Columbia* (2006).

Neri, Federico, et al. "Sentiment analysis on social media." *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. IEEE Computer Society, 2012.

Nithish, R., et al. "An Ontology based Sentiment Analysis for mobile products using tweets." *2013 Fifth International Conference on Advanced Computing (ICoAC)*. IEEE, 2013.

Olsson, Fredrik. "A literature survey of active machine learning in the context of natural language processing." (2009).

- Ohana, Bruno, and Brendan Tierney. "Sentiment classification of reviews using SentiWordNet." *9th. IT & T Conference*. 2009.
- Pang, Bo, and Lillian Lee. "Opinion mining and sentiment analysis." *Foundations and trends in information retrieval* 2.1-2 (2008): 1-135.
- Perkins, Jacob. *Python text processing with NLTK 2.0 cookbook*. Packt Publishing Ltd, 2010.
- Porter, Martin F. "An algorithm for suffix stripping." *Program* 14.3 (1980): 130-137.
- Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of Twitter." *International Semantic Web Conference*. Springer Berlin Heidelberg, 2012.
- Selvan, Lokmanyathilak Govindan Sankar, and Teng-Sheng Moh. "A Framework for Fast-Feedback Opinion Mining on Twitter Data Streams." *Collaboration Technologies and Systems (CTS), 2015 International Conference on*. IEEE, 2015.
- Taboada, Maite, et al. "Lexicon-based methods for sentiment analysis." *Computational linguistics* 37.2 (2011): 267-307.
- Tan, Luke Kien-Weng, et al. "Sentence-level sentiment polarity classification using a linguistic approach." *International Conference on Asian Digital Libraries*. Springer Berlin Heidelberg, 2011.
- Thusoo, Ashish, et al. "Hive-a petabyte scale data warehouse using hadoop." *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*. IEEE, 2010.
- Toman, Michal, Roman Tesar, and Karel Jezek. "Influence of word normalization on text classification." *Proceedings of InSciT 4* (2006): 354-358.
- Van Rossum, Guido. "Python Programming Language." *USENIX Annual Technical Conference*. Vol. 41. 2007.
- Wawer, Aleksander. "Is Sentiment a Property of Synsets? Evaluating Resources for Sentiment Classification using Machine Learning." *LREC*. 2010.
- Wang, Lipo, ed. *Support vector machines: theory and applications*. Vol. 177. Springer Science & Business Media, 2005.
- "Weka 3: Data Mining Software in Java." *Weka 3 - Data Mining with Open Source Machine Learning Software in Java*. N.p., n.d. Web. 03 Jan. 2017.
<<http://www.cs.waikato.ac.nz/ml/weka/>>.

Younis, Eman MG. "Sentiment Analysis and Text Mining for Social Media Microblogs using Open Source Tools: An Empirical Study." *International Journal of Computer Applications* 112.5 (2015).

MacArthur, Amanda. "The Real History of Twitter, In Brief - How the micro-messaging wars were won." *lifewire*, 3 Oct. 2016, <https://www.lifewire.com/history-of-Twitter-3288854>. Accessed 1 December 2016.

"It's what happening. Is" *Twitter*, 30 Jun. 2016, <https://about.Twitter.com/company>. Accessed 1 December 2016.

Appendix A

CC Coordinating conjunction	PRP\$ Possessive pronoun
CD Cardinal number	RB Adverb
DT Determiner	RBR Adverb, comparative
EX Existential there	RBS Adverb, superlative
FW Foreign word	RP Particle
IN Preposition or subordinating conjunction	SYM Symbol
JJ Adjective	TO To
JJR Adjective, comparative	UH Interjection
JJS Adjective, superlative	VB Verb, base form
LS List item marker	VBD Verb, past tense
MD Modal	VBG Verb, gerund or present participle
NN Noun, singular or mass	VBN Verb, past participle
NNS Noun, plural	VBP Verb, non-3rd person singular present
NNP Proper noun, singular	VBZ Verb, 3rd person singular present
NNPS Proper noun, plural	WDT Wh-determiner
PDT Predeterminer	WP Wh-pronoun
POS Possessive ending	WP\$ Possessive wh-pronoun
PRP Personal pronoun	WRB Wh-adverb