FAST DATA ANALYSIS METHODS FOR SOCIAL MEDIA DATA

by

Valentine Velaphi Nhlabano

Submitted in fulfilment of the requirements for the degree
Master of Science (Computer Science)

in the

Department of Computer Science
Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

November 2019

Abstract

The advent of Web 2.0 technologies which supports the creation and publishing of various social media content in a collaborative and participatory way by all users in the form of user generated content and social networks has led to the creation of vast amounts of structured, semi-structured and unstructured data. The sudden rise of social media has led to their wide adoption by organisations of various sizes worldwide in order to take advantage of this new way of communication and engaging with their stakeholders in ways that was unimaginable before. Data generated from social media is highly unstructured, which makes it challenging for most organisations which are normally used for handling and analysing structured data from business transactions. The research reported in this dissertation was carried out to investigate fast and efficient methods available for retrieving, storing and analysing unstructured data form social media in order to make crucial and informed business decisions on time. Sentiment analysis was conducted on Twitter data called tweets. Twitter, which is one of the most widely adopted social network service provides an API (Application Programming Interface), for researchers and software developers to connect and collect public data sets of Twitter data from the Twitter database.

A Twitter application was created and used to collect streams of real-time public data via a Twitter source provided by Apache Flume and efficiently storing this data in Hadoop File System (HDFS). Apache Flume is a distributed, reliable, and available system which is used to efficiently collect, aggregate and move large amounts of log data from many different sources to a centralized data store such as HDFS. Apache Hadoop is an open source software library that runs on low-cost commodity hardware and has the ability to store, manage and analyse large amounts of both structured and unstructured data quickly, reliably, and flexibly at low-cost. A Lexicon based sentiment analysis approach was taken and the AFINN-111 lexicon was used for scoring. The Twitter data was analysed from the HDFS using a Java MapReduce implementation. MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster. The results demonstrate that it is fast, efficient and economical to use this approach to analyse unstructured data from social media in real time.

Keywords

Sentiment analysis, Text Mining, Social Media, Social Networks, Machine Learning, Unstructured Data, Naïve Bayes, Big Data, Data Analysis, Apache Hadoop, MapReduce, Hadoop Flume

Supervisor

Supervisor: Dr P.E.N. Lutu

Department: Department of Computer Science

Table of Contents

CHAPTER 1: INTRODUCTION 1
1.1 Problem Statement 4
1.2 Focus of this research 5
1.3 Research Questions 6
1.4 Research Paradigm 7
1.5 Research contribution 7
1.6 Organisation of the report 9
CHAPTER 2: SOCIAL MEDIA CONCEPTS 10
2.1 Web 2.0 as an enabler for Social media and Social networks 10
2.2 Social Media 11
2.3 Social Networks 13
2.4 Microblogs 15
2.5 Twitter a Microblogging Social Network 17
2.6 Big Data 19
2.7 Data Mining and CRISP-DM 22
2.8 Text Mining 25
2.9 Summary 26
CHAPTER 3: ANALYSING SOCIAL MEDIA DATA 27
3.1 Social Media Analytics 27
3.2 Content-based analysis 28
3.2.1 Text Mining Approaches 28
3.2.2 Classification 29
3.2.3 Sentiment Analysis 31
3.2.4 Machine Learning Approaches 34
3.2.5 Sentiment Analysis Levels 37
3.2.6 Lexicon-based Approaches 39
3.3 Summary 41
CHAPTER 4: TECHNOLOGIES FOR BIG DATA RETRIEVAL AND PROCESSING 43
4.1 Characteristics of Social Media Data 44
4.2 Apache Hadoop 45
4.3 Hadoop Distributed File System (HDFS) 47
4.4 Apache MapReduce 48
4.5 Apache Flume 51
4.6 Summary 52
CHAPTER 5: RESEARCH METHODS 53
5.1 Research Paradigm 53

5.2 Data for Text Pre-processing Experiments 56
5.3 Twitter Data Collection 57
5.4 Streaming Twitter Data into Hadoop Using Apache Flume 58
5.5 Conclusion 60
CHAPTER 6: EXPERIMENTS FOR TEXT PRE-PROCESSING 61
6.1 Objectives of the Text Pre-processing Experiments 61
6.2 Study of Text Pre-processing methods on the Performance of Sentiment Analysis Models for Social Media Data 63
6.3 Data set and algorithms for the experiments 64
6.4 Text Pre-processing Methods 64
6.4.1 Stop word removal 65
6.4.2 Removal of URLs and @Username 65
6.4.3 Stemming using the Porter Stemmer 65
6.4.4 Feature Selection 67
6.5 Experimental Results for Text Pre-processing 70
6.6 Discussion of Experimental Results for Text Pre-processing 73
6.7 Conclusion 73
CHAPTER 7: EXPERIMENTS FOR SENTIMENT ANALYSIS 74
7.1 Objectives of the Experiment 74
7.2 Experiment setup 75
7.3 Sentiment Analysis with AFINN Lexicon 76
7.4 Tokenization 78
7.5 Sentiment Analysis using the MapReduce Algorithm 78
7.6 Experimental results for sentiment analysis 81
7.7 Discussion of the experimental results for sentiment analysis 88
7.8 Classifier Performance Evaluation 89
7.8.1 Accuracy 92
7.8.2 Precision 93
7.8.3 Recall 93
7.8.4 F-measure 94
7.9 Conclusion 95
CHAPTER 8: DISCUSSION 96
CHAPTER 9: CONCLUSION AND FUTURE WORK 98
9.1 Conclusion 98
9.2 Future Work 98
Appendix A: Installing Java and Hadoop on Windows 7 112
Appendix B: Installing Apache Flume 117

Appendix C: The Twitter application 121
Appendix D: MapReduce Algorithm 122
Appendix E: Pre-processing Implementation 125
Appendix F: A tweet 126
Appendix G: Snowball Porter Stemmer 127
Appendix H: Publications and Conference Presentations 130

List of Figures

Figure 2.1: Social networks ranked by number of active users (Statista, 2019).....16
Figure 2.2: Number of active Twitter users worldwide (Statista, 2019) 18
Figure 2.3: Phases of the CRISP-DM reference model (Chapman et al., 2000).....24
Figure 3.1: A traditional framework for text analytics (Hu & Liu, 2012) 28
Figure 3.2: An example of text mining (Fan et al., 2006) 29
Figure 3.3: Supervised classification (Bird et al., 2009) 30
Figure 3.4: Sentiment analysis process on product reviews (Medhat et al., 2014) 32
Figure 3.5: Sentiment classification approaches.....34
Figure 3.6: Naïve Bayes Algorithm 37
Figure 3.7: Twitter Data Sentiment Calculation (Yadav & Elchuri, 2013).....40
Figure 4.1: The Master/Worker Architecture of Hadoop (Stoneman, 2016).....46
Figure 4.2: HDFS Architecture (Borthakur, 2013).....48
Figure 4.3: MapReduce Algorithm 50
Figure 4.4: The Word Count Process 50
Figure 4.5: MapReduce Word Count Example 51
Figure 4.6: Apache Flume data model.....52
Figure 5.1: Design Science research process based on 6 steps as proposed by Pfeffers et al. (2006).....54
Figure 5.2: Log or Event data generators 58
Figure 5.3: Fetching Twitter data using Flume.....59
Figure 6.1: Flow diagram for the text pre-processing experiments 62
Figure 6.2: Example of the stemming process.....66
Figure 6.3: Categories of stemming algorithms 66
Figure 7.1: Overall System Architecture 76
Figure 7.2: Sentiment analysis algorithm.....79
Figure 7.3: Sentiment Analysis Process Workflow.....80
Figure 7.4: Scoring a tweet using AFINN.....82
Figure 7.5: MapReduce Counters.....83
Figure 7.6: Sentiment Classification Process Algorithm.....86
Figure 7.7: A Pie Chart Showing Sentiment Analysis Results 87
Figure 7.8: A Bar Graph Showing Sentiment Analysis Results.....87
Figure 7.9: A Plot Graph Showing Sentiment Analysis Results 88

List of Tables

Table 2.1: Types of social media (Barbier & Liu, 2011; Hu & Liu, 2012).....13
Table 5.1: Evaluation criteria for outcomes of Design Science (March & Smith, 1995) 56
Table 6.1: Text Pre-processing Experiment Results.....71
Table 6.2: T-test Results.....72
Table 7.1: Sample of entries in the AFINN-111 file.....77
Table 7.2: Data File System (DFS) Storage Types.....81
Table 7.3: Outputfile.csv 84
Table 7.4: Data Table 84
Table 7.5: Sentiment classification results.....86
Table 7.6: Estimated Impact of Scaling Hadoop Clusters.....88
Table 7.7: Tweet processing rates.....89
Table 7.8: Binary Classifier Confusion Matrix 90
Table 7.9: Twitter Data Confusion Matrix.....91

Table 7.10: Recall, Precision and F-Score Calculation 94
Table A.1: Java and Hadoop installation steps 112
Table B.2: Apache Flume installation steps 117

CHAPTER 1: INTRODUCTION

The proliferation of affordable Internet-enabled devices and the advent of Web 2.0 technologies have led to the rapid generation of social media data in the form of opinionated User Generated Content (UGC) and social networks. Advances in automated data processing, machine learning and natural language processing (NLP) presents a potential possibility to utilise this massive data source for a variety of purposes including processing it into useful business information for decision making by organisations (Sarker et al., 2015; Paul et al., 2016). The challenge is that this can only be achieved if researchers are able to address the unique methodological challenges that this massive data sources presents (Paul et al., 2016).

Machine learning is a technique for recognizing patterns from examples; It contains a set of methods which enable machines to learn meaningful patterns from data directly with minimal human interaction (Fu et al., 2019; Erickson et al., 2017). This provides the capability of extracting meaningful patterns from large data sets such as social media data and enables the acceleration of the pace of automation itself (Brynjolfsson & Mitchell, 2017). Natural language processing (NLP) provides techniques that support the conversion of text data into a structured representation that computers can process and thus enables them to process and derive meaning from human (i.e. natural language) input (Pons et al., 2016).

The lack of advances in this area of study is possibly due to the fact that before World Wide Web and social networks there was not much opinionated text (written text that contains opinions) readily available online or it could also be due to the fact that social media platforms have seen unprecedented worldwide growth (Paul et al., 2016). According to Liu (2012) for the first time in human history, a huge volume of opinionated data recorded in digital form is now available online for analysis. The studies carried out in this dissertation addresses the challenges that are presented by social media data and introduces fast and efficient machine learning and natural language processing techniques and methods for automatically processing and classifying social media data.

The term Web 2.0 describes a platform that supports collaborative and participatory way of creating and publishing web content by users (Cohn, 2018). Mesquita et al. (2016) have defined social networking as a social structure with people who are joined by a common

interest. This presented new unprecedented opportunities and challenges to both producers and consumers of the information from the perspective of knowledge discovery through data analysis (Aggarwal, 2011). Cohn (2018) defined social media as the use of web based and mobile technologies to turn communication into an interactive dialogue.

The relationship between Web 2.0, social media and UGC is that Web 2.0 is the technology and ideology that enables UGC, and UGC is the sum of all the possible things that people can make use of social media (Kaplan & Haenlein, 2010). Social networking is a sub-category of social media in the sense that it is a social structure of people who share common interests that exists within social media (Cohn, 2018). People create their profiles on social media channels and they interact with each other based the personal details that they read about each other. Since social networking is a sub-category of social media, some or most of the data analysis techniques used to analyse social media data are also used for the analysis social networks data.

In the business sector, traditionally computers were used to process and analyse data generated from business transactions into information which facilitated managers in making informed decisions. The last two decade has witnessed a vast amount of potentially useful data generated by users on social networking sites such as Facebook, Twitter, Instagram, YouTube, Flickr, and LinkedIn just to mention a few. Data generated from social networking sites if analysed effectively and timeously can empower organisations to make informed business decisions that can increase customer and employee loyalty, and take full advantage of the value of their efforts in servicing customers, marketing, employee relations, sales and product development. Understanding what customers think about their products and/or services empowers business organisations to act quickly and compete more effectively (IBM, 2013). The content of this data is usually made up of limited short sentence fragments or links to videos, images and websites. Perhaps the major difference is in the frequency of update of this data for example a user may post several updates in a single day (Java et al., 2007).

The challenge is that this data is generated every second, from different sources and comes in different formats which make it differ in many ways from data generated from business transactions. This data require special processing methods because in many cases the knowledge extraction process has to be efficient and close to real-time because storing all

generated data is nearly infeasible (Wu et al., 2014). These are characteristics of big data which presents an extreme challenge for discovering useful knowledge from this data. Big data is data that is characterized with large volume, heterogeneous, autonomous sources having distributed and decentralized control, and the challenge is to discover complex and evolving relationships among this enormous data (Wu et al., 2014). Katal et al. (2013) maintains that the most common use of big data is for the social media and customer sentiments i.e. keeping an eye on what the customers of the business organisation are saying about their products and services helps business organisations to get a kind of useful customer feedback which is then used to change and modify decisions and get more value out of their business.

Sentiment analysis also known as opinion mining is the study of people's opinions, attitudes, appraisals and emotions regarding events, individuals, issues, entities, topics and their attributes (Liu, 2015; Liu, 2012). Textual data can be categorized into two broad categories i.e. facts and opinions. Liu (2010) defined facts as objective expressions about entities, events and their properties, and opinions as subjective expressions that described people's sentiments, appraisals or feelings toward entities, events and their properties. Sentiment analysis has received a lot of attention in recent years due to its wide variety of practical applications, its promising commercial benefits and also due to the many interesting challenges and research problems it presents to the research field (Angiani et al., 2016). Sentiment analysis offers organisations the ability to automatically monitor public opinions towards their products and services and also events related to them in real time (Jianqiang & Xiaolin, 2017). For example, customers are interested in other customers opinions about a product before making a purchase, whereas business organisation are interested in finding out about their competitors or their suppliers and feedback from their customers pertaining to their products and services offerings. Investors are concerned about financial news related to their investments.

Opinions expressed in social networks play a major role in influencing public opinion's behaviour across areas as diverse as buying products, capturing the ''pulse'' of stock markets and voting for the president (Bai, 2011; Eirinaki, Pisal, & Singh, 2012). In Dietrich et al. (2015), the authors claim that 80-90% of future world's data growth is expected to come from document databases or unstructured text databases. Given this claim, sentiment analysis systems became critical in allowing organisations to make sense of this abundance

of unstructured text data, by automating this process in order to save hours of manual data processing. This dissertation presents an automatic Twitter data sentiment analyser, which downloads opinionated Twitter data for a given topic, utilise natural language processing techniques to pre-process this data and analyse it for user sentiments. This gives the organisation the capability of processing and monitoring their social media activities in real time for decision making.

1.1 Problem Statement

The sudden rise of social media and the adoption of social networks have led to the rapid generation of big data which presents opportunities for organisations if harnessed properly to construct valuable information for enhanced decision making process (Sivarajah et al., 2017). However, big data also presents unprecedented challenges to harnessing such large increasing volumes of data and has a complex nature that requires powerful technologies and advanced algorithms which traditional data processing tools can no longer be efficient (Oussous et al., 2018).

Business organisations and government institutions are utilizing social networking sites such as Facebook and Twitter to communicate and interact with their customers, as a result generating vast amounts of user generated content on a daily basis (He et al., 2015). These organisations should aim to process and analyse this data as quickly as possible in order to better service their customers and understand their competitors in order to achieve a competitive advantage. Analysing this data has huge benefits to the organisation such as determining the unfiltered and honest sentiment the public has on their products and/or services in an economic way without the need for explicit surveys. Opinionated social media data which is full of unbiased user sentiments is readily available online. Organisations can process this data for information instead of collecting it via other means such as surveys and questionnaires which are sometimes time consuming, expensive and can be biased.

However, the problem is that traditional data management systems methods are not capable of handling and processing this type of data efficiently and also most organisations lack the capability, the knowledge and understanding of the various available methods they can implement in order to analyse this data and convert it into useful business information for the benefit of their organisations (Oussous et al., 2018). Currently managers of

organisations are challenged to process and analyse vast amounts of social media data, but lack a framework within which to do so (Lee, 2018).

1.2 Focus of this research

This study investigates various social media data analysis methods, techniques and technologies that organisations can implement to analyse data from social media efficiently. Given the breadth of these methods a comprehensive list of the methods is outside the scope of this study. The study however addresses a number of data analysis methods for social media data and additionally provides an implementation of content based text analysis of social media data using Apache Hadoop an open source technology for distributed storage and distributed processing of very large data sets on computer clusters.

The reasons for narrowing the analysis to text data is in three folds. The first one is that most social networking sites are rich in unstructured text data and text analytics techniques can help efficiently handle data in text format in social networks for business and research purposes (Hu & Liu, 2012). The second reason is that the majority of useful sentiments are contained in unstructured text data on social networking sites. Lastly according to Liu (2010), much of the existing research on textual information has been focused on mining and retrieval of factual information such as information retrieval, Web search, text clustering, text classification and many other text mining and natural language processing tasks and little work has been done on the processing of opinions until recently, yet opinions are so important for both organisations and individuals, that whenever there is a need to make a decision, there is always a need to consult other's opinions before making that decision .

In analysing social networking data there are two sources of information namely User Generated Content (e.g. images, bookmarks, videos and texts) and the relationships and interactions between the network entities (e.g. customers, organisations, competitors, suppliers). Content-based analytics was performed on User Generated Content and Structure-based analytics also known as Social network analytics which is performed when analysing the relationships and interactions between the network entities. This research focuses on Content-based analytics in which the focus is on text data using Apache Hadoop which is a possible solution to efficiently storing and analysing big data generated by social

networking sites. Data from Twitter which is both a social networking site and a microblogging site was used for the empirical research in this study.

1.3 Research Questions

In order to achieve the research objectives, the following research questions were formulated. They include a main research question and the related sub-questions.

The main research question is: What technology and methods can be utilized to process social media text data fast and efficiently?

The main research question was broken down into the following sub-questions:

1. What is Web 2.0?
2. What is social media?
3. What is a social networking site?
4. What is social networking data?
5. What is big data?
6. Who is interested in social media data and why?
7. How can opinionated text data found on social networking sites be analysed to provide information for an organisation?
8. What analysis methods are available to automatically process data from social networking sites?
9. What type of information does social media data analysis provide?
10. How can social media data be stored efficiently?
11. How can data that is generated in large volumes, in a variety of different formats that arrives at high velocity be handled in real time as it is being generated?
12. How can speed and efficiency be achieved when working with big data?
13. What innovative methods and technological artefacts can be created to process social media text data fast and efficiently?
14. Can big data technology be employed to process social media data?
15. What natural language text processing techniques can be applied to social media data processing?
16. What impact does text pre-processing have on the performance of sentiment analysis models for social media data and machine learning?

1.4 Research Paradigm

The design science research paradigm was chosen for this research because it involves the creation and evaluation of artefacts. March & Smith (1995) described design science research as consisting of two basic activities i.e. build and evaluate. The process of constructing an artefact for a specific purpose is called building and the process of determining how well the artefact performs is the evaluation activity (Brady et al., 2013). According to Lukka (2003) design science is a research method for creating innovative constructions which are intended to solve problems faced in the real world and to make a contribution to the theory of the discipline in which it is applied. In this research the aim is to build and evaluate fast and efficient methods which solve problems which organisations face when dealing with big data challenges.

1.5 Research contribution

Analysis of social media data has several benefits for most business organisations and government institutions. These benefits may include development of new products and services, faster and better decision making, and in some cases cost reduction. Monitoring and analysing social media data enables organisations to get a great idea of what consumers are saying about their products and/or service as well as about their competitors. Analysis of social media data also provides an opportunity for an organisation to make informed decisions, and to enable the assessment of social networking campaigns and to adjust such campaigns in order to optimize and justify spending.

This research contributed by demonstrating that implementing and using a set of text pre-processing methods can lead to an improvement in accuracy and efficiency of sentiment analysis models. This is mainly due to the fact that online social media data contains a lot of noise and uninformative parts which have a negative impact on the processing of social media text data into information.

The second contribution of this study was implementing a fast and efficient, automatic sentiment analysis system which collects large streams of real-time public data; store this data in a central location and processes it in a parallel and distributed manor using low-cost

commodity hardware and open source software. According to Liu (2012) sentiment analysis systems are being applied in almost every business and social domain due to the fact that opinions are central to almost all human activities and they are influencers of human behaviour. Whenever people want to make decisions they tend to seek opinions of other people and this is not only true for individuals, but also for organisations.

The contributions of this research have significant beneficial effect on each of the functional areas of business organisations that mainly provide services or manufacture products. Social networks in general and micro-blogs (such as Twitter) in particular provide the potential to easily obtain customer feedback. Storage and analysis of social networking data creates a platform that supports faster and better decision making, development of new products and services and in some cases cost reduction. Monitoring and analysing social networking data enables business organisations to get a great idea of what consumers are saying about their products and/or service as well as about their competitors. It also gives the organisation a chance to assess the influence of their marketing strategies and activities as well as those of their competitors. Analysis of social networking data enables quick response in Social Customer Relationship Management (CRM) and engages customers in different scenarios, such as, ideas for Innovation, development of new products and services, word-of-mouth marketing, price comparisons and product reviews.

A customer relationship management platform that integrates social networking sites gives the business access to the same level of insight as traditional channels, plus the ability to use social tools for communicating internally. The business can monitor, track and benchmark social networking communications using familiar tools, reports, dashboards and metrics. This gives the organisation both a broad overview of its brand's reach and a much more granular, detailed view of each customer interaction. This places the customer right at the heart of the organisation.

The creation of social networking sites has emerged with surprisingly new possibilities for marketers to sell their products and services. Stelzner (2011) has compiled a much more comprehensive list of how a business can harness the power of social networking and the marketing strategies that organisation can employ to take advantage of social networks in a variety of ways as well as to promote brand awareness, interact and communicate with customers. Generated exposure for business, increased traffic or subscribers, improved

search ratings, resulted in new business partnerships, generated qualified leads, reduced overall marketing expense and improved sales as some of the benefits of social networks marketing were some of the benefits identified by Stelzner (2011).

1.6 Organisation of the report

Chapter 2 provides a literature review in order to define and provide background on some of the most important terms used in this study. Chapter 3 continues with the literature review from Chapter 2 and also provides an in-depth focus on the data analysis methods which are discussed in literature for text data. Chapter 4 provides the last part of the literature review in which the author explores the available fast and efficient methods for the analysis of social media data. Chapter 5 describes the research design and methodology that was used in conducting this research and also presents the data gathering process which was carried out to generate data for experiments conducted in this study. Chapter 6 provides an in-depth discussion of the two main experiments conducted in this research i.e. experiments for text pre-processing and sentiment analysis experiments. The results of the experiments from Chapter 6 are discussed in more details in Chapter 7. Chapter 8 concludes this research and also outlines implications and future work which was identified and may require further research to be fully addressed.

CHAPTER 2: SOCIAL MEDIA CONCEPTS

The aim of this chapter is to discuss the literature for technologies which are related to the study of social media. The discussion provides answers to the following research sub-questions: (1) What is Web 2.0? (2) What is social media? (3) What is big data? (4) What is a social networking site? (5) What kind of data is found on a social networking site?

This chapter is organized as follows: Section 2.1 to Section 2.8 provides a comprehensive literature review of Web 2.0, social media, social networks and big data. This is primarily to put the terms in perspective and provide direct meaning in the context that they are used in the study. Section 2.9 provides a summarised discussion of this chapter.

2.1 Web 2.0 as an enabler for Social media and Social networks

The term Web 2.0 is widely attributed to Musser & O'reilly (2007), but it was first quoted by Darcy in DiNucci (1999). According to Musser & O'reilly (2007) the term “Web 2.0” appeared in 2004, when the first official conference on Web 2.0 took place in a brainstorming session between O'Reilly and MediaLive International . Web 2.0 supports user-generated content, usability (ease of use, even by non-experts), and websites that can work well with other products, systems and devices. The term may seem to suggest a new version of the World Wide Web, but it does not refer to any actual change in technical specifications of the technology itself, rather it refers to changes in the ways software developers and end-users utilize the Web. On the other hand Web 1.0 represented a one-to-many online platform where a few business, organisations and individuals held a one-way dialog with people over the Internet. They could pass on information in a variety of ways, but the interaction was limited.

Web 2.0 is a term that describes a new way in which software developers and end-users started to utilize the World Wide Web as a platform whereby content is no longer created and published by specific individuals, but instead are continuously modified by all Internet users in a participatory and collaborative fashion (Kaplan & Haenlein, 2010). The core of Web 2.0 tools is the capability they provide for users to interact and contribute to generating and providing additional content. In other words Web 2.0 consists of content-publishing platforms that led to the growth of social media sites that resulted in the generation of a wide

variety of huge amounts of data. This resulted in an urgent need for analysing such data and converting it into useful knowledge to leverage business and other organisation decisions.

Communication with traditional media such as radio, television, newspapers and Web 1.0 websites was predominantly one-way, in which producers of information would publish information to the masses of media consumers. The introduction of Web 2.0 and contemporary social media has transformed this way of generating and disseminating information. Web 2.0 is a move from one-way communication from producers of media to a one in which everyone can generate and publish both text, videos, and or audio content in collaborative and interactive many-to-many social media dialogue in a virtual community (Barbier & Liu, 2011; Hu & Liu, 2012). A major advantage of this form of communication is that it provides an opportunity for an organisation to reach their customers and stakeholders in ways that were not possible before in both scale and extent. It also allows an organisation to communicate with extremely large numbers of people at an extremely low cost. Perhaps the greatest benefit is that the resulting data provides a rich source of new insights to consumer behaviour and marketing to a business organisation (Barbier & Liu, 2011).

Web 2.0 was seen as the next step for the Web and represents a many-to-many content creation in which individuals can set up their own websites and blogs, post videos, and fill the Web with user-generated content. Web 2.0 are a technology shifting the Web to turn it into a participatory platform, in which people not only consume content (via downloading) but also contribute and produce new content (via uploading). Web 2.0 is a collection of interactive and user-controlled online applications expanding the experiences, knowledge and market power of the users as participants in business and social processes. Minazzi (2015) described Web 2.0 services as those that support the creation of informal users' networks facilitating the flow of ideas and knowledge by allowing the efficient generation, dissemination, sharing and editing/refining of informational content.

2.2 Social Media

The terms social media and Web 2.0 are often used as interchangeable, but some observers relate the term Web 2.0 mostly with online services and the term Social media with the social aspects of Web 2.0 applications which include participation, openness, conversation,

community, and connectedness. Due to the diverse number of definitions of social media, Obar & Wildman (2015) have provided a synthesised definition of social media based on the following common properties among current social media services :

(1) Social media services are Web 2.0 Internet-based applications
(2) User-generated content is the essence of social media
(3) Users and groups create user-specific profiles for a site or application designed and maintained by a social media service
(4) Social media services enable the development of social networks online by enabling linking a profile with those of other individuals and/or groups.

Hobbs (2014) has defined social media as websites and applications or 'apps' (computer software programs) that enable individuals and/or organisations to interact, create, share and exchange information. Both Obar & Wildman (2015) and Hobbs (2014) have stated that social media refers to Internet based applications and websites.

Social media sites generate large quantities of valuable data through user profiles, ranging from their favourite books to movies, and such information can be targeted for very specific advertising. According to Liu (2012), 'acquiring public and consumer opinions has long been a huge business itself for marketing, public relations, and political campaign companies'. In general businesses and organisations always want to find consumer or public opinions about their products and services whereas individual consumers also want to know the opinions of existing users of a product or service before purchasing it. In the political environment, others want to find out opinions about political candidates before making a voting decision in a political election (Liu, 2012).

Table 2.1 shows a synthesized list of the common types of social media as discussed in Barbier & Liu (2011); Hu & Liu (2012). Best & Thompson (2018) and Vergeer & Hermans (2013) maintain that Twitter is both a microblogging and social networking service. In this study the author agrees with this argument so Twitter was added in both categories as shown in Table 2.1.

Table 2.1: Types of social media (Barbier & Liu, 2011; Hu & Liu, 2012)

Category	Examples
WIKIS	Wikipedia, Scholarpedia, Wikihow, Event maps
BLOGGING	Blogger, LiveJournal, WordPress
SOCIAL NEWS	Digg, Mixx, Slashdot
MICRO BLOGGING	Twitter, Google Buzz
OPINION & REVIEWS	ePinions, Yelp
QUESTION ANSWERING	Yahoo! Answers, Baidu Zhidao
MEDIA SHARING	Flickr, YouTube
SOCIAL BOOKMARKING	Delicious, CiteULike, StumbleUpon
SOCIAL NETWORKING	Facebook, Twitter, LinkedIn, MySpace, Orkut

Younis (2015) maintains that sentiment analysis can be applied on any form of textual messages such as blogs, product reviews, microblogs and most of the social media listed in Table 2.1. Extensive academic research on opinion mining and sentiment analysis has been conducted on social media data such as Twitter data (Agarwal et al., 2011; Liu, 2012; Pak & Paroubek, 2010; Pang & Lee, 2008; Vinodhini & Chandrasekaran, 2012; Wakade et al., 2012; Zhang & Liu, 2016). This dissertation presents efficient techniques for processing data generated by users on social networking sites which benefit both business organisations and government institutions were investigated.

2.3 Social Networks

Siegel (2006) has defined social networks as the actual virtual communities that are generally found over the web where members of common purposes or interest share an unlimited and unrestricted amount of information. Social networks provide a venue for socialization and business where friends, co-workers, and business contacts create a virtual community where they can interact. Social networks have become increasingly popular over the last ten years amongst Internet users especially the young generation, offering a unique way to publish opinions and information instantly. The last decade has seen a huge growth in the use of Social networking sites such as Facebook, Twitter, LinkedIn and YouTube.

Online social networks have become increasingly popular in the last decade, providing and enabling an efficient, user-friendly and possibly addictive way to connect and share information to maintain social connections and share information online (Benson et al.,

2010). Social networks have been proven to facilitate business relationships and building of social capital using electronic media. Social networks mainly differ from a traditional blog in that their content is usually made up of limited short sentence fragments or links to videos, images and websites. The word blog is short for “weblog” and refers to an online journal which displays the author’s posts in reverse chronological order. Another major difference is in the frequency of update. Java et al. (2007) have noted that on average, a prolific blogger may update his/her blog once every few days. On the other hand a microblogger may post several updates in a single day.

The reasons for businesses and organisations participation in social networking include assessing the opinions of the public about products and services that the organisations provide, soliciting opinions from the public, and communication and collaboration between stakeholders (Lutu, 2015). Lutu (2015) also stated that both big and small business organisations are routinely using social media for marketing and branding purposes and also argued that, for African countries, the major benefits of using social media are the low costs of engaging with the citizens and business customers as well as the ability to analyses the effectiveness of such engagements.

Utilising social network services such as Facebook, Twitter LinkedIn and Google+ via the Internet and the Web 2.0 technologies has become more affordable than communicating the traditional way such as sending mails via posts or making telephone calls. People are becoming more interested in and depending on social network for information, news and opinion of other users on different subject matters. Borgatti (2009) has defined a social network in mathematical terminology. He defined social networks as a graph in Mathematics, which is a collection of nodes (also referred to as vertices or actors); together with a set of ties (also known as edges or links) that link pairs of nodes. Borgatti (2009) maintains that a social network graph is typically used to represent social relations such as who are friends with whom, or who is the supervisor of whom. Simply put, a social network can be represented as a mathematical graph consisting of nodes and links used to represent social relations on social network sites.

The statistics in Figure 2.1 provide information on the most popular social networks worldwide as of January 2019, ranked by the number of active users (measured in millions). It shows that Facebook followed by YouTube and WhatsApp are the most popular social

networking sites as of January 2019 based on the number of active users (Statista, 2019). According to Statista (2019) approximately two billion Internet users are using social networks and the number of users are expected to rise due to increased mobile device usage and mobile social networks increasingly being adopted.

Availability in multiple language and a strong user engagement are some of the reasons cited by Statista (2019) that leads to the popularity and wide adoption of a social network. Social networks have a decidedly strong social impact leading to the blurring between offline and virtual life as well as the concept of digital identity and online social interactions. For the research reported in this dissertation, data from the social network Twitter (which is ranked at number 12 in Figure 2.1) was collected and analysed for sentiment analysis. Twitter and other social networks such as Tumblr are mainly about rapid communication and are aptly termed microblogs. Microblogs are discussed in more details in the next section.

2.4 Microblogs

Java et al. (2007) have described microblogging as a new form of communication in which users can describe their current opinions in short posts distributed by instant messages, mobile phones, email or the Web. Microblogs have ushered the world into a new era of social media and for business organisations this presents a marketing opportunity that surpasses the traditional middleman and connects business organisations directly with customers and has also given researchers access to massive quantities of data for empirical analysis. This growth has reshaped business and influenced companies and media organisations to increasingly seek ways to store and analyse this data for information about what people think and feel about their products and services. In their study Java et al. (2007) defined Twitter as a popular microblogging tool which has seen a lot of growth since it launched in October, 2006.

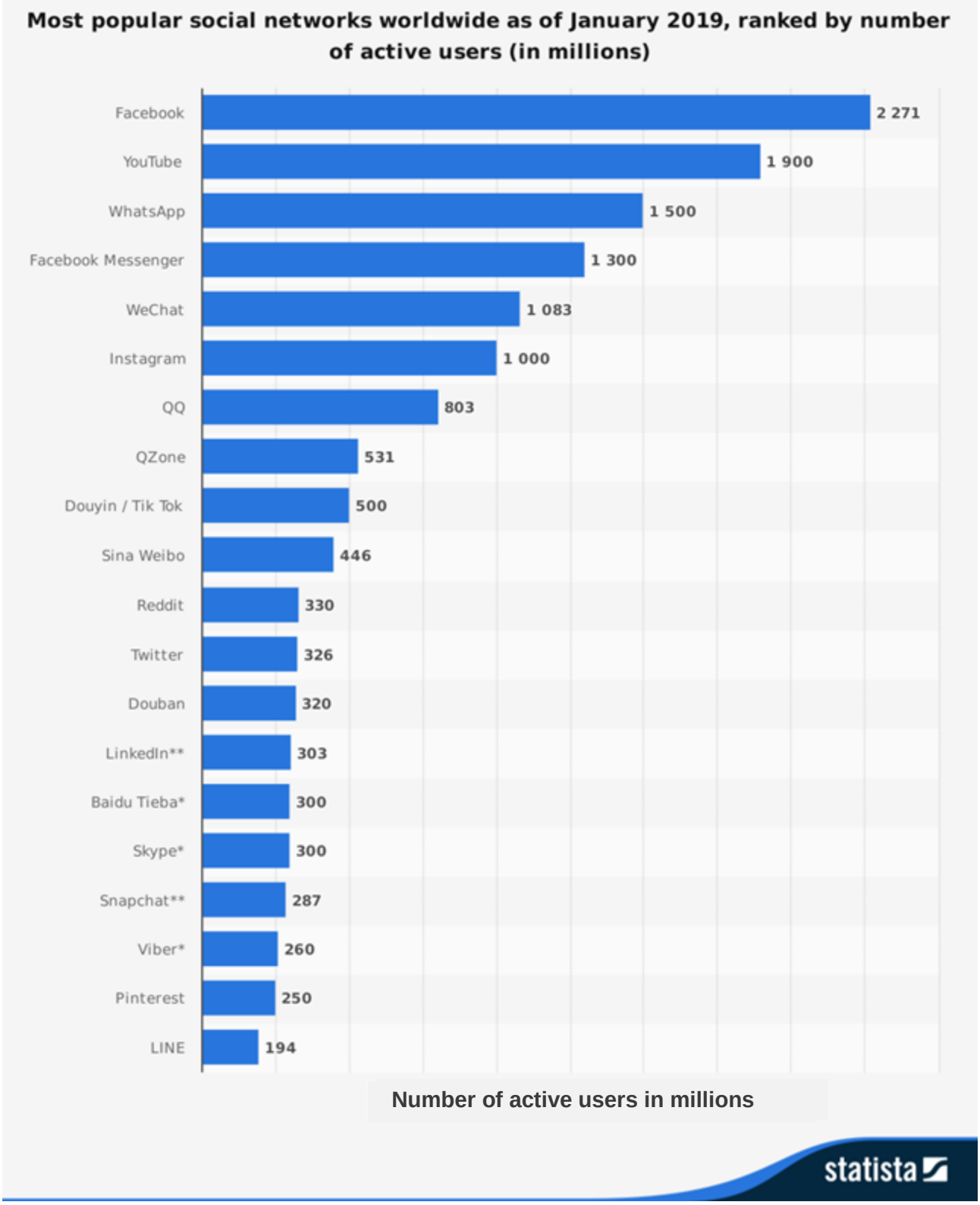

Figure 2.1: Social networks ranked by number of active users (Statista, 2019)

Bernard & Sobel (2009) described microblogging as a form of online word of mouth branding; they analysed 149,472 micro-blog postings containing branding comments, sentiments, and opinions and investigated the overall structure of these micro-blog postings and movement in positive or negative sentiment. They reported that research findings show that 80 percent of micro-blogs containing branding comments were information seeking or sharing. They also found that nearly 20 percent contained some expression of branding sentiments and of these, more than 50 percent were positive and 33 percent were critical of the company or product (Bernard & Sobel, 2009). This is the reason why business organisations are starting to poll these microblogs to get a sense of general sentiment for their product and services (Agarwal et al., 2011). People post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life and that companies manufacturing such products have started to poll these microblogs to get a sense of general sentiment for their product (Agarwal et al., 2011).

Microblogging sites such as Twitter facilitate and enable easily sharing messages either publicly or privately within a social network. Twitter can be used by business organisations as a fast, easy (and free) way to understand competitors and how they are performing. This platform can also be utilised by the business to keep in touch with their own clients for example by following them on Twitter in order to see what they are saying regarding their products and services. Twitter can be used to offer private discounts and sales announcements. It can also be used to provide internal updates to team members and employees and to get leads on business opportunities latest trends and news. The result is a large amount of useful data and information that can easily be created, shared, searched, promoted, disputed, and analysed.

2.5 Twitter a Microblogging Social Network

Twitter is a microblog and a social networking site which gives its users the ability to see what is happening in the world and what are people talking about right now (Twitter, 2018). It was created in March 2006 by Jack Dorsey, Evan Williams, Biz Stone and Noah Glass and was launched in July 2006 (Smith, 2019). Twitter is one of the most popular social networks worldwide, partly due to the fact that it provides the ability for its users to follow any other user with a public profile, which enables users to interact with their favourite

celebrities who frequently posts on their social media sites. Twitter is available in more than 40 languages around the world and the information available on Twitter ranges from breaking news and entertainment to politics, sports and everyday interests (Twitter, 2018). Users can access Twitter from their website: www.twitter.com via an array of mobile devices and SMS. According to (Statista, 2019) as of the fourth quarter of 2018, Twitter averaged at 321 million monthly active users as shown in Figure 2.2. On average around 6000 tweets are tweeted on Twitter every second, which corresponds to over 350 000 tweets sent per minute, which is equivalent to 500 million tweets per day and around 200 billion tweets sent per year (Smith, 2019). A live visualization of the tweets sent every second can be visualized from: http://www.internetlivestats.com/one-second/#tweets-band

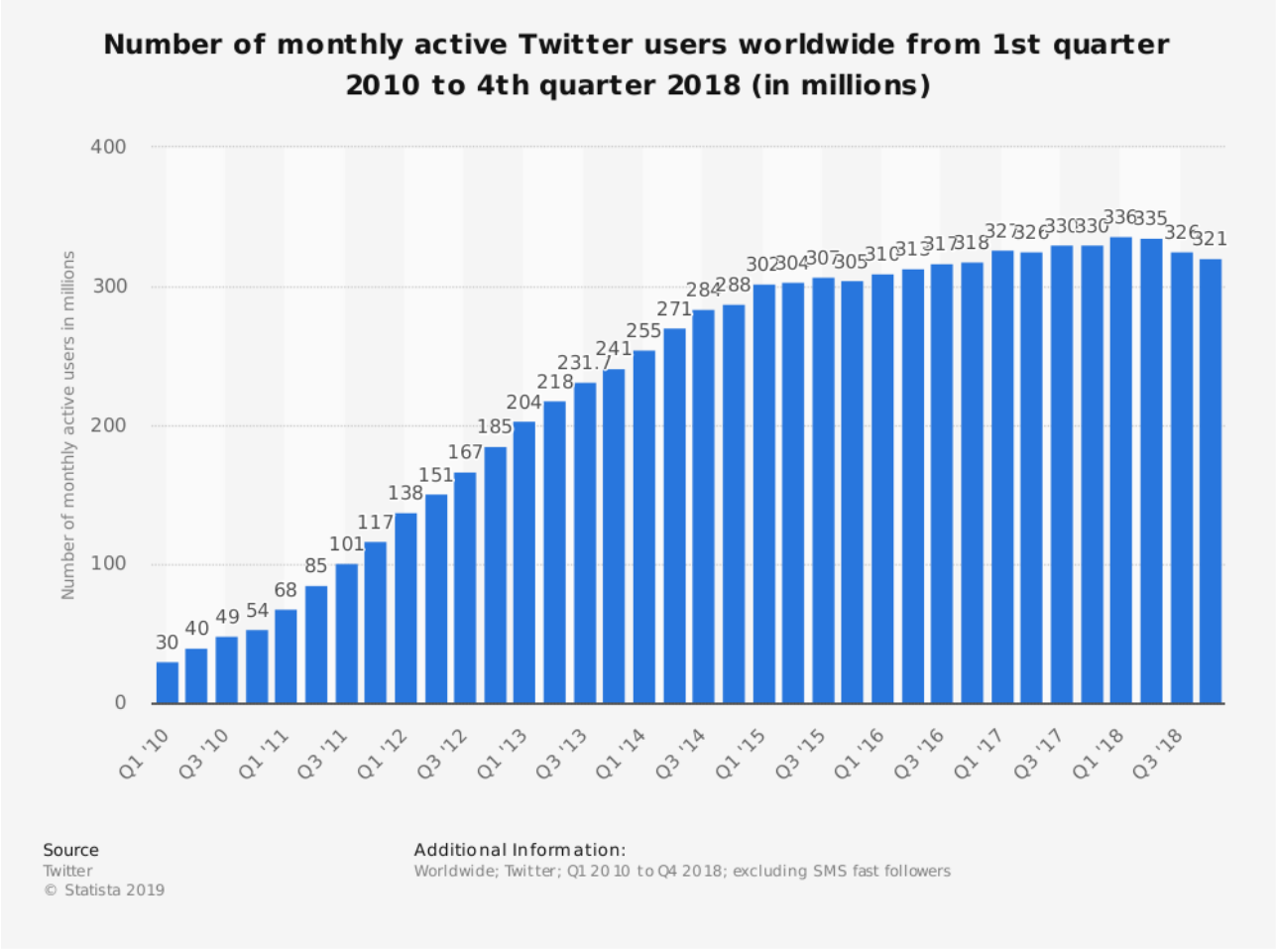

Figure 2.2: Number of active Twitter users worldwide (Statista, 2019)

Users interact on Twitter via posting short text messages called "tweets", to friends, or "followers" (Seyede, 2017). Statista (2019) describes Twitter as a social networking and microblogging service that enables its registered users to read and posts messages short

messages called “tweets”. These messages were initially limited to 140 characters, but have now since been increased to 280 characters and users are also able to upload photos and short videos (Twitter, 2018). It is this restriction that stimulates users to be very concise about their opinion, which makes Twitter a very rich source for sentiment analysis.

Twitter has several uses for both individuals and businesses. For personal use it is a good way to keep in touch with friends and quickly broadcast information about where you are and what you are up to, and for business it can be used to broadcast your company's blog posts, latest news and interact with customers, and/or to facilitate easy internal collaboration and group communication. Twitter is a free social networking and microblogging service that enables members to broadcast posts using multiple platforms and devices such as phone text message, desktop client or by posting at the Twitter.com website. Using Twitter anyone can follow anyone and Tweets are also posted on the Twitter website which makes them public and permanent, and are searchable which mean anyone can search tweets on Twitter, whether they are a member or not.

Twitter provides developer friendly streaming API for data retrieval purpose allowing the developers to search real time tweets from various users. Sheela (2016) mentions that Twitter provides two APIs, Stream API1 and the REST API2, and that the difference between the two is that the Streaming API supports long-lived connections and provides data in almost real-time where is the REST APIs support short-lived connections and are rate-limited meaning one can only download a certain amount of data using this API (usually 150 tweets per hour). This study takes advantage of the fact that Twitter uses an open-source API which is open and available to application developers. The present study utilizes this API to connect to the organisation’s Twitter timeline and download Tweets for data analysis.

2.6 Big Data

The term ‘big data’ has been used with several and inconsistent exceptions and lacks a formal definition. There is no single universally accepted definition for big data. According to Ward & Barker (2013) this is due to a shared origin between academia, industry and the media, so that various stakeholders provide diverse and often contradictory definitions. This lack of a consistent definition introduces ambiguity and hampers discourse relating to big data (Ward & Barker, 2013). De Mauro et al. (2016) mentioned that the degree of

popularity of the big data phenomenon has not been accompanied by a rational development of an accepted vocabulary.

All the definitions surveyed by Ward & Barker (2013) encompass at least one of the above factors, most encompass two. From these Ward & Barker (2013) have defined big data as a term describing the storage and analysis of large and or complex data sets using a series of techniques including, but not limited to: NoSQL, MapReduce and Machine learning. Mittal et al. (2018) described NoSQL as a database that provides a mechanism for storage and retrieval of data which is modelled in means other than the tabular relations used in relational databases. MapReduce is a computational paradigm, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster (Prudhvi et al., 2015). Robert (2014) has defined Machine learning as a set of methods that can automatically detect patterns in data. NoSQL databases are capable of processing large volumes of rapidly changing structured, semi-structured and unstructured data such as social network data.

Structured data is data that has well defined data definitions, stored in relational databases, often in tables (Kaisler et al., 2013). This data is comprised of clearly defined data types whose pattern makes them easily searchable. Abiteboul (1997) defined semi-structured data as data that is neither raw nor very strictly typed as in conventional database systems, which arise often when integrating several (possibly structured) sources. The term unstructured data on the other hand mean different things in different contexts (Blumberg & Atre, 2003). According to Blumberg & Atre (2003), unstructured data in the context of relational database systems, refers to data that can't be stored in rows and columns and this data must be stored in a BLOB (binary large object) or CLOB (character large object) available in most relational database management system (DBMS) software. Unstructured data is typically large text data and may include e-mail files, word-processing text documents, image files and video files. This data either does not have a pre-defined data model or is not recognized in a pre-defined manner. Merrill Lynch cited a rule of thumb in 1998 stating that, somewhere around 80 – 90% of potentially usable business information may originate in unstructured form (Christopher, 1998). Gandomi & Haider (2015) also stated that unstructured data constitute 95% of big data.

De Mauro et al. (2016) also provided a unified definition for big data and of the main research themes in literature. They concluded that the nucleus of the concept of big data includes the following aspects:

(1) “Volume”, “Velocity” and “Variety”, to describe the characteristics of data
(2) “Technology” and “Analytical Methods”, to describe the requirements needed to make proper use of such data
(3) “Value”, to describe the transformation of information into insights that may create economic value for businesses and society.

The consensual definition for big data was that, big data is the information asset characterized by such a High Volume, Velocity and Variety that require specific technology and analytical methods for its transformation into value (De Mauro et al., 2016). De Mauro et al. (2016) argued that their definition is compatible with the usage of terms such as “big data technology” and “big data methods” when referring directly to the specific technology and methods referenced in the main definition.

Big data is a term that describes the very large volume of data; both structured and unstructured that a business generates or receives on a day-to-day basis, but it is not the amount of data that is of important. The important factor is what organisations do with this data. Big data can be analysed for insights that enables better decision making and strategic business goals. According to Zohuri & Moghaddam (2017), the term “big data” is relatively new, but the act of gathering and storing large amounts of data for eventual analysis has been around for ages. Big data concept gained momentum in the early 2000s when industry analyst Doug Laney from META Group (now Gartner) articulated the now-mainstream definition of big data as the three Vs (Douglas, 2001):

(1) Volume (amount of data) refers to the quantity of generated and stored data. Its size determines the value and potential insight and whether it can actually be considered big data or not. This data is collected by the organisations from a variety of sources that includes business transactions, social media and information from sensors or machine-to-machine data.
(2) Velocity (speed of data in and out) is an indication of the rate at which data are generated and the speed at which it should be analysed and acted upon. Gandomi & Haider (2015) claim that the wide adoption of digital devices such as smartphones

and sensors has led to an increase in the rate at which data is created and is demanding a need for real-time analytics and evidence-based planning.

(3) Variety (range of data types and sources) refers to the different types of structured and unstructured data that organisations can collect, such as transaction-level data, video, audio, text and/or log files. Katal et al. (2013) argued that the data being produced is not of single category and that all this data is totally different consisting of raw, structured, semi-structured and even unstructured data which is difficult to process using existing traditional analytic systems.

While Gartner (2015) and now much of the industry, continue to use the "3Vs" model for describing big data, IBM scientists add a fourth "V" to the big data definition in addition to the three V's (Data & Hub, 2015). The fourth "V" is Veracity which is an indication of data integrity and the ability for an organisation to trust the data and be able to confidently use it to make crucial decisions.

The size of the databases used in today's businesses is growing at exponential rates each day. There is a need to store, process and analyse these large volumes of data for business decision making. In most business organisation and scientific applications, there is a need to process terabytes of data in efficient manner on regular basis. This has given rise to the big data problem faced by many organisations due to the inability of conventional software tools and database systems to manage or process the big data sets within acceptable time limits. Processing of data can include various operations depending on usage like sorting, tagging, highlighting, indexing and searching operations. It is not possible for single or few machines to store or process this enormous amount of data in a tolerable time period. If organisations can harness the power of big data, they can monetize data into valuable insights such as identifying emerging opportunities, improve the customer experience, enhance operational efficiencies, and reduce risks. Social media data has big data properties and this study focuses on introducing fast and efficient methods for handling and processing this type of data.

2.7 Data Mining and CRISP-DM

In Zhu (2007), the definition of data mining was given as, "data mining is the process of exploration and analysis by automatic or semiautomatic techniques, of large quantities of data in order to discover meaningful patterns and rules". According to Simoudis (1996) data

mining is the process of extracting valid, previously unknown, actionable and comprehensible information from large databases and using this information to make crucial and informed business decisions. In the data mining literature many authors argue that, the term data mining is appropriately named as “Knowledge mining from data” or simply “Knowledge mining” (Jain & Srivastava, 2013; Ali & Tuteja, 2014).
The essence of data mining is about solving problems by analysing data already present in existing databases. The data used in the data mining process must be invariably present in substantial quantities. The patterns discovered by this process must be meaningful i.e. it must lead to some advantage for the organisation for example an economic advantage. Useful and meaningful patterns from data mining facilitate making non-trivial predictions on new data as well.

Witten et al. (2016), claims that there are two dissipations for the expression of a pattern: as a black box whose inner workings are incomprehensible and as a transparent box which reveals the structure of the pattern. Witten et al. (2016) argue that the difference between the two is whether or not the patterns that are mined are represented in terms of a structure that can be examined, reasoned about and can be used to inform future decisions. Such patterns are called structural simply because they capture the decision structure in an explicit way, i.e. the pattern help explain something about the data.

Data mining is concerned with what kind of patterns that can be mined, so based on the kind of data that can be mined, Ali & Tuteja (2014) claim that they are two kinds of function involved in data mining: (1) Descriptive task – the objective of this data mining task is to derive patterns (i.e. correlations, clusters, trends trajectories and anomalies) which are able to summarize the underlying relationships in the target data. This task is mainly exploratory in nature and frequently require post-processing techniques to explain and validate the results (2) Classification and Predictive task – predictive data mining task involves using some variable or fields in the data set to predict unknown or future values of other variables of interest and generates the model of the system described by the given data set that can be utilized to perform tasks such as classification, prediction or estimation. Put simply the primary goal of a predictive data mining model is to predict the future outcomes based on passed records with known answers. The classification task requires that the data mining algorithm to partition the input space in such a way as to separate the examples based on their class. The data in the KDD process, possibly originates from many sources. It can be

data from Social media such as blogs, microblogs, or social networks. Azevedo & Santos (2008) mention that in the last years there has been a huge growth and consolidation of the data mining field and some efforts are being done that seeks the establishment of standards in the area.

CRISP-DM is an acronym that stands for Cross-Industry Standard Process for Data Mining (Chapman et al., 2000). The CRISP-DM process is an open standard process model that describes common approaches used by data mining experts which was developed by means of the effort of a consortium initially consisting of Daimler Chrysler (then Daimler-Benz), Statistical Package for the Social Sciences (SPSS) and NCR Corporation (originally National Cash Register)

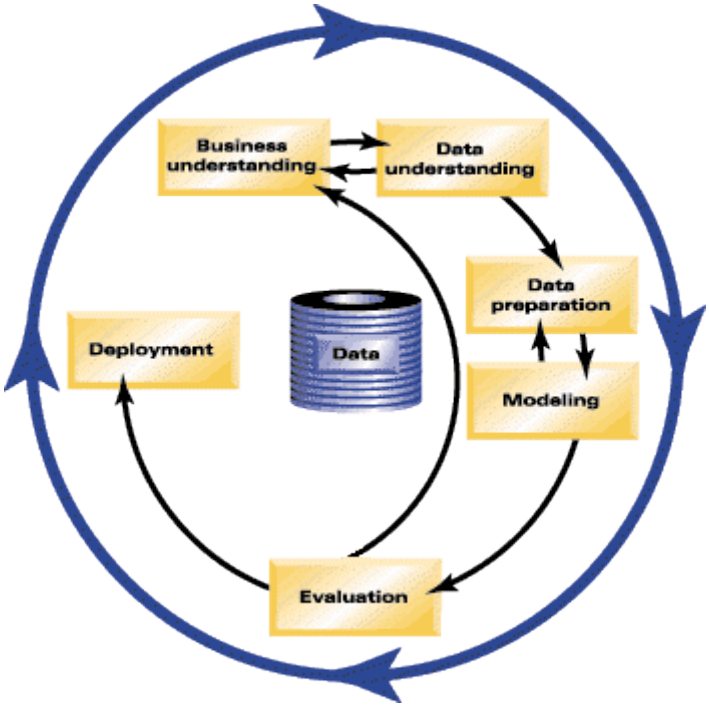

Figure 2.3: Phases of the CRISP-DM reference model (Chapman et al., 2000)

Azevedo & Santos (2008) state that the CRISP-DM process consists of six stages Figure 2.3 which are described below:

(1) Business understanding – this initial phase is about understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

(2) Data understanding – The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover first insights into the data or to detect interesting subsets to form hypotheses for hidden information.
(3) Data preparation - The data preparation phase covers all activities to construct the final dataset from the initial raw data.
(4) Modelling - In this phase, various modelling techniques are selected and applied and their parameters are calibrated to optimal values.
(5) Evaluation - At this stage the model (or models) obtained are more thoroughly valuated and the steps executed to construct the model are reviewed to be certain it properly achieves the business objectives.
(6) Deployment - Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented in a way that the customer can use it.

The activities in this study are a subset of the activities involved in data mining and CRISP-DM as discussed in the sections above. This dissertation aims to introduce automated methods and techniques to analysis large quantities of data in order to discover meaningful pattern in that data much like Data mining and CRISP-DM.

2.8 Text Mining

Text mining is also known as text data mining (Hearst, 1997) or knowledge discovery from textual databases. Fayyad et al. (1996), define text mining as the process of extracting interesting and non-trivial knowledge or patterns from unstructured text documents. According to Fayyad et al. (1996) and Simoudis (1996) text mining can be viewed as an extension of data mining or knowledge discovery from structured databases. A more recent definition of text mining by Younis (2015) defines text mining as the automated process of discovering and revealing new knowledge, relationships and patterns in unstructured textual data resources. Another definition of text mining by Feldman & Sanger (2007) describes text mining as a new and exciting area of Computer Science research that attempts to solve the crisis of information overload by fusing techniques from data mining, machine learning, information retrieval, natural language processing and knowledge management. Tan (1999) agrees with Feldman & Sanger (2007) in that, text mining is a multidisciplinary field, that also encompass information extraction, text analysis, information retrieval, clustering,

categorization, visualization, machine learning, database technology and data mining. This dissertation utilises a lot of the text mining techniques, such as machine learning, database technology and visualization when gathering data and conducting experiments.

Text mining is believed to have a high commercial value and according to Gupta & Lehal (2009) 80% of information is stored in text format and as much as knowledge can be extracted from other sources, unstructured text remain the largest readily available knowledge source. The importance of text data is amplified in Dietrich et al. (2015) where the authors claim that 80-90% of future world's data growth is expected to come from document databases or unstructured text databases and also in Allahyari & Kochut (2015) in research sponsored by International Data Corporation (IDC) the authors have predicted that by 2020 the volume of text data which are generated from Social media in a variety of forms will increase to 40 zettabytes (1 billion terabytes), which represents a 50-times growth since the beginning of year 2010. According to Allahyari et al. (2017) text mining techniques are related to the traditional data mining and knowledge discovery approaches.

2.9 Summary

The current study focuses on social media data analysis methods. In this Chapter background on social media was provided and other related terms. Literature review was conducted and important concept pertaining to this study was discussed in an effort to prepare and familiarize the reader for the subsequent chapters. The next chapter continues with the literature review and provides an in depth discussion of the literature related to data analysis methods available in literature for text data in general and social networks in particular.

CHAPTER 3: ANALYSING SOCIAL MEDIA DATA

This chapter continues the literature review from the previous chapter and explores the available data analysis methods which are applicable to social media data as well as the information that these methods provide including the benefits of such information to an organisation. The discussion provides answers to the following questions: (1) What type of information is found on social media? (2) Which analysis methods discussed in literature can be applied to social media data? (3) What type of information these analysis methods yield that can be of value to an organisation processing this information?

This Chapter is organized as follows: Section 3.1 starts off with a discussion on Social media analytics. Social Media analytics is divided into two broad categories i.e. Content-based analysis and Structure-based analytics. Section 3.2 discusses Content-based analysis including the traditional text analytics since most of the methods used in text Content-based analytics are from the traditional text analytics such as Text Mining, Clustering, Classification, Summarization, Information Extraction, Question Answering, Concept Linkage, Information Visualization and Sentiment Analysis. Section 3.5 provides a summary discussion of this chapter.

3.1 Social Media Analytics

Social media analytics is a term used to refer to the analysis of both structured and unstructured data found in social media (Gandomi & Haider, 2015). It is concerned with developing tools and platforms which can be utilized by an organisation to gather, analyse, monitor, visualize and summarize data from Social media, generally motivated by specific requirements from a target application (Zeng et al., 2010). According to Gandomi & Haider (2015) and Aggarwal (2011), there are two sources of data in social media. These are User Generated Content (UGC) and the relationship and interactions between the network entities. Using this categorization, there are two groups that can be used to categorize social media analytics, that is Content-based analysis and Structure-based analytics (Gandomi & Haider, 2015). In this dissertation the author narrows the discussion to focus on Content-based analysis mainly focusing on sentiment analysis on text data.

The main reason for focusing on text data is that, the research aims to analyse user sentiments of which the majority of user sentiment is contained in text data. This study focuses on analysis methods for content information in order to yield valuable insight in the social network data. Content-based analysis methods are discussed in the next section.

3.2 Content-based analysis

Content-based analysis focuses on the analysis of user generated content posted by users on social networks for example images, videos, customer feedback, product reviews, and bookmarks. This type of data has big data characteristics in the sense that it is unstructured, noisy, dynamic and voluminous. Social networks are rich in various kinds of contents such as text and multimedia data because of the wide variety of methods by which users can contribute content to the network. The analysis of multimedia data such as images and videos is outside the scope of this study as the main focus of the analysis of the text content from social networks. The ability to utilize text mining algorithms effectively in the context of social networking data is very critical for a variety of applications. According to Aggarwal & Wang (2011), social networks require text mining algorithms for a wide variety of tasks such as key word search, classification and clustering. The discussion will first introduce traditional text analytics methods and the information that they provide, since they can also be applied to text content from social networks.

3.2.1 Text Mining Approaches

Text analytics techniques from text mining can be implemented to efficiently deal with textual data in social media for both research and business purposes. Figure 3.1 shows a traditional framework by Hu & Liu (2012) for text analytics in social media.

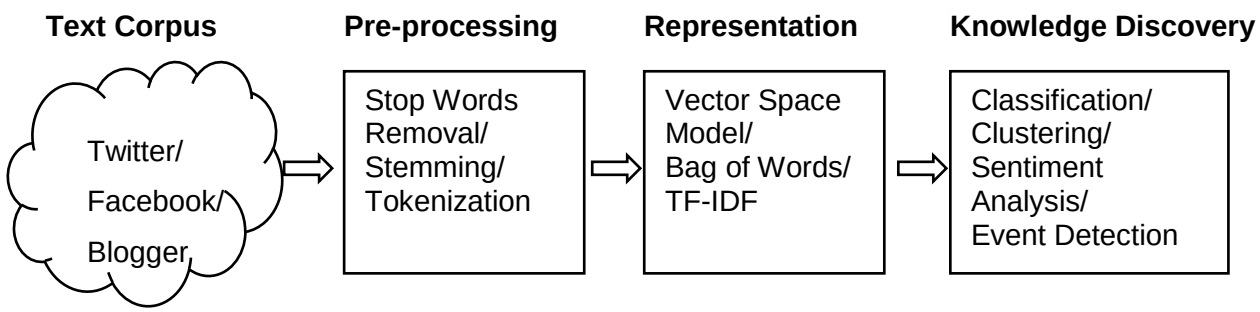

Figure 3.1: A traditional framework for text analytics (Hu & Liu, 2012)

A text mining process begins by retrieving a document from a collection of documents and analysing the document by checking for character set or its document format. Then a repetitive text analysis process is executed until all the information is retrieved from the document. In the text mining example shown in Figure 3.2, for this specific target information only Information extraction, Clustering and Summarization are shown, but many other combinations of techniques can be utilised depending on the target information to be captured. The results of this process are useful information that can be stored in a management information system as knowledge for the organisation.

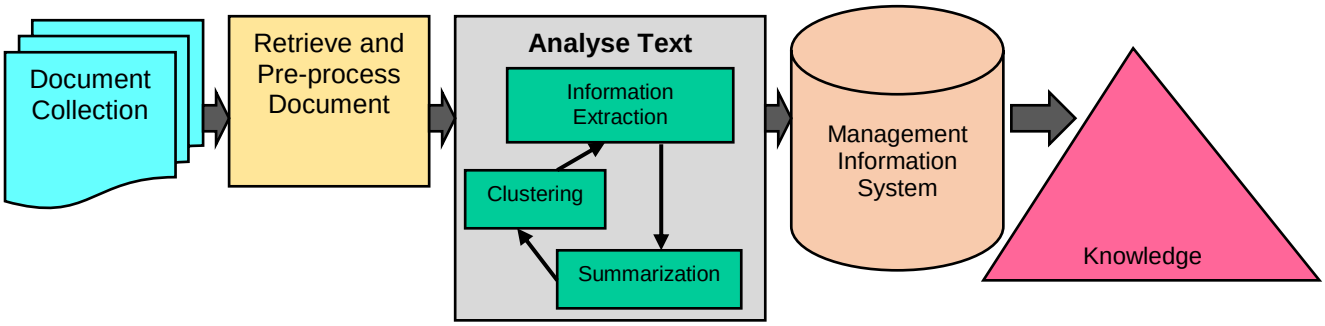

Figure 3.2: An example of text mining (Fan et al., 2006)

Text mining involves building technology that combines a human's linguistic abilities with the accuracy and speed of computers. Fan et al. (2006) stated that the field of natural language processing (NLP) has produced technologies that enable computers to process natural language, enabling them to analyse, understand and even generate text. Clustering, categorization, summarization, information extraction, concept linkage, question answering and Information visualization are some of the technologies in text mining according to Fan et al. (2006).

3.2.2 Classification

Text classification (or text categorization) is a text processing technique of automatically assigning predefined categories to unlabelled text documents (Selvi et al., 2017; Sebastiani, 2002). According Barbier & Liu (2011), classification is a common supervised approach and mainly utilized when the data set has labels or a subset of the data set has labels. Put simply classification is the task of identifying the main themes of a document (Yang & Pedersen, 1997b). The main difference between clustering and classification is that in clustering the documents are clustered in real time as opposed to using predefined topics or labels as in

classification. The classification process begins with the classification algorithm and a set of training data which includes class labels for each data element as shown in Figure 3.3.

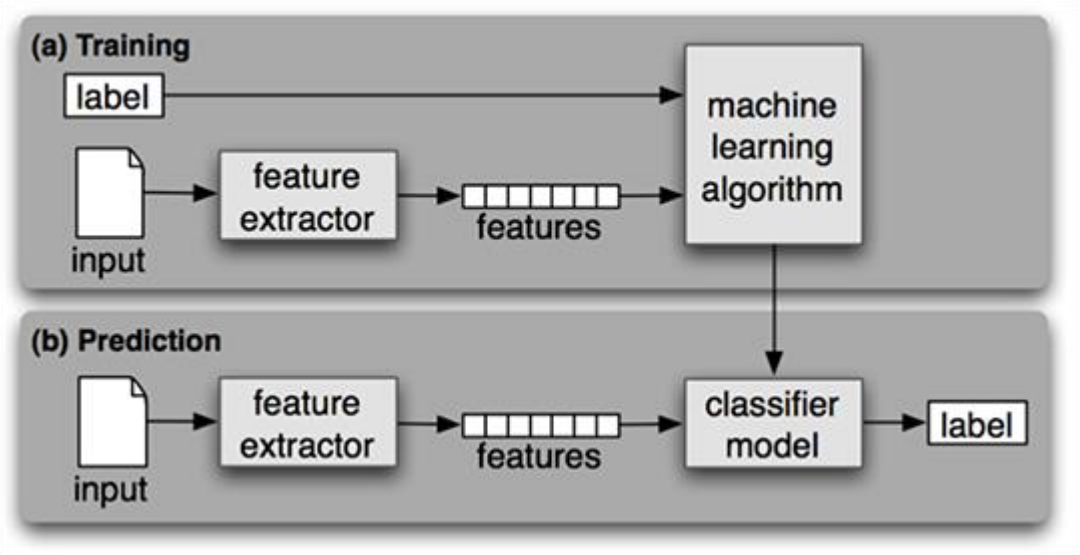

Figure 3.3: Supervised classification (Bird et al., 2009)

The algorithm learns from the training data and builds a model which will be used to automatically categorize new text documents into the relevant distinct class label provided with the training data. The increasing importance of text classification is mainly due to the increase in textual information and digital documents available online, raising the need for indexing and summarization of these documents to facilitate effective retrieval. Classification is performed using Machine learning (ML) techniques and Statistical classification methods in which a general inductive process automatically construct an automatic text classifier by learning from a set of already pre-classified documents the characteristics or properties of the categories of interest (Sebastiani, 2002). During the classification of particular documents, the classification algorithm treats the documents as a data structure called a bag of words. This way the algorithm does not attempt to process the actual information in the documents like in the case of information extraction. Using the bag of words the classification algorithm count only the words that appear in the document and using this count, the program identifies the main topics covered in the document. The final phase involves a ranking method for the documents in which the documents are ranked in order of which document has the most content on a given topic. Utilizing machine learning approaches yield higher accuracy results compared to the results achieved by human experts, not to mention the cost saving in terms of expert man power, since there is no

involvement from either knowledge engineers or domain experts to construct the classifier or for changing to different sets of categories (Sebastiani, 2002).

3.2.3 Sentiment Analysis

The current study focuses on performing sentiment analysis on social networks text data, primarily looking at Twitter data. Sentiment analysis is a “big suitcase” of natural language processing (NLP) problems (Poria et al., 2017) and is relatively an old research problem (Liu, 2012) with much of the research being focused mostly on extracting sentiment from conventional text i.e. formal text such as documents or text found on traditional online media platforms such as blogs and news platforms. It involves solving and tackling many NLP tasks such as Named entity recognition (Ma et al., 2016), Aspect extraction (Poria et al., 2016), Sarcasm detection (Poria et al., 2016), Concept extraction (Rajagopal et al., 2013), Personality recognition (Majumder et al., 2017) etc.

Sentiment analysis or opinion mining is perhaps the most common social networking data analysis method which has existed since the early 2000s (Pang & Lee, 2008). Zhang & Liu (2016) have defined sentiment analysis or opinion mining as the computational study of people’s opinions, sentiments, attitudes, appraisals and emotions regarding entities and their aspects expressed in text. In another study Vinodhini & Chandrasekaran (2012) have defined sentiment analysis as a type of natural language processing for tracking the mood of the public about a particular product or topic. Twitter has been used for sentiment analysis in a lot of studies for different purposes for example, the sentiment analysis on Twitter has been run yearly at SemEval (Semantic Evaluation) since 2013 (Hltcoe, 2013; Hendrickx et al., 2013; Rosenthal et al., 2015; Nakov et al., 2016; Rosenthal et al., 2017). SemEval is an ongoing series of evaluations of computational semantic analysis systems which explore the nature of meaning in language.

Traditionally consumers would gather feedback from their trusted friends or family before making a decision to purchase a product, but today the trend has shifted to identifying the opinions of a variety of individuals from around the world using social networks and product review forums. In general, the purpose of sentiment analysis is to determine the attitude of the speaker, author or other subject with respect to some topic or overall contextual polarity or emotional reaction to a document, interaction or to an event. An example of the sentiment

analysis process for a product review is shown in Figure 3.4. As illustrated in Figure 3.4, sentiment analysis can be considered a classification process whose target is to find opinions, identify the sentiments they express and then classify their polarity.

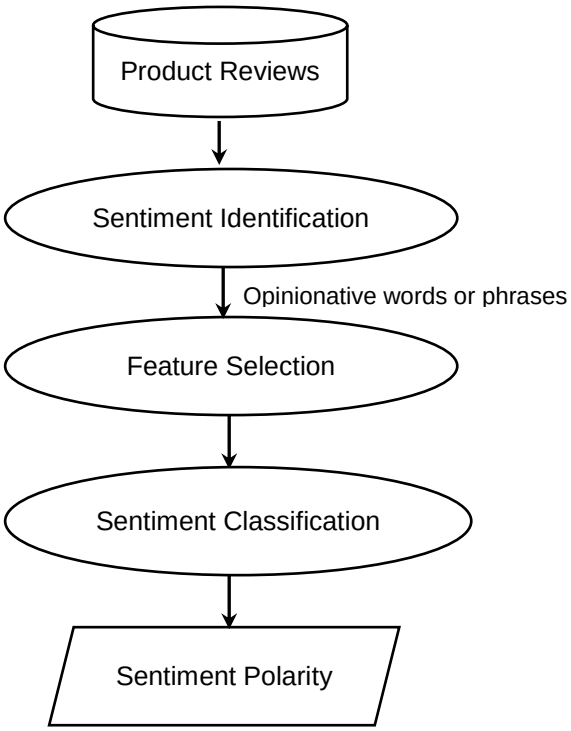

Figure 3.4: Sentiment analysis process on product reviews (Medhat et al., 2014)

Opinions are so important not only for individuals, but for organisations as well, that whenever there is a need to make a decision people want to hear other people's opinions. Liu (2012) maintains that textual information in the world can generally be categorized into two main categories, which are facts and opinions. Facts refer to objective expressions about events, entities and their characteristics, whereas opinions are generally subjective expressions that describe people's sentiments, feelings or appraisals concerning entities, events and their characteristics. Given that the concept of opinion is very broad, this study focuses on opinion expressions that convey people's positive or neutral or negative sentiments. An organisation performing sentiment analysis can download data generated from their social networking site. This involves the organisation building a system that connects to social networking sites' public APIs such as the Twitter API to collect data and examine opinions about the product and/or services made in blog posts, comments, reviews or tweets. In this study to achieve speed and efficiency big data technologies were utilized to perform sentiment analysis of social networks data. Hobbs (2014) from The Parliamentary

Office of Science and Technology in London advocates that sentiment analysis can provide comprehensive insights on public reactions to specific events in ways that have not previously been possible.

According to Thelwall et al. (2011) there exists three different methods in Sentiment analysis:

(1) Machine learning based methods – these methods are based on models that are calibrated to categorised data (training data). The calibrated model or machine can then be used to categorise new data. This works in the same way that a parameterised equation works when predicting the value of the response variable in regression analysis. The training is based on features (or words) which that have an effect on the data polarity.

(2) Lexical based methods – this approach is based on constructing a Lexicon (a structure that keeps track of words and possibly information about them) where the words are called, lexical items. Using the lexicon the overall polarity of the text is determined by a possibly weighted count of those lexical items.

(3) Linguistic analysis – this approach utilises the syntactic characteristics of the words, phrases, negations and structure of the text to estimate the text orientation. The linguistic approach is usually combined with a Lexicon based approach method.

In another study by Medhat et al. (2014), the authors argued that sentiment classification can be roughly divided into Machine learning approach and Lexicon based approach as depicted in Figure 3.5.

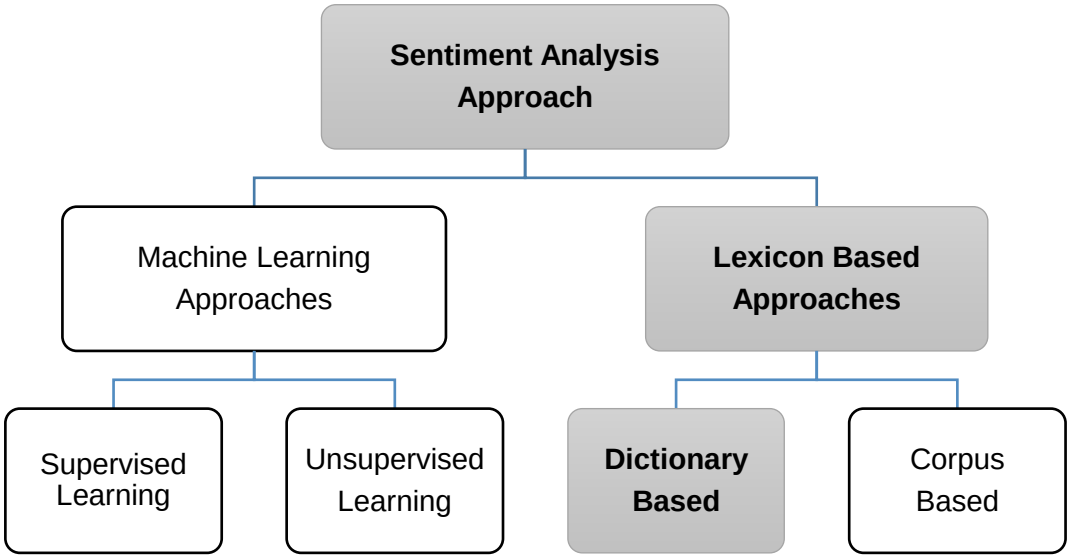

Figure 3.5: Sentiment classification approaches

3.2.4 Machine Learning Approaches

In machine learning there are generally three learning methodologies, i.e. supervised learning, unsupervised learning and semi-supervised learning (Narendra et al., 2016). In supervised learning the system builds the classification model based on labelled data, whereas in unsupervised learning the learning mechanism is built using unlabelled data and semi-supervised the classifier is build based on both labelled and unlabelled data (Narendra et al., 2016). According to Vinodhini & Chandrasekaran (2012) machine learning approaches treat sentiment analysis as a text classification problem and they mainly belongs to the supervised learning classification in general and text classification methods in particular. In a supervised machine based learning system, there are two set of documents required, i.e. the training set and the testing set. The training set is used to train the classifier to learn to differentiate characteristics of documents and a test set is utilized to validate the performance and accuracy of the classifier, hence the name “supervised learning”. Several machine learning techniques have been adopted for text classification (Pang et al., 2002), such as Naïve Bayes, Support Vector Machines, Maximum entropy, Decision tree, K-Nearest neighbours, ID3, C5 etc. Naïve Bayes classification which is one of the most commonly used machine learning technique is discussed in more detailed in the next section.

3.2.4.1 Naïve Bayes (NB)

The Naïve Bayes algorithm is a widely adopted algorithm for document classification (Tan et al., 2009). Multinomial model and multi-variate Bernoulli model are two of the commonly used models for text categorization. The Naïve Bayes is a probabilistic classifier, based on the Bayes theorem and perhaps one of the simplest classifier to code in any programming language due to the simple mathematics involved (Hu & Liu, 2004). The Bayes' theorem is stated mathematically as in (3.1):

$$P(A\,|B) = \frac{P(B\,|A)\,P(A)}{P(B)} \tag{3.1}$$

where A and B are events and $P(B) \neq 0$. $P(A\,|B)$ is a conditional probability of event A occurring given that B has occurred. $P(B \mid A)$ is the likelihood of event B occurring given that A has occurred. $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other and this is known as the marginal probability. In order to predict the probability that a given feature set belongs to a particular label, the Naïve Bayes adapts the above equation as follows:

$$\mathrm{P(label|features)} = P(label) * P(feature|label)/P(features) \tag{3.2}$$

In text classification A and B in (3.2) are replaced by $label$ and $features$ as show in (3.2). The label represents a sentiment class and the features represent the word events in the document. In this paper the multinomial model is discussed, in which a document is an ordered sequence of word events which are drawn from the same vocabulary V. The assumption is that the lengths of documents are independent of class. According to Tan et al. (2009), a Naïve Bayes assumption is made which states that, the probability of each word event in a document is independent of the word's context and position in the document i.e. each document d_i is drawn from a multinomial distribution words with as many independent trials as the length d_i. This results in the so called "bag of words" representation for document which was proposed in Pang et al. (2002). The bag of words is an unordered document representation which constructs a word presence feature set from all the words of a text instance, where each word becomes a key with the value 'True'. Using the bag-of-

words model the position of the words is ignored (i.e. the bag of words assumption) and the frequency of each word is used (Pang et al., 2002).

Using the multinomial model, an estimate of the probability of a word given its class is obtained as following,

$$P(w_t/c_k) = \frac{\Sigma_{i=1}^{|D|} \mathrm{N}_{t,i} \,.\, P(C_k \backslash d_i) + 1}{\Sigma_{t=1}^{|V|} \Sigma_{i=1}^{|D|} \mathrm{N}_{t,i} \,.\, P(C_k \backslash d_i) + |V|} \tag{3.3}$$

where $\mathrm{N}_{t,i}$ is the number of appearances of word w_t in document d_i, $|V|$ and $|D|$ refer to the vocabulary size and dataset size respectively. The class prior probability can be estimated as the Maximum Likelihood Estimate:

$$P(c_k) = \frac{\Sigma_{i=1}^{|D|} P(C_k \backslash d_i)}{|D|} \tag{3.4}$$

As a result the conditional probability of $P(C_k \backslash d_i)$ can be estimated as,

$$P(c_{\,k} \mid d_i) \propto \; P(C_k) \prod_{t \in |V|} (P(W_t \backslash C_k))^{\mathrm{N}_{t,i}} \tag{3.5}$$

The Bayes' rule can finally be written down for classification decision as following,

$$c = arg\ max_{C_k} \left(P(C_k) \prod_{t \in |V|} (P(W_t \backslash C_k))^{\mathrm{N}_{t,i}} \right) \tag{3.6}$$

The training method consists of relative-frequency estimation of $P(label)$ and $P(feature_i \mid label)$, using add-one smoothing. Add-one smoothing also known as Laplace smoothing, is a technique used to eliminate zeros for unknown or small amount of data or unseen features during training by adding 1 (Cherian & Bindu, 2017). The number of possible words is then added to the divisor, to balance this addition, so that the division will never be greater than one. Add-one smoothing is done in order to 'smooth' the probability estimates for n-gram models so that any n-gram component is given a non-zero probability

for words/features which do not occur in the particular sample during training. An n-gram is a sequence of N words, a bigram (or 2-gram) is a two-word sequence of words like “good morning”, or “thank you” and a trigram (or 3-gram) is a three-word sequence of words like “how are you” or “take good care” (Martin & Jurafsky, 2009). The pseudocode of the Naïve Bayes algorithm discussed above is shown in Figure 3.6.

function TRAIN NAIVE BAYES(D, C) **returns** log $P(c)$ and log $P(w|c)$

for each class $c \in C$ # Calculate $P(c)$ terms
  N_{doc} = number of documents in D
  N_c = number of documents from D in class c
  $logprior[c] \leftarrow \log \frac{N_c}{N_{doc}}$
  $V \leftarrow$ vocabulary of D
  $bigdoc[c] \leftarrow$ **append**(d) **for** d $\in$ D **with** class c
  for each word w in V # Calculate $P(w|c)$ terms
    $count(w,c) \leftarrow$ # of occurrences of w in $bigdoc[c]$
    $loglikelihood[w,c] \leftarrow \log \frac{count(w,c) + 1}{\sum_{w' \text{ in } V} (count(w',c) + 1)}$
return $logprior$, $loglikelihood$, V

function TEST NAIVE BAYES($testdoc$, $logprior$, $loglikelihood$, C, V) **returns** best c

for each class $c \in C$
  $sum[c] \leftarrow logprior[c]$
  for each position i in $testdoc$
    $word \leftarrow testdoc[i]$
    if $word \in V$
      $sum[c] \leftarrow sum[c] + loglikelihood[word,c]$
return $\text{argmax}_c\ sum[c]$

Figure 3.6: Naïve Bayes Algorithm

3.2.5 Sentiment Analysis Levels

Apart from the different sentiment analysis approaches identified above, sentiment analysis is usually conducted at one of the three levels (Liu, 2012):

(1) Document level – at the document level of analysis the task is to classify whether an entire document expresses a positive or a negative sentiment (Pang et al., 2002; Vijyalaxmi et al., 2013). For example given a document the system determines whether the author is expressing a positive or a negative opinion for that specific topic. In the case of product reviews, the system determines whether the review expresses an overall positive or negative opinion about that product. According to Liu (2012) this is task is

commonly known as document-level sentiment classification. Liu (2012) also points out that the challenge with document level analysis is that it assumes that each document expresses opinions on a single entity (e.g., a single product or a single movie). Since this assumes there is only one entity, it is not the most suitable for documents which evaluates or compare multiple entities.

(2) Sentence level – the sentence level is similar to the document level except that at the sentence level the task is to classify each sentence individually to see if it expresses a negative, neutral or positive opinion (Liu, 2012). Sentence level adds more flexibility than the document level due to the fact that it is able to differentiate the objective sentences from the subjective sentences and this can be used as first filter (Turney, 2002). It is also important to note that there are objective sentences expressing opinions and subjective sentences not transmitting any sentiment. Another important aspect to point out regarding sentence level sentiment analysis is that the valence of a sentence is not simply the sum of the polarities of its constituent words, because automatic systems learn a model from labelled training data using a large number of features such as word and character n-grams, valence association lexicons, negation lists, word clusters and even embedding-based features (Cambria et al., 2017).

(3) Entity and aspect level – this is the most fine-grained level of sentiment analysis, previously known as the feature level (Liu, 2012). Unlike both document level and sentence level which do not discover what exactly people liked or did not like, aspect level finds the target for each opinion instead of focusing on language units, like sentences, documents or paragraphs. The goal of entity and aspect level is to identify the opinion of sentiment on entities and their different aspects. According to Liu (2012) an opinion without its target being identified is of little use, so aspect level is based on the idea that an opinion consists of a sentiment(which is either positive or negative) and a target (of the opinion). For example, a sentence like “although the interior design of this car is very beautiful, its engine performance is very poor” has a clear positive tone, but the sentence is not entirely positive. The sentence can be categorized as positive with regard to the interior design, but very negative about its engine performance. Liu (2015) states that the majority of real-time sentiment analysis systems are based on aspect level sentiment analysis.

3.2.6 Lexicon-based Approaches

In this study one of the experiments utilizes the lexicon-based approach to perform sentiment analysis on Twitter data regarding movie reviews. The Lexicon-based techniques work on the assumption that the collective polarity of a document or sentence is the sum of polarities of the individual words or phrases in the document (Turney, 2002; Kaushik & Mishra, 2014). Lexicon-based approaches can be further subdivided into two approaches i.e. Dictionary-based approach and Corpus-based approach, which are discussed in more details in the next section. Dictionaries for lexicon-based approaches can be created manually as described in (Tong & Koller, 2001; Stone et al., 1966) or automatically using seed words to expand the of words (Hatzivassiloglou & McKeown, 1997; Turney, 2002; Turney & Littman, 2003). The majority of the lexicon-based research has focused on using adjectives as indicators of the semantic orientation of text (Hatzivassiloglou & McKeown, 1997; Hu & Liu, 2004; Taboada et al., 2006).

Taboada et al. (2011) suggested that when performing sentiment analysis using the lexicon-based methods the first step is to compile a list of adjectives and corresponding sentiment orientation values into a dictionary. After that, for any given text, all the adjectives in that text are extracted and annotated with their sentiment orientation value, utilizing the dictionary scores. The sentiment orientation scores are finally aggregated into a single score for the text.

An example of a sentiment calculation algorithm for Twitter data using the lexicon-based approach by Yadav & Elchuri (2013) is shown in Figure 3.7. The sentiment calculation shown in Figure 3.7 is based on a set of heuristics built on the sentiment orientation of the words. In this calculation blind negation words are extracted from the sentence i.e. words which point out the absence or presence of some sense that is not desired in a product feature (Yadav & Elchuri, 2013). The occurrence of the blind negation in a text automatically indicates negative sentiment. If a blind negation word is found in the text then the rest of the text is skipped and the sentiment is blindly assigned a negative score. The next step involves the extraction of the sentiment words. If negation words (i.e. words which reverse the polarity of sentiment) occur in proximity (2 word distances) can change the overall sentiment polarity of the word and if a sentiment word is not found in the text, the sentiment negation word

becomes additive to the negative sentiment list. Finally the sentiment of the tweet text is aggregated as the sum of the sentiments from all the variables.

```
Data: Pre-processed Twitter data
Result: Output: Positive, Negative, Neutral
Find the list of sentiment words SentiList, its position in the sentence;
Find the list of sentiment negation words SentiNegat, its position in the sentence;
Find the list of blind negation words BlindNegat, its position in the sentence;
if BlindNegat then
        return negativity;
else
        if SentiList and SentiNegat then
                foreach word in the SentiList do
                        if word is at most the distance of 2 from SentiNegat then
                                Revert the polarity of the word;
                        end
                end
        else
                if SentiNegat then
                        Add the SentiNegat to the negative SentiList;
                end
        end
end
SentiSum=0;
foreach word in the SentiList do
        SentiSum=SentiSum+sentiment of word;
end
if Hashtag is present then
        Find all the sentiment words in hash tag using regex matching and
        add them to SentiList
end
if Emoticon is present then
        Find sentiment of the emoticon and
        add emoticon, it's sentiment to SentiList
end
SentiType="neutral";
if SentiSum > 0 then
        SentiType="positive";
end
if SentiSum < 0 then
        SentiType="negative";
end
return SentiType;
```

Figure 3.7: Twitter Data Sentiment Calculation (Yadav & Elchuri, 2013)

3.2.6.1 Dictionary-based Approach

The dictionary-based approach involves using a dictionary which contains synonyms and antonyms of a word (Rajput & Solanki, 2016). One of the simple techniques when using this approach is based on using a small set of seed opinion words and an online dictionary such as SentiWordNet (Baccianella et al., 2010) or SenticNet (Cambria et al., 2014). The idea is to collect a small set of opinion words manually with known orientations and then to

subsequently grow this list by searching in the online dictionary for their synonyms and antonyms (Aung & Myo, 2017). The newly discovered words are appended to the seed list. This occurs in an iterative process that only stops when no more new words are found. At the end of the iterative process, manually inspections can be conducted to edit and correct errors.

3.2.6.2 Corpus-based Approach

The corpus-based approach is a data-driven approach which provides access not only to the sentiment labels, but also to the domain context which is a huge advantage of this method (Aung & Myo, 2017). Although Liu (2010) argued that using the corpus-based approach alone to identity all opinion words is not as effective as the dictionary-based approach because it is hard to prepare a huge corpus to cover all English words. The corpus-based approach depends on syntactic patterns or patterns that occur together along with a seed list of opinion words to find other opinions in a large corpus. According to Rajput & Solanki (2016) there are two methods in the corpus based approach:

(1) Statistical Approach – using the statistical approach, if a word appears intermittently in the middle of positive words, then its polarity is positive and if it appears frequently among negative words, then its polarity can be considered negative. In the event that the word has equal frequencies, it can be considered neutral.

(2) Semantic Approach – this approach assigns similar sentiment values to semantically close words. Neviarouskaya et al. (2010) suggested that the semantically close words can be obtained by getting the list of sentiment words and repeatedly growing the initial set with synonyms and antonyms and then determining the sentiment polarity for an unknown word by the relative count of positive and negative synonyms of that word.

3.3 Summary

The goal of a sentiment analysis system involves building a system that collects and examines public opinions regarding written text such as opinions made about products and services in blog posts, comments, reviews or in tweets. There are two main approaches in accomplishing this task, namely machine learning and the list or corpus approach. This dissertation adopts the Naïve Bayes for the text pre-processing experiments and the list or

corpus approach with AFINN-111 sentiment lexicon for sentiment analysis experiments which are discussed in more detail in Chapter 5. AFINN-111 is perhaps one of the simplest and most widely adopted lexicons that have been used extensively for sentiment analysis and was built specifically for microblogs such as Twitter. This chapter reviewed literature on some of the prominent data analysis methods which are applicable to social networking data. Sentiment analysis, text mining and graph mining are some of the methods which can be utilised in the analysis of social networking data. The next chapter examines and determines how speed and efficiency can be achieved using these analysis methods.

CHAPTER 4: TECHNOLOGIES FOR BIG DATA RETRIEVAL AND PROCESSING

Speed and efficiency are of paramount importance when processing social network data. This chapter discusses the importance of speed and efficiency when processing social network data and it also explores ways and technologies that can be utilized by an organisation in order to achieve these goals. When analysed properly and on time data generated from social networks gives valuable results. This chapter provides answers to the following research questions: (1) How can speed and efficiency be achieved when working with big data? (2) What technology can be utilized to process social media data fast and efficiently? (3) Can big data technology be employed to process social media data? (4) How can social media data be stored efficiently? (5) How can data that is generated in large volumes, in a variety of different formats that arrives at high velocity be handled in real time as it is being generated?

Social media data has big data properties and big data is a collection of large data sets that cannot be handled or processed using traditional computing technologies. Apache Hadoop is an open source framework which was designed for processing large data sets in parallel across cluster nodes in order to handle large and complex unstructured data sets such as social network data which does not fit into tables and requires a lot of processing. Hadoop and its related projects are collectively known as the Hadoop Eco System (Chaturvedi et al., 2015). The major components of Hadoop are the Hadoop Distributed file system (HDFS) and the MapReduce programming model.

This Chapter starts off with a discussion on how speed is achieved using Apache Hadoop in Section 4.1. A comprehensive discussion on Apache Hadoop a technology that is used by many organisations as a solution to big data challenges is provided in Section 4.2. In Section 4.3 an in-depth discussion on MapReduce is provided. Section 4.4 discusses Hadoop Flume a data ingestion tool used to move log data generated by application servers into Hadoop Files System. A brief description of Apache Manhout, a machine learning library from Apache is provided in Section 4.6. The Chapter ends with a discussion and a conclusion in Section 4.7 and Section 4.8 respectively.

4.1 Characteristics of Social Media Data

It was stated in Chapter 3 that data generated from social media data has big data characteristics. Kumari (2016) has argued that the real time data obtained through social media is complex and possesses the 3Vs' of big data. Recalling from De Mauro et al. (2016) big data is described as the information asset that is characterized by high volume, high velocity and variety which requires specific technology and analytical methods for its transformation into value. The challenge of velocity comes with the need for speed in order to handle the speed with which new data is created or existing data is updated while the system is required to make sense of the data immediately upon its creation (Chen et al., 2013). This means for organisation to process this information and gain value from it, they need to utilize technology that is capable of processing data with such characteristics. “Big data Represents the progress of the human cognitive processes, usually includes data sets with sizes beyond the ability of current technology, method and theory to capture, manage, and process the data within a tolerable elapsed time” (Graham-Rowe et al., 2008).

In the definition of big data by Graham-Rowe et al. (2008) this study is mainly interested on the emphasis of “tolerable elapsed time”. In any data processing environment time is of the essence and given the nature of big data due to its massive size processing this kind of data will definitely cause a bottleneck if the issue of speed is not addressed. The unprecedented data volumes generated from social network platforms such as Twitter require an effective data analysis and prediction platform to achieve fast response and real time classification for such big data (Wu et al., 2014). The need for fast real time data processing requirements of big data can be achieved using Apache Hadoop which is an open-source framework for distributed storage and distributed processing of very large data sets on computer clusters and is discussed in more detail in Section 4.4.

The volume of data generated from social network raises the issue of efficiency in processing this massive data. In most cases, the knowledge extraction process has to be very efficient and close to real-time because storing all observed data is nearly infeasible (Wu et al., 2014). This study relies on Apache Hadoop an open source framework from Apache to address the issue of efficiency. Stoneman (2017) has defined Hadoop as a big data platform with two functions, storing huge amounts of data in safe, reliable storage, and running complex queries over that data in an efficient way.

4.2 Apache Hadoop

The Apache Hadoop software library is an open-source framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware, licensed under the Apache License 2.0 (Apache Hadoop, 2016). Lock (2016) has defined Hadoop as an open source project from the Apache Software Foundation (ASF) which provides distributed processing of large data sets of multiple varieties designed for scalability and flexibility in managing big data. Looking at this definition, it can be readily deduced that Hadoop is capable of storing and analysing big data from Social media such as Twitter data efficiently, which is the focus of this research.

According to Olson (2010) beginning in the early 2000s, Google faced a serious challenge of crawling, copying, and indexing the entire Internet continuously, this led to Google's engineers designing and building a new data processing infrastructure to solve this problem. Olson (2010) emphasised that the two key services in this system were the Google File System, or GFS, which provided fault-tolerant, reliable, and scalable storage, and MapReduce, a data processing system that allowed work to be split among large numbers of servers and carried out in parallel. Olson (2010) also stated that the GFS and MapReduce were designed from the very beginning to run on the commodity server hardware that Google used throughout its data centre. Commodity server hardware is relatively inexpensive, readily available, all purpose, standardized and highly compatible hardware that is widely available and is easily interchangeable with other hardware of the same type.

In Apache Hadoop (2016) Hadoop was described as consisting of the following modules:

1. Hadoop Common: this is the common utility library that provides common functionality which is used to supports the other Hadoop modules.
2. Hadoop Distributed File System (HDFS): this is the distributed file system that provides high-throughput access to application data. HDFS stores data on the compute nodes, providing very high bandwidth across the cluster.
3. Hadoop YARN (Yet another Resource Negotiator): A framework for job scheduling and cluster resource management.
4. Hadoop MapReduce: A YARN-based system which performs parallel processing of large data sets. MapReduce is a computational paradigm, where the application is

divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster.

According to Stoneman (2017) Hadoop is a distributed technology, sharing work among many servers. Stoneman (2017) describes a Hadoop cluster as a classic master/worker architecture, in which the clients primarily make contact with the master, as shown in Figure 4.1. In a Hadoop cluster the master knows where the data is distributed between the worker nodes, and it also manages queries, splitting them into multiple tasks among the worker nodes. The worker nodes store the data and execute tasks sent to them by the master. When storing a large file in Hadoop, Stoneman (2017) states that the file gets split into many pieces, called blocks, and the blocks are shared between the workers in the cluster. Hadoop stores multiple copies of each block for redundancy at the storage level by default, three copies of each block are stored. When storing a 1 GB file on the cluster, Hadoop will split it into eight 128 MB blocks, and each of those eight blocks will be replicated on three of the nodes (Stoneman, 2017).

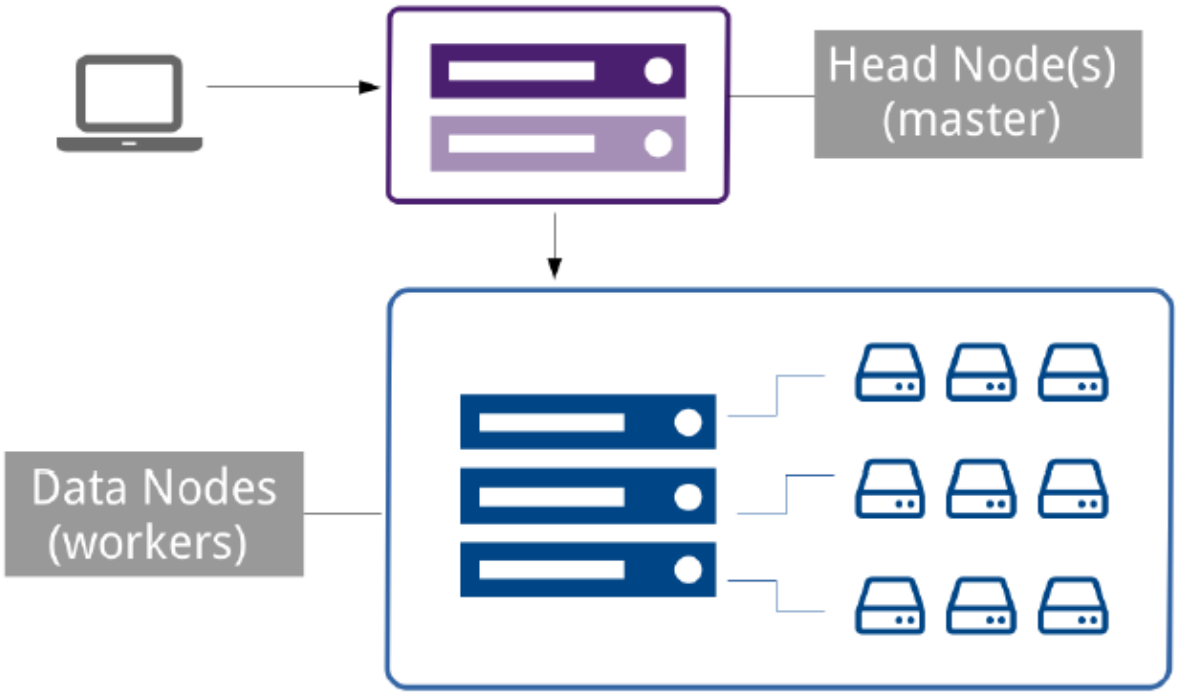

Figure 4.1: The Master/Worker Architecture of Hadoop (Stoneman, 2016)

Chaturvedi et al. (2015) have described the core Hadoop as consisting of Hadoop Distributed File System (HDFS) and MapReduce (MR) and that the whole system is known as the Hadoop Eco System. Some of the reasons organisations use Hadoop is mainly its ability to store, manage and analyse large amounts of both structured and unstructured data

quickly, reliably, flexibly and at low-cost (Hortonworks, 2018). Hortonworks (2018) listed the following as some of the advantages of using Hadoop:

1. Scalability and Performance – distributed processing of data local to each node in a cluster enables Hadoop to store, manage, process and analyse data at petabyte scale.
2. Reliability – large computing clusters are prone to failure of the individual nodes in the cluster. Hadoop is fundamentally resilient, that is when a node fails, processing is re-directed to the remaining nodes in the cluster and data is automatically re-replicated in preparation for future node failures.
3. Flexibility – unlike traditional relational database management systems, there is no need to have to create structured schemas before storing data. Data can be stored in any format, including semi-structured or unstructured formats. The data is then parsed and a schema is applied to the data when reading it.
4. Low Cost – as opposed to proprietary software, Hadoop is open source and runs on low-cost commodity hardware.

In the sections that follow the core components of Hadoop are discussed in more detail.

4.3 Hadoop Distributed File System (HDFS)

The HDFS is a distributed file system designed to run on commodity hardware which has many similarities with existing distributed file systems (Borthakur, 2013). In Borthakur (2013) the author points out that the differences of HDFS and other distributed file systems are significant because HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. Borthakur (2013) added that HDFS provides high throughput access to application data and is suitable for applications that have large data sets. HDFS was originally built as infrastructure for the Apache Nutch web search engine project and is now an Apache Hadoop subproject.

Ismael (2016) has described the HDFS as Hadoop's own rack-aware file system, which is a UNIX-based data storage layer of Hadoop. According to Ismael (2016) the HDFS is derived from concepts of Google filesystem and that an important characteristic of Hadoop is the partitioning of data and computation across thousands of hosts, and the execution of application computations in parallel, close to their data. Data files are replicated as sequences of blocks in the cluster on the HDFS. A Hadoop cluster scales computation

capacity, storage capacity, and I/O bandwidth by simply adding commodity servers and can be accessed from applications in many different ways (Ismael, 2016). Borthakur (2013) has also described HDFS as a distributed file system designed to run on commodity hardware and states that the HDFS has master/worker architecture, as shown on Figure 4.2.

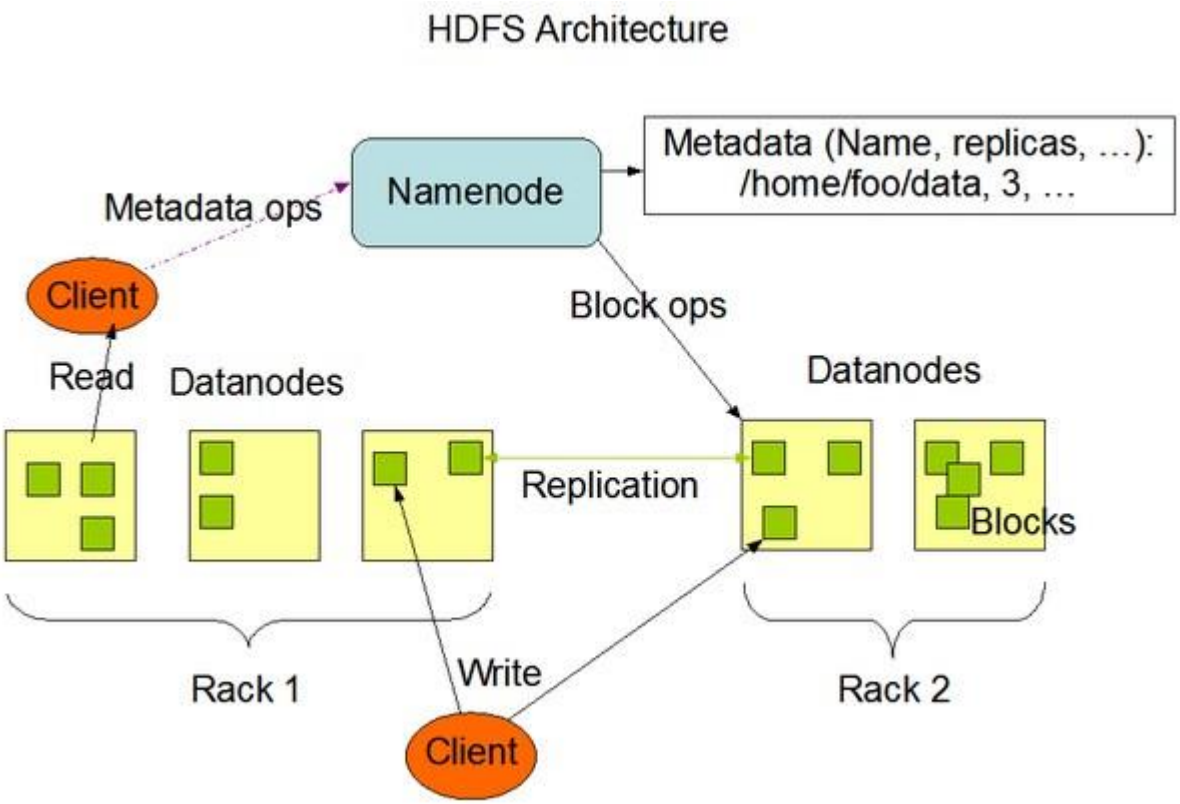

Figure 4.2: HDFS Architecture (Borthakur, 2013)

4.4 Apache MapReduce

Apache Hadoop implements a computational paradigm known as MapReduce, in which the application is split into several small fragments, each may be executed or re-executed on any node in the cluster (Agarwal, 2016). MapReduce is a computational model and a software framework for writing applications which are run on Hadoop and these programs are capable of processing massive amounts of data in parallel on large clusters of computational nodes (Pol, 2016). Pol (2016) has described MapReduce as a software framework for easily writing applications which process large amounts of data (i.e. multi-terabyte datasets) in parallel on large clusters (i.e. thousands of nodes) of commodity hardware in a fault-tolerant and reliable manner.

A MapReduce job typically splits the input dataset into independent chunks which are then processed by the map task in an isolated and parallel manor. The MapReduce framework sorts the output of the maps and then feeds this output as the input to the reduce tasks. Normally both the input and the output of the jobs are stored and saved in a file system. The MapReduce framework handles the scheduling of tasks; it also monitors them and re-executes any failed tasks. Basically the MapReduce framework operates exclusively on <key, value> pairs. It views the input to the job as a set of <key, value> pairs and generates a set of <key, value> pairs as the output of the job, which can possibly be of different types. The output does not necessarily need to be of the same type as the input record. The input pair may even map to zero or many output <key, value> pairs.

The MapReduce paradigm consists of two different tasks or phases that mainly deal with <key, value> pairs of data and these phases run sequentially in a cluster (Rodrigues et al., 2016). The output of the Map phase becomes input to the Reduce phase. These phases are explained below:

(1) The Map Phase – the mapper maps input <key, value> pairs to a set of intermediate <key, value> pairs. The map task captures the input and this input is divided into key value pairs

(2) The Reduce Phase – the reduce phase reduces a set of intermediate values which share a key to a smaller set of values. It maps input <key, value> pairs to a set of intermediate <key, value> pairs. The reduce phase consists of three primary sub-phases i.e. shuffle, sort and reduce.

The classic “Hello World” for Hadoop MapReduce is the famous “WordCount” program used to illustrate how MapReduce work. It is a simple example of an application that uses the MapReduce to count the number of occurrences of each word in a given input set. The only difference is that a real world implementation of MapReduce performs the steps on terabytes of data across thousands of computers in parallel. The WordCount program performs the following tasks:

(1) Read a file with words in it

(2) Determine what words are contained in the file

(3) Count how many times each word appears in the file and potentially rank or sort the results

The above WordCount process is implemented using the pseudo-algorithm in Figure 4.3:

```
map(String key, String value)
// key: document name
// value: document contents
 for each word w in value
   EmitIntermediate(w, "1")

reduce(String key, Iterator values):
// key: word
// values: a list of counts
  for each v in values:
    result += ParseInt(v);
    Emit(AsString(result));
```

Figure 4.3: MapReduce Algorithm

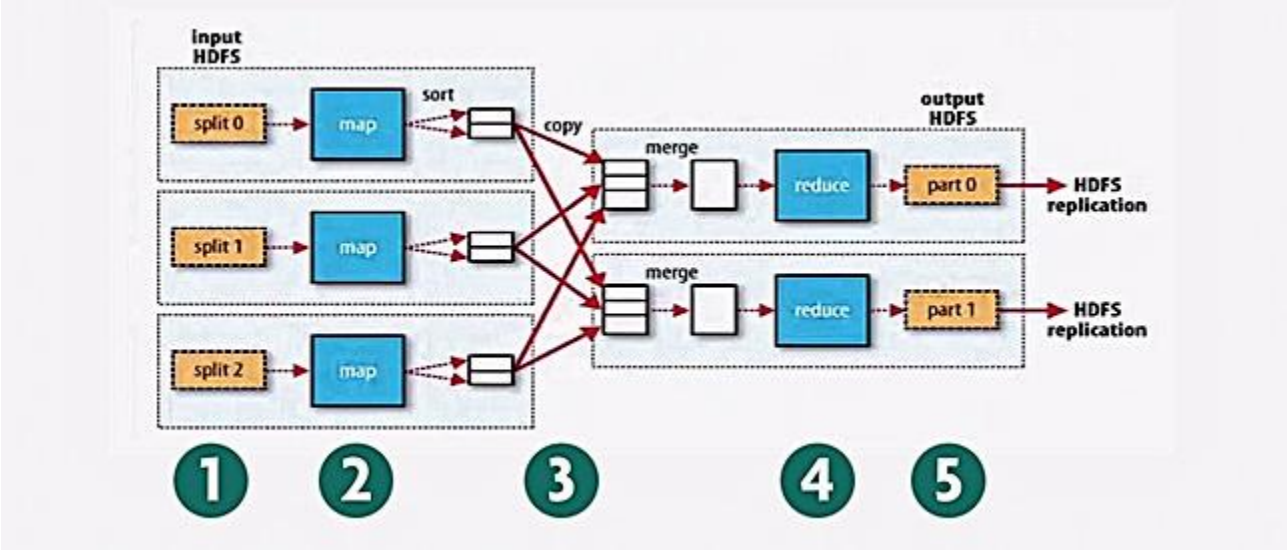

Figure 4.4: The Word Count Process

The MapReduce workflow for such a word count function would follow the steps as shown in Figure 4.4 as follows:

(1) The system takes input from a file system and splits it up across separate Map nodes

(2) The Map function or code is run and generates an output for each Map node in the word count function, every word is listed and grouped by word per node

(3) This output represents a set of intermediate key-value pairs that are moved to Reduce nodes as input

(4) The Reduce function or code is run and generates an output for each Reduce node in the word count example, the reduce function sums the number of times a group of words or keys occurs

(5) The system takes the outputs from each node to aggregate a final view

Figure 4.5 shows how the steps in Figure 4.4 will be applied to count a set of give words using the WordCount implementation.

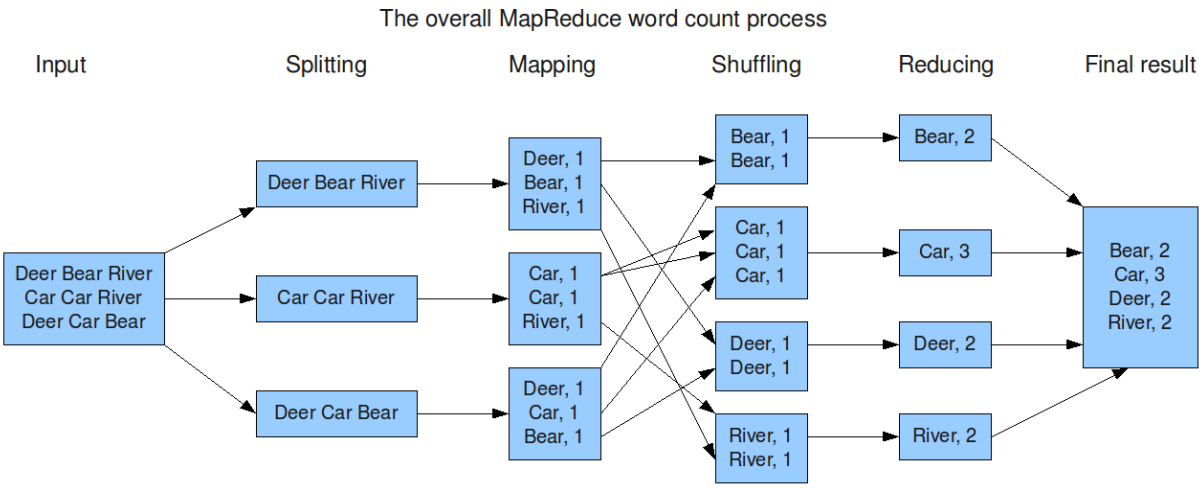

Figure 4.5: MapReduce Word Count Example

4.5 Apache Flume

The velocity or the speed of big data is one dimension which can be used to describe big data and can be thought of as the frequency of data generation or the rate of data delivery. This presents a challenge in the collection of this data in real time before we can make sense of the data or possibly take action. According to Russom (2011) the leading edge of big data is streaming data. In this study in order to solve this velocity challenge of social networks data, Apache Flume was utilised. Apache Flume can be described as a distributed, reliable and available tool or a service or a data ingestion mechanism for efficiently collecting, aggregating and moving large amounts of streaming data such as log files and events from various sources to a centralized data store such as HBase or HDFS (The Apache Software Foundation, 2019).

Flume is highly robust, reliable, configurable, and fault tolerant with tuneable reliability mechanisms and many failover and recovery mechanisms. It makes use of a simple extensible data model that facilitates online analytic application. The Flume data model is depicted in Figure 4.6.

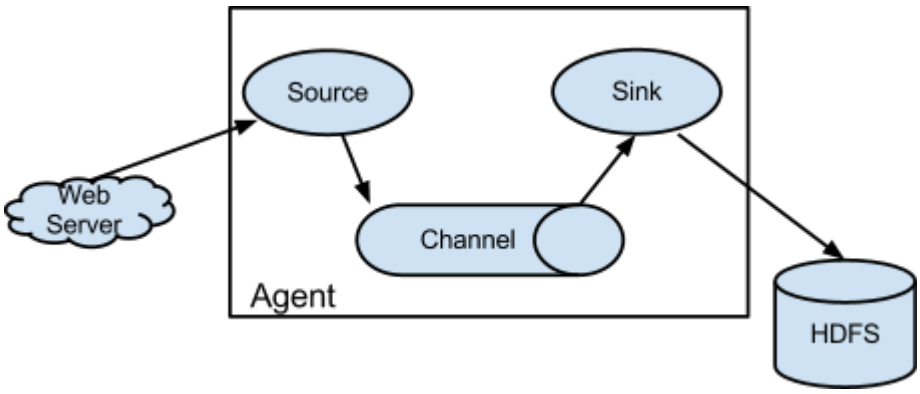

Figure 4.6: Apache Flume data model

An in depth discussion on how Apache Flume was setup to stream data from Twitter is provided in Chapter 5 and full detail of the complete installation are provided in Appendix B.

4.6 Summary

This chapter has provided background on the Apache Hadoop ecosystem. Details of how Apache Hadoop was download, installed and configured on a Windows 7 computer system are provides in Appendix A. In this study Twitter data was stored in the Hadoop File System for efficient storage. Most Hadoop systems are mainly installed and implemented on Ubuntu; this is mainly because Hadoop was built on top of Linux. The next chapter describes how Twitter data is streamed into HFS using Hadoop Flume and also how the data is accessed in the HFS for processing using MapReduce. The review of literature in this Chapter has concentrated largely on Apache Hadoop and the Hadoop Eco System which comprise of a number of Open Source Software, Libraries and Tools which can be utilized in this study for data collection, storage and analysis. Thus this chapter provides a basis for Chapter 5 which delves deeper and shows how data was stored in the HDFS for the purpose of this research.

CHAPTER 5: RESEARCH METHODS

User opinions found on blogs, product review sites, microblogs and social networks in general can provide a lot of insight which can enable an organisation to improve the quality of their products and to enhance services that they render to the customers. Twitter is a popular microblogging service that users use to create status messages known as “tweets” (Vinodhini & Chandrasekaran, 2012). These tweets in most cases express honest and unfiltered opinions about different topics, including products and services. Using Twitter enables users to answer the question: what is happening in the world and what people are talking about right now? That is, it gives a picture of what is currently trending in the world at the present moment. Users access Twitter via the web or via web enabled mobile devices. In this study sentiment analysis is performed on these opinions in order to understand what users are saying about a given topic.

This Chapter starts off with a discussion on the research paradigm which was used in this study in Section 5.1. Section 5.2 discusses how data was collected from Twitter using the Twitter API. Section 5.3 provides a discussion on Apache Flume and also provides information on how the data was streamed from Twitter and efficiently stored in the Hadoop File System (HDFS). Section 5.4 provides a discussion of this chapter and its relevance to the current study. Finally Section 5.5 provides a summary of the chapter in the conclusions.

5.1 Research Paradigm

The design science research paradigm was used for this research. Simon (1996) conceptualized design science research as a pragmatic research paradigm that calls for the creation of innovative artefacts to solve real-world problems. In other words, design science research combines a focus on the Information Technology artefact with a high priority on relevance in the application domain. Vaishnavi & Kuechler (2013) have described design science research as research that involves the creation of new knowledge through design of novel or innovative artefacts (things or processes that have or can have material existence) and analysis of the use and or performance of such artefacts along with reflection and abstraction to improve and understand the behaviour of aspects of Information Systems. Pfeffers et al. (2006) stated that there was a lack of a generally accepted process for design science in Information Systems. They decided to design a design science research process

model that would be consistent with prior literature, a model that would provide a nominal process for doing design science research and finally a model that would provide a mental model for presenting and appreciating design science in Information Systems (Pfeffers et al., 2006). The result of Pfeffers et al. (2006) synthesis is a process model consisting of six activities in a nominal sequence which are described and illustrated in Figure 5.1:

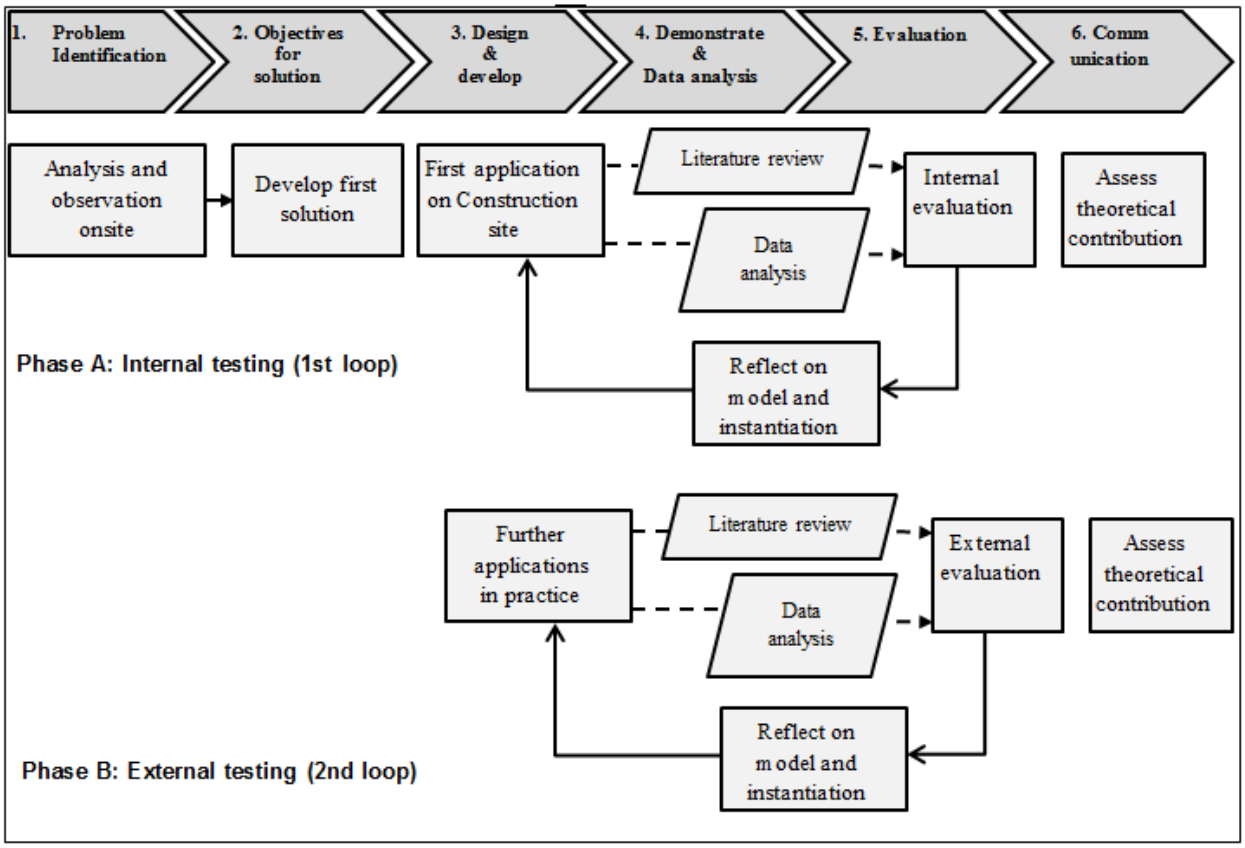

Figure 5.1: Design Science research process based on 6 steps as proposed by Pfeffers et al. (2006)

The six steps in the process model are described in (Pfeffers et al., 2006) as follows:

(1) Problem identification and motivation – this activity defines the specific research problem and justifies the value of a proposed solution.

(2) Objectives of a solution – in this activity the problem definition is used to deduce or infer the objectives of a solution. The inferred objectives can either be quantitative e.g. terms in which a desirable solution would excel more than the current one or qualitative e.g. where a new artefact is expected to provide solutions to problems not yet addressed in the current system.

(3) Design and development – the main activities in this step is to determine the artefact's desired functionality and its architecture and then creating the artefactual solution.

(4) Demonstration – in this step a demonstration of the effectiveness of the artefact to solve the problem is provided. Generally this could involve the use of the artefact in experimentations, simulations, a case study, proof or other appropriate activities to measure its effectiveness and efficiency in providing the solution.

(5) Evaluation – this step provides an evaluation of the artefact by observing and measuring how well the artefact supports a solution to the problem. At this point a comparison is done of the objectives of a solution to actual observed results from the use of the artefact as shown in the demonstration (step 4 above). According to Pfeffers et al. (2006) at the end of this step the researchers could decide whether to iterate back to step 3 to try and improve the effectiveness of the artefact or to continue on to the next step and defer further improvement to subsequent projects and also the feasibility such an iteration plays a crucial role in determining the way forward. A set of evaluation criteria by (March & Smith, 1995) are given in Table 5.1 below. In this dissertation the design science output as described in Table 5.1 was Instantiation. This is because the artefacts were created and their efficiency and effectiveness was measured and how well they worked.

(6) Communication – in this step communication is provided for the artefact, the problem and its importance, its design, its utility and uniqueness, and its effectiveness to the researchers involved and other relevant stakeholders such as practising professionals when appropriate. Pfeffers et al. (2006) suggests that in the case of scholarly research publications researchers might opt to use the structure of this process to structure the paper itself, just like the nominal structure of an empirical research process (i.e. problem definition, literature review, hypothesis development, data collection, analysis, results, discussion and conclusion) is a common structure utilised for empirical research papers.

The design science research was appropriate for this research given that the five steps described in (Pfeffers et al., 2006) were all followed in conducting the experiments for this study. Several artefacts were designed and developed; their efficacy to solve the problem was demonstrated and evaluations of their effectiveness were conducted. Several iterations to improve the artefacts in terms of accuracy were conducted and finally a scholarly research publication (Nhlabano & Lutu, 2018) was communicated.

Table 5.1: Evaluation criteria for outcomes of Design Science (March & Smith, 1995)

DESIGN SCIENCE OUTPUT	DEFINITION	IMPORTANT EVALUATION CRITERIA (March & Smith, 1995)
Construct	A conceptualization used to describe problems and specify solutions for example it can be formal (data modelling) or informal (cooperative work).	Completeness, simplicity, elegance and ease of use.
Model	A set of propositions or statements expressing relationships among constructs. A description of how things are.	Fidelity with real world phenomena, completeness, level of detail, robustness and internal consistency.
Method	A set of steps used to perform a task. They are based on a set of underlying constructs (language) and parts of the model as input.	Operationality (the ability to perform the intended task or the ability of humans to effectively use the method if it is not algorithmic), efficiency, generality and ease of use.
Instantiation	Is the realization of the artefact in its environment?	Does it work and how well does it work? Why did it work and why did it not work? Efficiency and effectiveness of the artefact and its impact on its environment and its users.

5.2 Data for Text Pre-processing Experiments

In the experimental study of the impact of text pre-processing on the performance of sentiment analysis models for social media data, the movie review corpus v2.0 was chosen as the input data for conducting the experiments. The movie review corpus is a collection of user movie reviews from the Internet movie database which was compiled by Bo Pang at Cornell University. This data set can be found and downloaded at the following link: http://www.cs.cornell.edu/people/pabo/movie-review-data/. Several studies have been

conducted using the movies review data set. A list of 100 papers using the movie review data set is listed in chronological order on the following website: https://www.cs.cornell.edu/people/pabo/movie-review-data/otherexperiments.html.

The corpus consists of 2000 rated movie reviews, comprising of an equal number of reviews for each sentiment group (i.e. 1000 positive and 1000 negative) stored in individual files (i.e. 1 text file per review) which was first introduced in Pang & Lee (2004). The categories are exclusive; making a classifier trained on them a binary classifier. Binary classifiers or binomial classification have only two classification labels, and will choose one or the other.

5.3 Twitter Data Collection

In order to share information, Twitter provides companies, developer and users with programmatic access to Twitter data through their Application Programming Interfaces (APIs). These APIs allow people to build software that integrates with Twitter and provides a platform to access public Twitter data that users have chosen to share with the world. Twitter also supports APIs that allow users to manage their own non-public Twitter data (such as Direct Messages). Such data is only provided to developers whom the users have authorized to do so.

The Twitter APIs include a wide range of endpoints. An endpoint is an address that corresponds with a specific type of information. Twitter endpoints fall into five primary groups:

(1) Accounts and users – this endpoint is used to manage an account's profile and settings, mute or block another user, request information about an authorized account's activity, manage users and followers etc.

(2) Tweets and replies – this endpoint provides access to public Tweets and replies and it also allows software developers to post Tweets. Through this endpoint software developers can access Tweets by searching for specific keywords or by requesting a sample of Tweets from specific accounts. In this research the Tweets and replies end point was utilised to collect data from Twitter.

(3) Direct Messages (DM) – this endpoint provides access to the DM conversations of users who have explicitly granted permission to a specific application.

(4) Ads – this suite of APIs enables software developers to help organisations automatically create and manage advertisement campaigns on Twitter.
(5) Publisher tools and Software Development Kits (SDKs) – these tools enable software developers and publishers to embed Twitter timelines, share buttons and other Twitter content on webpages.

To stream data from Twitter, the following steps were followed:

(1) An application was created and registered on Twitter on the following link: https://apps.twitter.com/ (as shown in Appendix C)
(2) Apache Hadoop was installed (as shown in Appendix A)
(3) Apache Flume was installed and configured (as shown in Appendix B)

Figure 5.2 depicts how Apache Flume can be used to fetch data from various services and transport it to centralized stores such as HDFS and HBase.

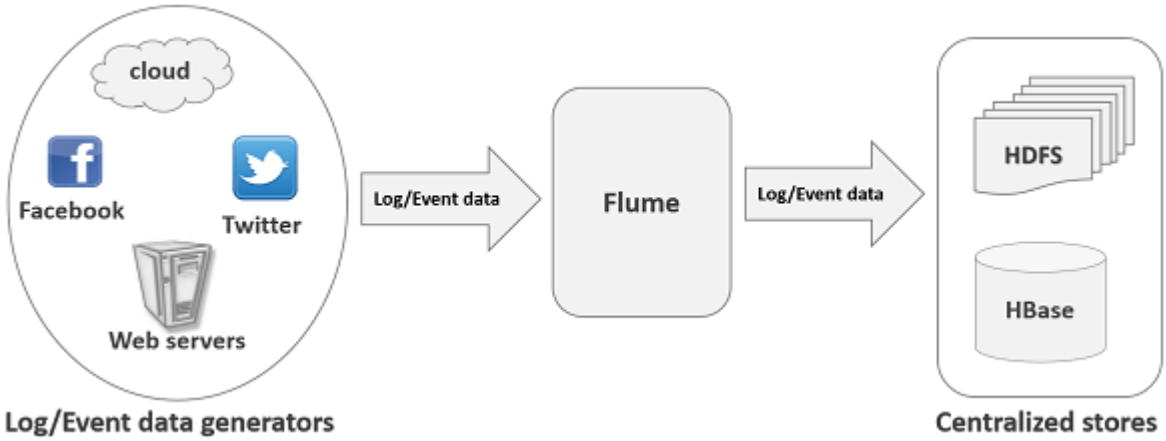

Figure 5.2: Log or Event data generators

5.4 Streaming Twitter Data into Hadoop Using Apache Flume

In this study, one of the analysis tasks was to perform sentiment analysis on Twitter data. Twitter is a social networking site which was described in more details in Chapter 2, Section 2.5. The data which was used in this study was generated from a Twitter data source. A source (e.g. a Twitter source) is a component of a Flume Agent that receives data from data generators and transfers this data to one or more channels in the form of Flume events. This data will be in the form of log files and events. A log file can be described as a file that lists

events or actions that can happen in an operating system e.g. a request made to the server can be one of the entries written to the log file. An event or a Flume event represents the basic unit of the data transported inside Flume. The channel buffers this data to a sink, which then pushes it to the HDFS. A channel provides a bridge between the sources and the sinks, it is essentially a transient store which receives the events from the source and buffers then until these events are consumed by a sink. The purpose of a sink is to consume events (i.e. the data) from the channels and send it to its destination (i.e. a centralized store such as HBase or HDFS or even another agent).

In order to analyse Twitter data, the data has to be moved and stored in a central store efficiently (in this case the HDFS). This is where Apache Flume comes to the rescue. As shown in Figure 5.3, Flume can be used to move the log data generated by an application server into the HDFS at a very high speed. In this study Flume was utilized to move data from Twitter into the HDFS as show in Figure 5.3. There are several types of channels for example, Java Database Connectivity (JDBC) channel, File system channel or a Memory channel. For this study a memory channel was utilized as shown in Figure 5.3.

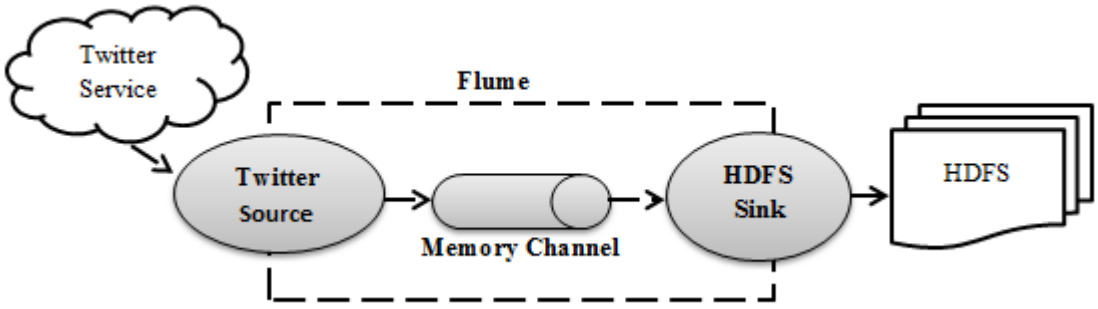

Figure 5.3: Fetching Twitter data using Flume

In order to filter tweets about the movie Avengers Endgame, the Twitter source had to be configured so that it can only filter and download tweets that contain the keywords Avengers Endgame. This was achieved by setting the Twitter agent's keywords property shown in the extract of the Flume configuration file below:

```
TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = dTDmMrN5XNmBynOIQzyZj1mmn
TwitterAgent.sources.Twitter.consumerSecret = UyQ6WHd7ARlVig9SdvBsDTntZLXWxHdOFW8O0QLc4MKp0OEU38
```

TwitterAgent.sources.Twitter.accessToken = 2493822120-T1EBfSWIzBvCstWLoAF78zbe6y1nsUTcRfit8Ss
TwitterAgent.sources.Twitter.accessTokenSecret=VNUomCbmthjpdthpoI2BVU43AQXCN0Rd8NZDNrQSh8pfL
TwitterAgent.sources.Twitter.keywords = Avengers Endgame

The Twitter agent can be configured to download data regarding any topic by changing the keywords to the desired topic. The point of this study was to generate data regarding the reviews of the movie Avengers Endgame so that sentiment analysis can be performed in order to determine whether the movie goers thought the movie was very good, good, neutral, bad or very bad. Sentiment analysis on this data is reported in the next chapter.

5.5 Conclusion

In this study Twitter data regarding the movie Avengers Endgame was collected from Twitter and this data was stored in HDFS. A Twitter application was registered and created on Twitter as described in more details in Appendix C. Apache Flume was used to stream these tweets from Twitter into the HDFS via a memory channel. Details of how Apache was downloaded, installed and configured are provided in Appendix B. This chapter provides a comprehensive description of the technology which was utilized and the steps which were followed to setup and configure this technology. To stream data from Twitter, a Twitter application was created and Apache Flume was used to provide a memory channel to sink this data into the HDFS. The data stored in the HDFS was used as the input to the sentiment analysis experiments which are discussed in Chapter 6.

CHAPTER 6: EXPERIMENTS FOR TEXT PRE-PROCESSING

Online text found on user reviews and public opinions found in blogs, discussion forums and social networks, such as tweets are typically short, incomplete, noisy, usually informal and full of different orthographical and grammatical mistakes, misspellings, typo-graphic errors, and uninformative parts such as HTML tags, scripts and advertisements (Jianqiang & Xiaolin, 2017). Twitter data which is used in this study is an example of online text which requires cleaning before it can be analysed accurately. The challenges inherent to online text can be mitigated by implementing a series of text pre-processing techniques.

In this dissertation two types of experiments were conducted. The first experiment was to study the impact of text pre-processing on the performance of sentiment analysis models for social media data. In the experiment the data from the movie review corpus was run through a series of text pre-processing methods before sentiment analysis was performed on the data and the accuracy was measured before and after applying the text pre-processing methods. The second experiment which was conducted was on Apache Flume, HDFS, MapReduce and the AFINN-111 lexicon which is discussed in Chapter 7.

This Chapter is organized as follows: Section 6.1 discusses the objectives of the text pre-processing experiments. In Section 6.2 a discussion of the background on the study of text pre-processing is provided. A discussion on the data and algorithms used for the text pre-processing experiments is provided in Section 6.3. The details of the text pre-processing methods which are implemented in this study are discussed in section 6.4. The results of the text pre-processing experiments are presented in section 6.5 and finally an in-depth discussion on the experimental results is provided in Section 6.6.

6.1 Objectives of the Text Pre-processing Experiments

The objective of this experiment was to study the impact of text pre-processing on the performance of sentiment analysis models for social media data. The experimental results demonstrate that Text pre-processing methods increases the predictive accuracy of the resulting models for sentiment classification as was demonstrated in the experiments conducted in this study (Nhlabano & Lutu, 2018). Figure 6.1 show the overall proposed workflow utilized in this experiment.

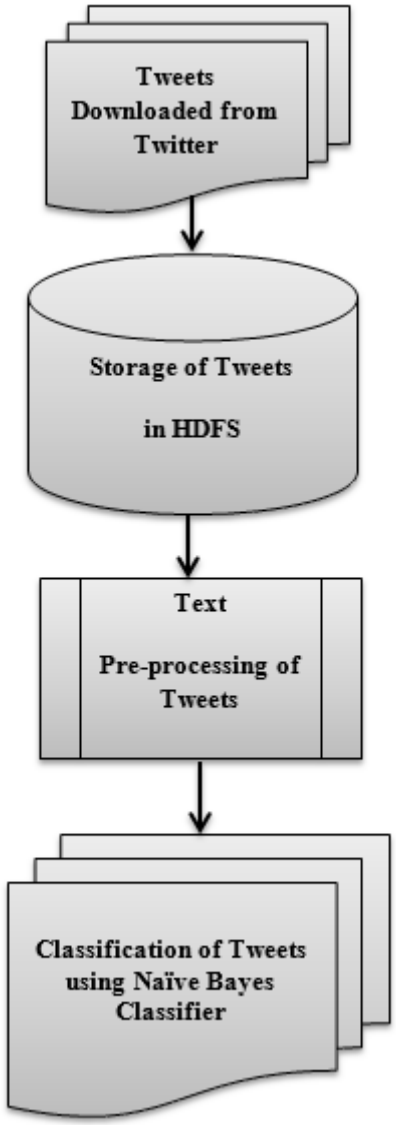

Figure 6.1: Flow diagram for the text pre-processing experiments

In order to evaluate the impact of the various pre-processing methods exploratory experiments were conducted to answer the following questions:

(1) What impact do commonly used pre-processing methods for Social media data have on the predictive accuracy of the resulting models for sentiment classification?

(2) Do applying pre-processing methods such as stop words removal and stemming improve the accuracy and or performance of the classifiers, such as the Naive Bayes Classifier?

(3) Does reducing dimensionality through feature selection improve classifier performance?

6.2 Study of Text Pre-processing methods on the Performance of Sentiment Analysis Models for Social Media Data

The purpose of sentiment analysis (discussed in more details in Chapter 3) is to predict the polarity of a given text item in order to determine if the author is expressing positive, negative or a neutral opinion about a given topic. Social media sites generate large volumes of sentiment rich textual data that is inherently unstructured. Organisations without automated systems rely on human readers to monitor the data from social media. The difficulty with such a method is that an average human reader will experience challenges selecting appropriate sites and accurately summarizing the data and opinions found on these social media sites. Human analysis of text information can lead to bias, mental and physical limitations. This may cause inconsistent results as the volume of data to be processed increases. It is clear from these limitations that automated opinion mining and summarization systems are required in order to increase consistency and accuracy in the analysis of data. An objective sentiment analysis system reduces the subjective human biases and mental limitations (Liu & Zhang, 2012).

Haddi et al. (2013), stated that there exist three main approaches to sentiment analysis i.e. Machine learning methods, Lexicon based methods and Linguistic analysis methods. Machine learning is the most widely adopted approach towards sentiment classification. Classification algorithms such as Naive Bayes, Maximum Entropy and Support Vector Machines (SVM) are used to create predictive classification models. The experiments reported in this chapter utilize the Naïve Bayes algorithm. In developing an automated sentiment analysis system, one of the most important steps is text pre-processing.

In this dissertation one of the tasks was to investigate the influence of text pre-processing methods used for social media data. Haddi et al. (2013) have defined text data pre-processing as the process of cleaning and preparing data for classification. According to García et al. (2012) this poor text quality causes a lot of noise and also causes tools designed to analyse formal text to suffer a critical performance and accuracy decrease. Haddi et al. (2013) have observed that text pre-processing reduces noise in the text and it also helps in improving the performance of the classifier. This consequently speeds up the classification process, thus aiding in real time sentiment analysis. The difficulties caused by

noise in the text affects the robustness of the analysis and increase the computational complexity of the classification process.

6.3 Data set and algorithms for the experiments

In this study the movie review corpus v2.0 was utilised when conducting the experiments. This data set is described in more details in Chapter 5. Before removing stop-words or applying stemming there were a total of 1,293,948 word and 48,813 unique words in the dictionary. Each file was used as a single instance for both training and also for testing the Naive Bayes Classifier (NB) classifier, which was discussed in more details in Chapter 3. For training the classifier 750 positive reviews and 750 negative reviews were used and the remaining 500 reviews (250 positive and 250 negative) were used for testing the classifier.

The algorithms used in designing a solution for the experiments, were created using Python 2.7 (Python, 2017) and Natural Language Tool Kit 3.2.5 (NTKL) (Bird & Loper, 2004). Python is a widely adopted high-level programming language developed under an OSI-approved open source license which makes it freely usable and distributable and this also applies for commercial use as well and is available on (Python, 2017). NLTK is a collection of open source program modules, tutorials and problem sets that provide packages in the toolkit are updated regularly. NLTK is written in Python and the reason for using NLTK and Python in this research is that they are both distributed under the GPL open source license available freely. The other reason is that they are fast and efficient when the dataset is big in terms of performing classification and feature extraction. Even though the movie review corpus was used, the machine learning methods and features used in this study are not specific to movie reviews, and can easily be adapted to other domains as long as sufficient training data exists.

6.4 Text Pre-processing Methods

The following text pre-processing methods were used to clean the text before sentiment scoring:

6.4.1 Stop word removal

Rajaraman & Ullman (2011) described stop words as the most frequent common words such as “the” or “and” which help construct ideas, but do not carry any significance themselves. In general stop words are high-frequency words with little lexical content and their removal leads to a reduction in the dimensionality of the term. For the text pre-processing experiments the English stop words list from the Natural Language Toolkit 3.2.5 (NLTK) was utilized. NLTK is a collection of open source program modules. The NLTK 3.2.5 comes with a stop words corpus list which has 2,400 stop words for 11 languages and a list of 128 English stop words (Bird et al., 2007). Removing the stops words from the movies review corpus reduced the number of features from 1,583,820 to 955,610 which means 60% of the corpus were actual features and 40% were stop words. This is a massive reduction which certainly has a huge impact on the size, efficiency and accuracy of the classifier.

6.4.2 Removal of URLs and @Username

In this study, there was no interest in following the web links as that would not contribute anything towards the sentiment analysis efforts and also URLs do not carry much information regarding the sentiment of the tweet, so the URLs where removed from the given tweet before scoring. The URLs were matched and removed from the tweets in order to refine the tweet content.

Usernames in Twitter typically starts with the “at” sign (i.e. @) before the name, this is how users refer to one another on Twitter. The reason for removing the @Username from the tweet is that it has no significant in calculating the sentiment score of the tweet. Since the username will not appear in the AFINN-111 lexicon, this means its entry will not be found in the file, so it’s presence in the tweet will only increase the system’s computational complexity unnecessarily.

6.4.3 Stemming using the Porter Stemmer

Stemming is a text pre-processing method utilized mainly in Information Retrieval systems, Text mining as well as Natural language processing applications to reduce a word to its root/stem (Vijayarani et al., 2015). The purpose of stemming is to reduce the number of words by removing various suffixes of the words in text. The goal is also to reduce inflectional forms and derivationally related forms of a word to a common base or root form (Jivani,

2011). According to Vijayarani et al. (2015), stemming is done in order to remove various suffixes, in order to have accurately matching stems and this helps to reduce the number of words and saves time and memory space. Figure 6.2 is an example of the stemming process. Given the various forms of the word 'presentation', 'presenting', 'presents', and 'presented', it can be reduced to the word, present by the stemming process.

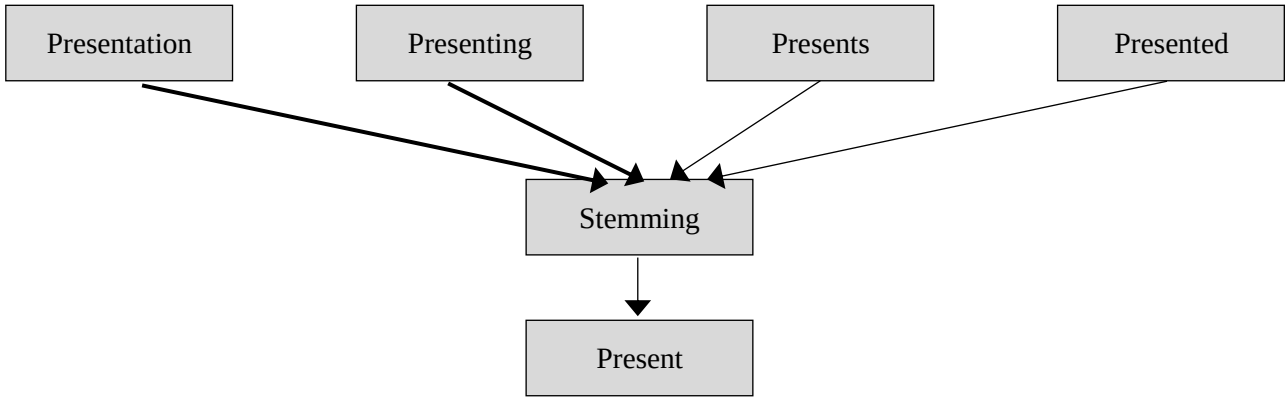

Figure 6.2: Example of the stemming process

Stemming algorithms can be broadly classified into three groups (Jivani, 2011). The three categories are truncating methods, statistical methods and mixed methods as shown in Figure 6.3.

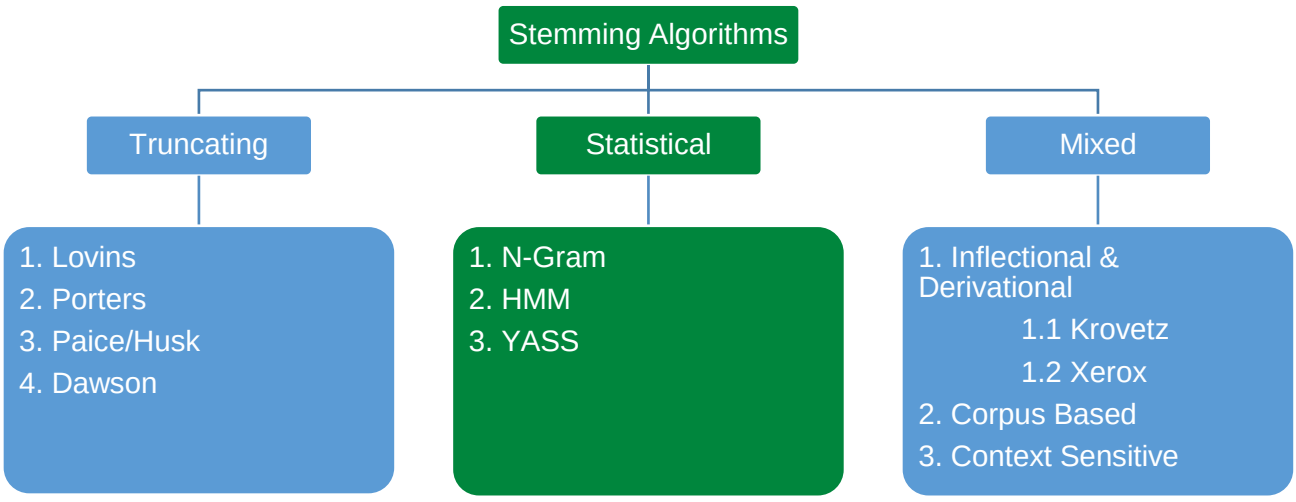

Figure 6.3: Categories of stemming algorithms

In this study a snowball implementation of the Porter Stemmer (Porter, 1980) was used to stem the words in a given tweet or in a movie review. Snowball is a small string processing

language which was designed specifically for creating stemming algorithms for use in Information Retrieval (IR). The complete Porter stemmer algorithm is given in Appendix G.

6.4.4 Feature Selection

In machine learning, a predictive classification task consists of several predictor variables and one predicted variable. In text mining, the predictor variables are all the words that can appear in the text. Specific to sentiment analysis, the predicted variable is the sentiment, which can take on the values: positive, negative, or neutral. Predictor variables are also known as features, and the number of features is called the dimensionality of the classification task.

The high dimensionality of Social media text data poses a severe challenge to feature selection and feature extraction methods with respect to their efficiency and effectiveness. In this study dimensionality was reduced through feature selection as a pre-processing step in order to remove irrelevant and redundant data, thereby increasing learning accuracy of the classifier, as explained in Perkins (2014). In general feature selection methods choose features from the original set based on some criteria such as, correlation, information gain, and mutual information to filter out irrelevant and redundant features. According to Cunningham (2008), the objective of feature extraction is to identify features that are correlated with or predictive of the class label. In this study the objective was to select features that will construct the most accurate classifier and to transform a list of words into a feature set that is usable by the classifier. The benefits of this step are not only for reasons of computational efficiency but also to improve the accuracy of the classifier. This goal was achieved by eliminating low information features and utilizing high information features. According to Perkins (2014), low information features refer to words that are common to all labels. In this study these are features that are common across all classes and therefore contribute little information to the classification process. In the movie review corpus, these are words which are common in both positive and negative reviews and these features can decrease performance.

Removing these low information features gives the model clarity by eliminating noisy data and prevents the curse of dimensionality which is a phenomenon that arise when analysing and organizing data in high-dimensional spaces. The problem with the curse of

dimensionality is that when the dimensionality increases so does the volume of the space. Consequently the available data becomes sparse which creates problems for any method that requires statistical significance because the amount of data required supporting a statistically sound and reliable result often grows exponentially with the dimensionality.

Using Higher information features, which refers to features that are strongly biased towards a single classification label (Perkins, 2014), can increase performance and at the same time reduce the size of the model. This results in less memory usage along with faster training and classification of the model. To get the high information features from the movie review corpus the information gain was calculated for each word in the corpus. Information gain measures the number of bits of information acquired for category prediction by knowing the absence or presence of a term in a document (Yang & Pedersen, 1997a). For text classification as in this case information gain is a measure of how common a feature is in a particular class relative to how common it is in all other classes, for example for a given word that occurs primarily in positive movie reviews and rarely in negative movie reviews is a high information feature.

The Chi-square (X^2) is one of the best metrics for Information gain. To calculate information gain in this study the Chi-squire implementation included in the metrics package in the NLTK was utilized. The Chi-square in the NLTK is the Phi-square multiplied by the number of features, as in Manning & Schütze (1999). The Phi-square which is also referred to as the "Mean square contingency coefficient" denoted by ∅ is the square of the Pearson correlation coefficient, is a measure of the amount of the association between two binary variables (i.e. living/dead, black/white, success/failure and in the case of this study positive/negative). It is based on Chi-square coefficient which depends upon the strength of sample size and relationship and is denoted by the equation below:

$$\emptyset = \sqrt{\frac{X^2}{n}} \tag{6.1}$$

As can be observed in Equation 6.1 above the ∅ eliminates sample size n by dividing Chi-square by n and computing its square root. This is the property of the Phi-square that makes it applicable to our goal to reduce dimensionality. The Phi-coefficient square for a 2

by 2 contingency table relates the square of Phi-coefficient with the Chi-squared coefficient and is represented by the equation below:

$$\emptyset = \frac{a.d - b.c}{\sqrt{e.f.g.h}} \tag{6.2}$$

To calculate the Chi-square the NLTK implementation multiplies the Phi-square with the number of features as indicated in the Equation 6.3 below:

$$X^2 = \frac{N(O_{11}\ O_{22} - O_{12}\ O_{21})^2}{(O_{11} + O_{11})(O_{11} + O_{21})(O_{12+}\ O_{22})(O_{21} + O_{22})} \tag{6.3}$$

where N represents the total words count in the corpus, O_{11} represents the positive word frequency, O_{22} represents the total negative word count, O_{12} represents the total positive word count, and O_{21} represents the negative word frequency. The parameters defined in Equation 6.3 vary depending on whether the calculation is for the positive or negative scores. To utilize the X² to score each feature firstly two frequencies need to be calculated which are used as input to the Chi-square: (1) Frequency Distribution for overall frequency of words, and (2) Conditional Frequency Distribution where the conditions are the class labels. The two frequencies are calculated twice for each class (i.e. for both positive and negative score). For example, scoring the word 'Magnificent' which is the top most informative feature in the Movie reviews corpus is achieved by utilizing Equation 6.3 above and calculating the positive and negative scores and adding them together. The positive score is calculated as shown in Equation 6.4 as follows:

$$\boldsymbol{Positive} = \frac{n\_xx(n\_ii\ n\_oo - n\_io\ n\_oi)^2}{(n\_ii + n\_io)(n\_ii + n\_oi)(n\_io + n\_oo)(n\_oi + n\_oo)} \tag{6.4}$$

$$\frac{1583820(33\ * 751252 - 832531 * 4)^2}{(33 + 832531)(33 + 4)(832531 + 751252)(4 + 751252)}$$

=19.90263231804343

where *n_xx* represents the total word count in the corpus, *n_ii* represents the positive word frequency, *n_oo* represents the total negative word count, *n_io* represents the total positive word count, and *n_oi* represents the negative word frequency. The Negative score is calculated as shown in Equation 6.5:

$$\boldsymbol{Negative} = \frac{n\_xx(n\_ii\ n\_oo - n\_io\ n\_oi)^2}{(n\_ii + n\_io)(n\_ii + n\_oi)(n\_io + n\_oo)(n\_oi + n\_oo)} \quad (6.5)$$

$$\frac{1583820(4 * 832531 - 751252 * 33)^2}{(4 + 751252)(4 + 33)(751252 + 832531)(33 + 832531)}$$

= 19.90263231804343

where *n_xx* represents the total word count in the corpus, *n_ii* represents the negative word frequency, *n_oo* represents the total positive word count, *n_io* represents the total negative word count, and *n_oi* represents the positive word frequency. Lastly, its positive and negative scores are added together as shown below:

$$\boldsymbol{Word\ Score} = \textbf{19.90263231804343 + 19.90263231804343}$$

=39.80526463608686

This exercise is repeated for each feature in the corpus. Once all the scores are calculated, then the words are sorted by score. Finally the top 10 000 features are taken and are put in a set so that each file in the movie review corpus is classified based on the presence of these high information words.

6.5 Experimental Results for Text Pre-processing

This section reports the results of 10 experiments conducted to assess the accuracy of the Naïve Bayes Classifier before and after applying the pre-processing methods described in Section 6.5. Feature selection had perhaps the largest influence on the accuracy of the classifier. Before applying pre-processing an accuracy of 0.728 was recorded. After applying the feature selection method, an accuracy of 0.93 was recorded. This shows a massive improvement of about 0.202, which is more than 20% from the initial test without pre-processing. Ten samples of 50 instances (i.e. 25 positive and 25 negative) were taken for each test set from the movie review corpus. The results from the experiments conducted

are shown in Table 6.1 and their respective percentage accuracy. To calculate the accuracy of the classifier the accuracy method defined in the utility class defined in the Natural Language Tool Kit (NLTK) was utilised. In this method accuracy is defined as:

$$\text{Accuracy} = \frac{Count\ of\ correct\ predictions}{Count\ of\ tested\ instances} \tag{6.6}$$

In the results shown in Table 6.1 the first column shows the ID of each test set, the second column stores the percentage accuracy recorded without Pre-processing and the third column shows percentage accuracy after pre-processing was applied to the data set.

Table 6.1: Text Pre-processing Experiment Results

Test ID	**Percentage Accuracy**	
	No Pre-processing (NP)	***With Pre-processing (WP)***
T1	78	94
T2	70	94
T3	74	96
T4	74	90
T5	62	88
T6	78	88
T7	68	98
T8	72	92
T9	76	94
T10	76	98
Mean	**72.8**	**93.2**

To make sure that the observable differences for the result in Table 6.1 were not explained by some random variation and to demonstrate that they were not due to chance, the Paired Sample t-test was used. This is a statistical procedure used to determine whether the mean difference between two sets of observations is equal to zero. For the test that was conducted, the mean difference to be tested was:

$$\mu_d = \mu_{WP} - \mu_{NP} \ . \tag{6.7}$$

The following two hypotheses on the mean difference were used:

$$H_0 : \mu_d = 0 \tag{6.8}$$

This is the null hypothesis which states that there is no difference in accuracy and,

$$H_1 : \mu_d \neq 0. \tag{6.9}$$

This is the two tailed alternative hypothesis which states that there is an increase in accuracy. To perform the t-test, the Data Analysis add-in tool found in Microsoft Excel (2013) was utilised. The results from this process are show in Table 6.2:

Table 6.2: T-test Results

t-Test: Paired Two Sample for Means		
	No Pre-processing	***With Pre-processing***
Mean	72.8	93.2
Variance	25.06666667	13.51111111
Observations	10	10
Hypothesized Mean Difference	0	
df	9	
Alpha level	0.05	
t Stat	-11.27930965	
P(T<=t) two-tail	1.30266E-06	
t Critical two-tail	2.262157163	

In Table 6.2 if it is the case that (t Stat < -t Critical two-tail) or (t Stat > t Critical two-tail), the null hypothesis is rejected. In this case (t Stat < -2.262), therefore the null hypothesis was rejected. That means the researcher is 95% (i.e. alpha level 0.05) confident that the observed difference between the sample means 20.4% (i.e. 93.2 - 72.8) is convincing enough to say that the average accuracy without pre-processing and with pre-processing differ significantly. Therefore it can be concluded that the text pre-processing methods discussed in section 6.5 of this dissertation does improve the accuracy of the Naïve Bayes classifier.

6.6 Discussion of Experimental Results for Text Pre-processing

The text pre-processing experiments results contributed the following two benefits: (1) improving the predictive accuracy of sentiment classification models by applying various text pre-processing methods described in the literature and, (2) increasing sentiment classification model performance by reducing the dimensionality of the classification model through feature selection. The methods used for the experiments in this section were also applied in the next experiment to clean Twitter data before analysis using a lexicon based sentiment analysis approach as discussed in Chapter 7.

6.7 Conclusion

In this Chapter the experiments conducted was to study the impact of text pre-processing on the performance of sentiment analysis models for social media data. In these experiments different text pre-processing methods for Social media data were presented and applied to the movie review corpus. The data was then used to train the Naïve Bayes Sentiment classifier. Several experiments were conducted with the trained Naïve Bayes classifier for accuracy in order to evaluate and demonstrate the predictive accuracy of the resulting model for sentiment classification. The results indicate that improving feature selection will improve the classifier accuracy and increase performance of the classifier and also decreases the size of the model, which results in less memory usage as well as faster training and classification. This means that the learning accuracy of the classifier improves when the proposed pre-processing methods discussed in this study are applied. Reducing dimensionality through feature selection is one of the most effective pre-processing methods that can be used to improve classifier performance along with stop word removal and stemming.

CHAPTER 7: EXPERIMENTS FOR SENTIMENT ANALYSIS

The real time data found on social networks such as Twitter are complex and they possess the 3V's of big data (Kumari, 2016). This means Twitter data has high volume, velocity and variety which require specific technology and analytical methods for it to be transformed into valuable information for an organisation (De Mauro et al., 2016). The challenge of high volume data that arrives at high speeds in different variety of formats is how to handle the speed with which new data is created or existing data is updated while the system is required to process and make sense of the data immediately upon its creation (Chen et al., 2013).

The challenges described above can be addressed by making use of Apache Flume, Apache Hadoop and MapReduce. Apache Flume is a distributed, reliable, and available system which is used to efficiently collect, aggregate and move large amounts of log data from many different sources to a centralized data store such as HDFS. Apache Hadoop is an open source software library that runs on low-cost commodity hardware and has the ability to store, manage and analyse large amounts of both structured and unstructured data quickly, reliably, and flexibly at low-cost. MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

This chapter is organized as follows: Section 7.1 discusses the objectives of the experiments conducted in this chapter. Section 7.2 discusses the setup of the experiments conducted in this chapter, followed by Section 7.3 which discusses sentiment analysis using the AFINN lexicon. A string manipulation process called Tokenization is discussed in Section 7.4. Section 7.5 of this chapter discusses the sentiment analysis process using the MapReduce paradigm. The results for the experiments conducted in this chapter are presented in Section 7.6 and they are further discussed in Section 7.7. Finally the discussions on classifier performance evaluation is presented in Section 7.8.

7.1 Objectives of the Experiment

The objective of the experiments reported in this chapter were to create and study a system that can handle social media data which has big data properties. The system was able to handle unstructured social media data that arrives at high speeds from different sources,

store this data efficiently and processes it fast, in as close to real time as possible using low-cost commodity hardware.

7.2 Experiment setup

There are two ways to install Hadoop, these are: single node and multi node installations. In this study Hadoop was configured to run on a single node Hadoop cluster. As the name suggest, a single node Hadoop cluster has only a single machine whereas a multi-node Hadoop cluster will have more than one machine. The single node cluster is mainly used for studying and testing purposes to easily and efficiently test the sequential workflow in a smaller and more manageable environment. In a single node Hadoop cluster, all the daemons i.e. *datanode, namenode, tasktracker and jobtracker* run on the same machine or host, where as in a multi-node Hadoop cluster, all the daemons are up and run on different machines (Sinha, 2019). The multi-node setup has a master slave architecture in which one machine acts as a master that runs the namenode daemon while the machines acts as slave or worker nodes to run other Hadoop daemons. In order to achieve some concurrency or parallel processing for this experiment using a single node cluster, the *mapred.tasktracker.map.tasks.maximum and mapred.tasktracker.reduce.tasks.maximum* properties were set to 2 in the mapred-site.xml. These two property settings denote the maximum number of map tasks and reduce tasks that will be run simultaneously by a task tracker. In this case a maximum of 2 map tasks and 2 reduce tasks were set to run by the task tracker.

The data for the experiment was downloaded from Twitter a popular microblogging service which was discussed in more details in Chapter 5. A Twitter application that utilizes the Twitter API was created as described in Appendix B. In order to improve speed and efficiency it was proposed that the system work on Hadoop Ecosystem, a popular and widely adopted distributed parallel processing platform which uses the MapReduce programming paradigm. A total of 3,031,956 Tweets regarding the movie “Avengers Endgame” was collected over a period of 28 days. Apache Flume was used as a memory channel to filter and sink the generated Twitter data into HDFS. Sentiment analysis was performed on this data to determine the sentiments about the movie’s reviews on Twitter using the AFINN-111 lexicon which is described in more details in the Section 7.2. A lexicon based approach was used in this experiment because it is simple, more viable and more practical approach to

sentiment analysis of Twitter data since there's no need for training data (Yadav & Elchuri, 2013).

In most of the literature many authors such as Subramaniyaswamy et al. (2015) argue that Hadoop runs at its best in Ubuntu Linux. In many cases the reason Hadoop is mostly installed on Linux is mainly because Hadoop was built on Linux and also that Hadoop distribution already includes native libraries that are built for Linux, so this makes it much easier to set up and maintain on Linux computer systems than on any other platform. In this study Hadoop was installed on Microsoft Windows. It was possible to setup Hadoop on Microsoft Windows because starting from version 2.2 Apache supports Microsoft Windows operating system. The overall system overview of the study for tweet sentiment analysis is shown below in Figure 7.1. As can be seen in Figure 7.1 the system consists of the Twitter application (Twitter Streaming API), Flume, Hadoop File System (HDFS), the Java MapReduce implementation and the AFINN-111 lexicon which was used for scoring the tweets.

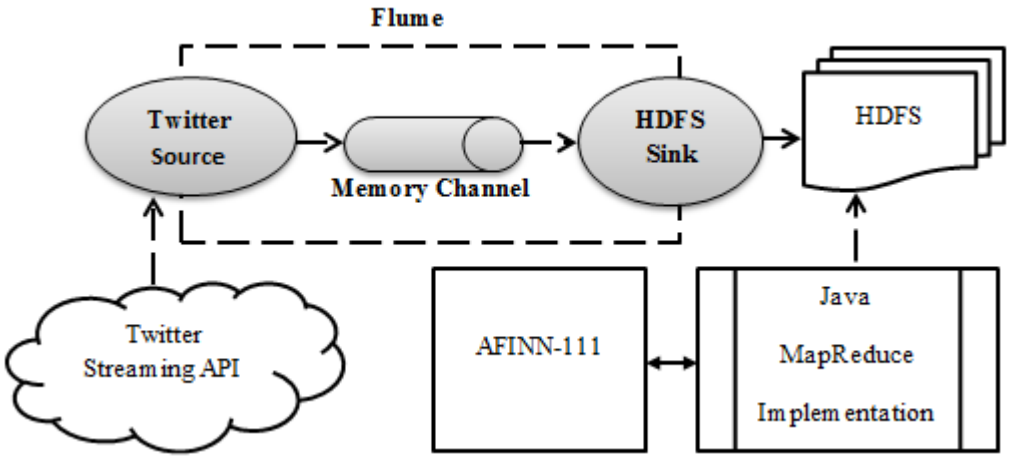

Figure 7.1: Overall System Architecture

7.3 Sentiment Analysis with AFINN Lexicon

In this study the AFINN-111(Nielsen, 2011) sentiment lexicon which is a dictionary of English words rated for valence with an integer between minus five (strongly negative) and plus five (strongly positive) was used. In Psychology, when discussing emotions, the term Valence refers to the intrinsic attractiveness/"good"-ness (i.e. positive valence) or averseness/"bad"-ness (i.e. negative valence) of an event, situation or object (Frijda, 1986). For example

emotions commonly referred to as “negative”, such as fear and anger are said to have a “negative” valence and emotions such as joy have a “positive” valence. According to Nielsen (2011) AFINN-111 was specifically constructed for Microblogs, it was constructed by manually labelled postings from Twitter which are scored for sentiment based on their positive or negative valence.

The reason AFINN-111 was chosen in this study was mainly because in a comparative study between various features of Twitter sentiment analysis by Koto & Adriani (2015) using four different datasets and nine feature sets, the experiments revealed that AFINN and SentiStrength (Thelwall et al., 2012) are the current best features for Twitter sentiment analysis. The latest version AFINN-111 contains 2477 words and phrases which were manually labelled by Finn Arup Nielsen between years 2009 and 2011. Afli et al. (2017) state that this 11 point range (-5 to +5) allows for finer granularity of analysis, considering that many of the current English language sentiment lexicons do not go beyond simply rating a word as positive, negative and neutral, which does not consider the degree of intensity of the sentiment expressed. An evaluation of the AFINN words list is available in Nielsen (2011) and Hansen et al. (2011). Table 7.1 shows some example entries from AFINN-111 lexicon.

Table 7.1: Sample of entries in the AFINN-111 file

Word	Polarity Score
await	-1
awaited	-1
awaits	-1
award	3
awarded	3
awards	3
awesome	4
awful	-3
awkward	-2
axe	-1
axed	-1
backed	1
backing	2
backs	1
bad	-3

7.4 Tokenization

This processing involves breaking up a given sentence into its individual sequence of words which are called tokens. The tokens are stored in an array of strings with each string stored in its own array position. This process is done in order to separate the words in a tweet so that each word is looked up in AFINN to find its individual polarity value. In this study Twokenizer a tokenizer designed specifically for tweets was utilized (Gimpel et al., 2010). According to Alhessi & Wicentowski (2015) Twokenizer properly handles the tokenization process of tweets without contorting URLs, hashtags or mentions.

7.5 Sentiment Analysis using the MapReduce Algorithm

The data stored in the HDFS was processed using a Java implementation of the MapReduce programming model. The full Java implementation of the MapReduce program is given in Appendix B. Figure 7.2 shows the pseudo code of the algorithm for performing sentiment analysis which was used to process the tweets in this experiment. Data was collected from Twitter using Apache Flume and Twitter API (as described in Appendix C). This data was stored as flume files in the HDFS. The MapReduce connects to the HDFS and processes these files one after the other using the logic in Figure 7.2. The processing of the files is depicted by the workflow in Figure 7.3. For every new file in the HDFS, the file was opened for reading and each tweet in the file was processed in sequence as depicted in Figure 7.3. One of the most important steps in the processing of the tweets is text pre-processing which forms a crucial step that every tweet has to go pass through. Text pre-processing methods used in this study were discussed in more details in the previous Chapter. As demonstrated in the first experiment in Chapter 6 text pre-processing improves the accuracy of the sentiment classification process and reduces the noise in the data, so the same text pre-processing methods were applied to the second experiment as well.

```
for each file in HDFS
begin
    open file
        for each tweet in file
        begin
            text pre-process tweet to clean it
            int tweet_total_sentiment_score = 0
            tokenize tweet
            for each word in tweet
            begin
                if word is found in AFINN
                    tweet_total_sentiment_score += word_score_in_AFINN
                else
                    go to next word in tweet
            end
        end
    close file
    write tweet ID and tweet_total_sentiment_score to Output csv file
end
go to next file
```

Figure 7.2: Sentiment analysis algorithm

Figure 7.4 illustrates how a single tweet is read from the HDFS and traces it as each text pre-processing method is applied to it and how the final score is eventually calculated by the MapReduce code using the AFFIN lexicon.

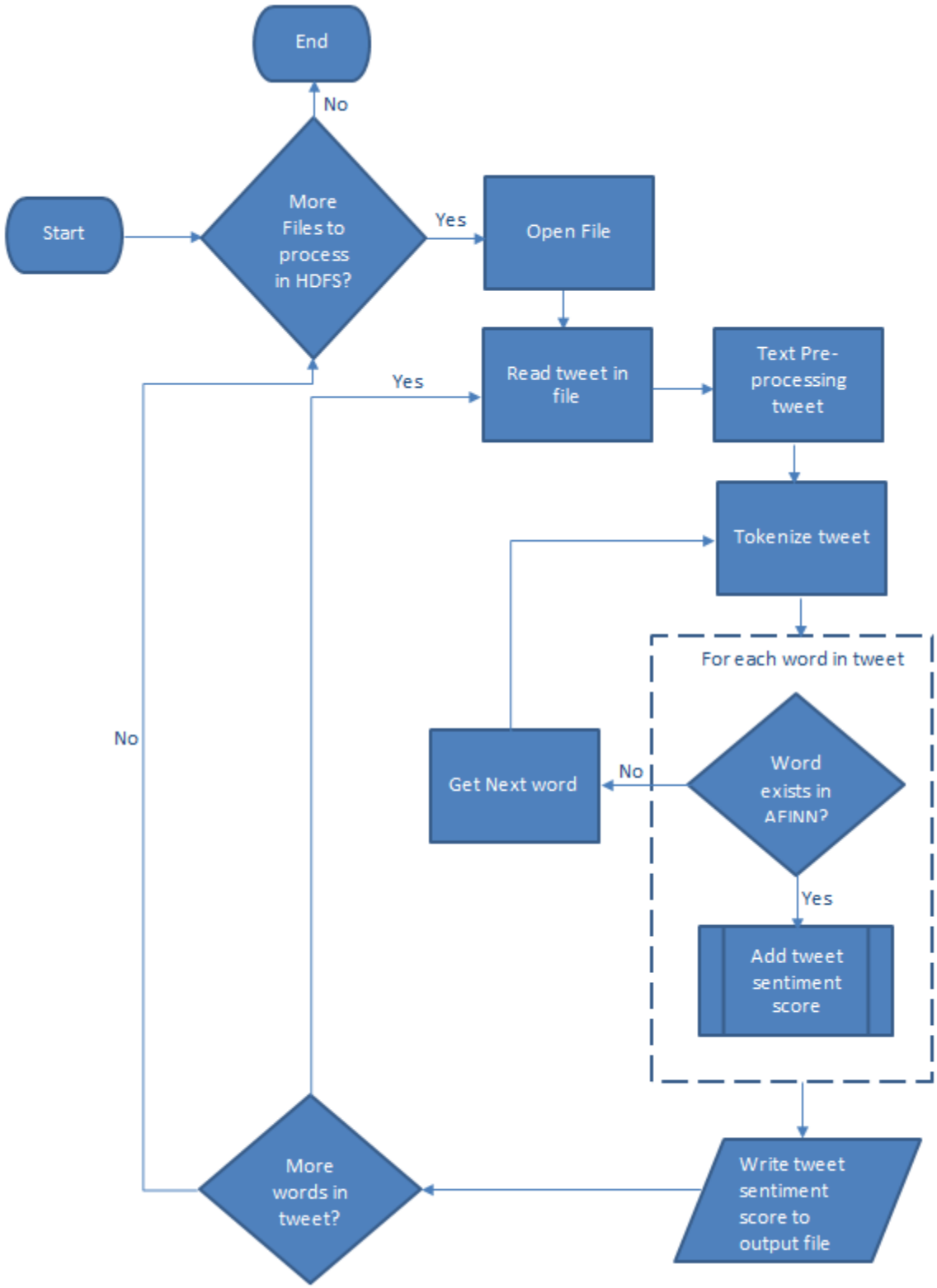

Figure 7.3: Sentiment Analysis Process Workflow

7.6 Experimental results for sentiment analysis

A total of 3,031,956 tweets mentioning the keyword “Avengers Endgame” in their text were downloaded over a period of 28 days and were analysed to determine the sentiments of the movie goers towards the movie Avengers Endgame which was released on 22 April 2019. The tweets were stored in the HDFS on disk on a single node cluster which was configured to a maximum capacity of 465.76 gigabytes and used a capacity of 25.23 gigabytes which was only 5.42% of the total capacity as shown in Table 7.2. A total of 6,441 files were downloaded and stored on 6,431 blocks i.e. 12,872 total file system objects in the HDFS.

Table 7.2: Data File System (DFS) Storage Types

Storage Type	Configured Capacity	Capacity Used	Capacity Remaining	Block Pool Used	Nodes In Service
DISK	465.76 GB	25.23 GB (5.42%)	200.87 GB (43.13%)	25.23 GB	1

In the Hadoop Framework, whenever any MapReduce job gets executed, the framework initiates counters to keep track of the job statistics such as number of bytes read or written to files. These counters reports various metrics for the MapReduce job and they also come in handy in the debugging processing when something goes wrong with the job. The counters for the experiment conducted for this dissertation are shown in Figure 7.5.

MapReduce built in counters consist of three categories: (1) File System Counters – these counters track two main details i.e. it tracks the number of bytes read and the number of bytes written by the file system. In this experiment, there were 165,431,027 read operations and 6,431 write operations in the HDFS. The write operations correlate with the number of the total data files which were downloaded. (2) MapReduce Framework – also known as task counters, these types of counters gather information about tasks over the course of their execution and the results are summarized over all the tasks in a given MapReduce job. (3) Job Counters – these counters are maintained by the job tracker so they do not need to be sent across the network, like the rest of the counters. Their purpose is to measure job-level statistics not values that change while the job is running. For example job counters can

measure the number of map tasks that were launched over the course of a job, even job that failed.

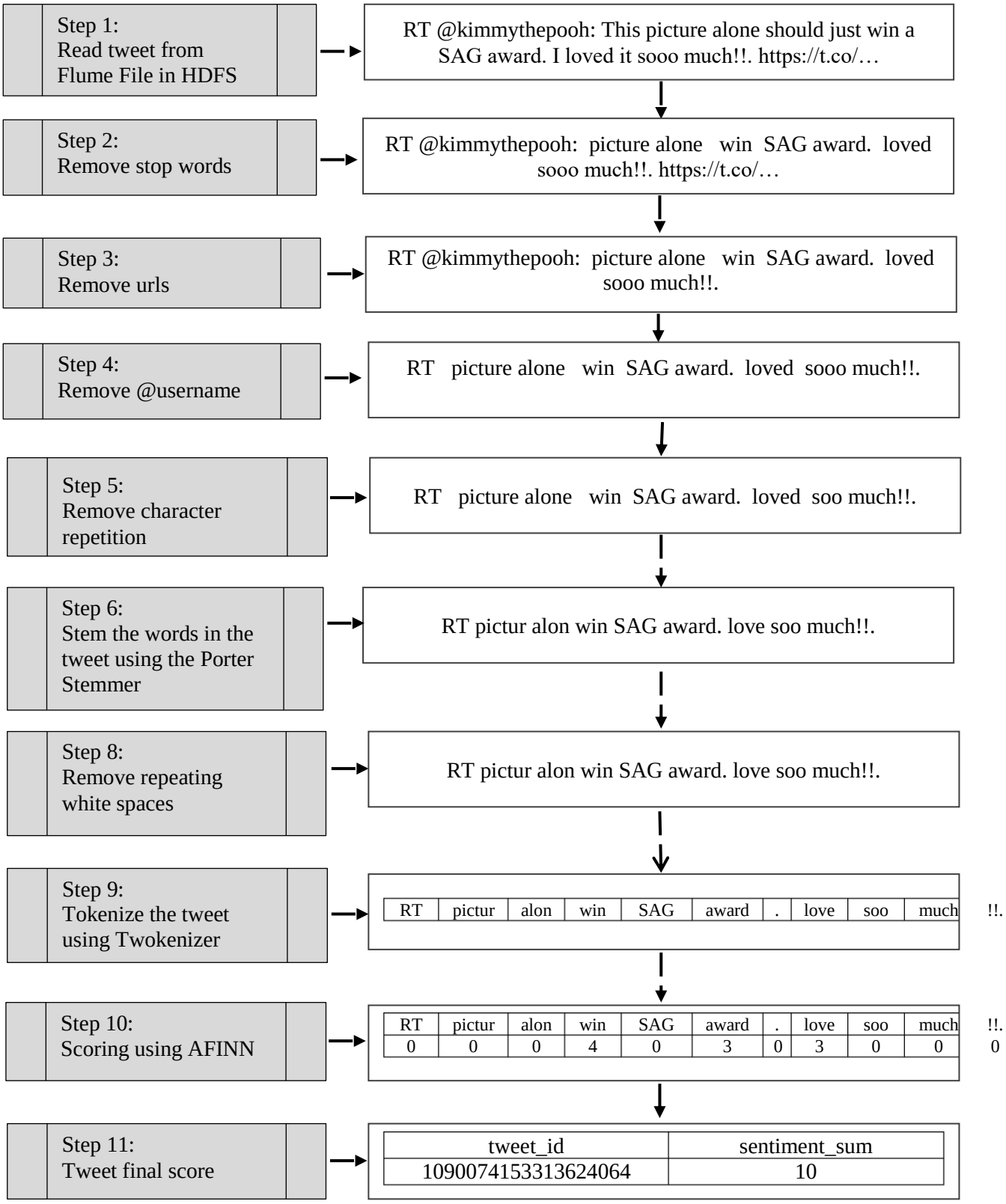

Figure 7.4: Scoring a tweet using AFINN

```
File System Counters
        FILE: Number of bytes read=173160477023
        FILE: Number of bytes written=322946169658
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=106581256753905
        HDFS: Number of bytes written=72945047
        HDFS: Number of read operations=165431027
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=6431
Map-Reduce Framework
        Map input records=3031956
        Map output records=3031956
        Map output bytes=72945047
        Map output materialized bytes=79047527
        Input split bytes=771360
        Combine input records=0
        Combine output records=0
        Reduce input groups=7856
        Reduce shuffle bytes=79047527
        Reduce input records=3031956
        Reduce output records=3031956
        Spilled Records=6063912
        Shuffled Maps =6428
        Failed Shuffles=0
        Merged Map outputs=6428
        GC time elapsed (ms)=220885
        CPU time spent (ms)= 12335000
        Physical memory (bytes) snapshot=13405769728
        Virtual memory (bytes) snapshot=33911930880
        Total committed heap usage (bytes)=12489133654016
Shuffle Errors
        BAD_ID=0
        CONNECTION=0
        IO_ERROR=0
        WRONG_LENGTH=0
        WRONG_MAP=0
        WRONG_REDUCE=0
File Input Format Counters
        Bytes Read=26734252149
File Output Format Counters
        Bytes Written=72945047
```

Figure 7.5: MapReduce Counters

The results of the process described in Figure 7.4 above were written to a Comma Separated Value (CSV) file by the MapReduce and stored in the HDFS. An extract of the results CSV file is shown in Table 7.3 below. The output file contains only two columns i.e. the Tweet Id column which uniquely identify each tweet and the Sentiment score column which store the total sentiment score of the tweet. There was no need to also write the Tweet

Text in this file since the original tweet can be found by looking up the Tweet Id in the original input file and this also reduces the size of the output file as shown in Table 7.4.

Table 7.3: Outputfile.csv

Tweet ID	Sentiment Score
1082320599131860000	-3
1082320613623020000	0
1082320618777960000	0
1082320635144080000	0
1082320636876340000	0
1082320640223460000	0
1082320641695640000	3
1082320648721100000	-3
1082320650335860000	-3
1082320651120100000	0
1082320654110780000	-3
1082320655079660000	0
1082320655381680000	0
1082320665879940000	3
1082320667293340000	-3
1082320677917590000	0
1082320687166110000	2

Table 7.4: Data Table

Tweet ID	Tweet Text
1082320599131860000	RT @kimmythepooh: This picture alone should just win a SAG award. I loved it sooo much!!. https://t.co/…
1082320613623020000	RT @MARMETHAZINE: This is honestly more epic than Avengers Endgame. https://t.co/OM9QFRJ5TM
1082320618777960000	RT @MTV: Congrats to @Avengers: Endgame on winning #BestMovie at the 2019 #MTVAwards! ?? https://t.co/42xKjUSLQz
1082320635144080000	RT @ComicBookNOW: AVENGERS: ENDGAME Anthony Mackie Addresses Taking Over Captain America From Chris Evans!\nhttps://t.co/I7JG9WVgiV https://…

Recalling from the literature review, in sentiment analysis there are many different types of sentiment analysis system. These systems ranges from focusing on polarity detection (positive, neutral or negative), detection of feelings and emotions (happy, sad, angry etc.), intent analysis (such as interested vs. not interested), aspect-based analysis (focusing on a particular aspects or features of a product or service people talk about) and multilingual sentiment analysis. A polarity analysis was implemented in the current study in which the

system considers the amount of positive and negative terms that appear within a given text, in this case, within a given tweet regarding the movie Avengers: Endgame, a 2019 American superhero film based on the Marvel Comics superhero team the Avengers, produced by Marvel Studios. In order to improve the level of polarity of the opinion the current study takes a fine-grained sentiment analysis approach, which is a subtype of the polarity analysis with a total of five sentiment classification categories.

The tweets were classified into the following five sentiment categories described below. The categories can even be further mapped onto a 5-star rating in a review, for example ranging from very positive = 5 stars and very negative = 1 star. The sentiment classification categories are as follows:

(1) Very Negative (< -3) – this class represents all tweets that scored the worst. Tweets with a sentiment score below minus three were placed in this category. This category represents a very bad rating of the movie. People in this category were not happy at all with the movie, in a 5 rating this would be rated a 1.

(2) Negative (-3 =< and < 0) – people in this class were not happy with the movie, but they were not as dissatisfied as the ones above. This category represents customers the business can win over by adjusting the product offering or service in order to address the issues which led to their dissatisfaction.

(3) Neutral (= 0) – Most of the tweet regarding the movie Avengers Endgame were neutral, these is most likely because the opinions were not necessarily reviews, they could probably be just discussion around the movie. This also means that the total sentiment score of the tweet was zero.

(4) Positive (0 < and =< 3) – tweets which were categorized as positive had a sentiment score above zero and below three. These tweets maps to a rating of four stars, meaning they thought the movie was good.

(5) Very positive (< 3) – movie goers in this category were thrilled with the movie and they had a lot of positive sentiments towards the movie. In the case of products and services these customers are very satisfied with the organisation's products or services. The aim is to keep all customers of the organisation in this category.

The following algorithm shown in Figure 7.6 below was used to classify each Tweet based on the score as explained above:

```
if (tweet_total_sentiment_score is greater than 3) then
        sentiment_class = 'Very Positive'
else if (tweet_total_sentiment_score is less than 3 and tweet_total_sentiment_score is
greater than 0) then
        sentiment_class = 'Positive'
else if (tweet_total_sentiment_score is equal to 0) then
        sentiment_class = 'Neutral'
else if (tweet_total_sentiment_score is less than 0 and tweet_total_sentiment_score is
greater than -3) then
        sentiment_class = 'Negative'
else if (tweet_total_sentiment_score is less than -3) then
        sentiment_class = 'Very Negative'
```

Figure 7.6: Sentiment Classification Process Algorithm

Table 7.5 shows the sentiment classification results for the data collected in this study.

Table 7.5: Sentiment classification results

CATEGORY	RANGE	COUNT
Very Negative	(< -3)	37,740
Negative	(-3 =< and < 0)	140,356
Neutral	(= 0)	2,384,041
Positive	(0 < and =< 3)	438,759
Very Positive	(< 3)	31,060

Figure 7.7 shows the results on a pie chart, Figure 7.8 shows the results on a bar graph and Figure 7.9 shows the same results on a plot graph for the sentiment classification results conducted in this experiment as recoded in Table 7.5 above. The three visualizations shown in Figure 7.7, Figure 7.8 and Figure 7.9 enables organisations to quickly visualize and interpret their results much more clear.

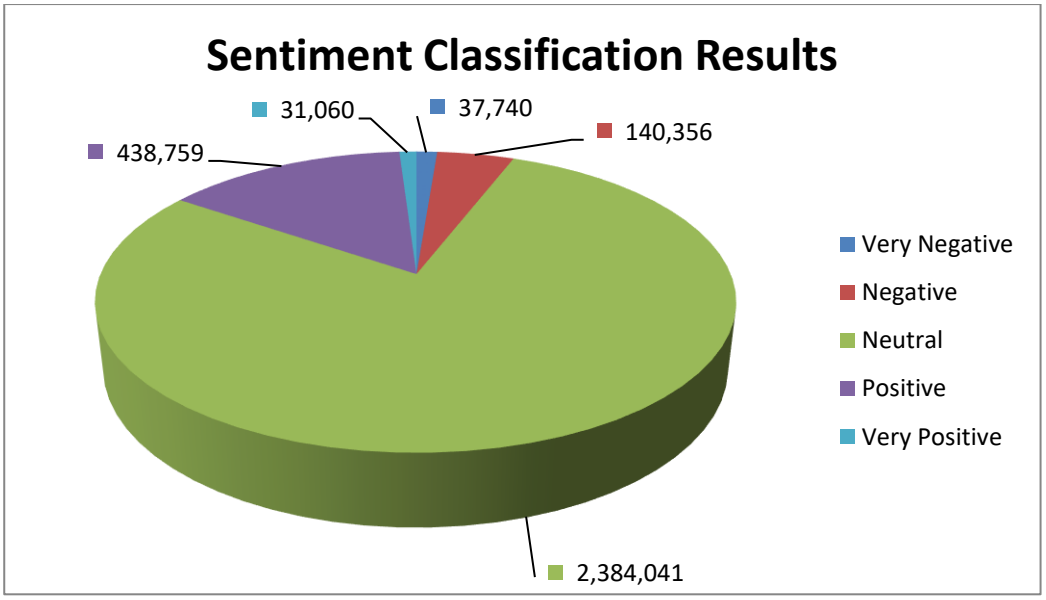

Figure 7.7: A Pie Chart Showing Sentiment Analysis Results

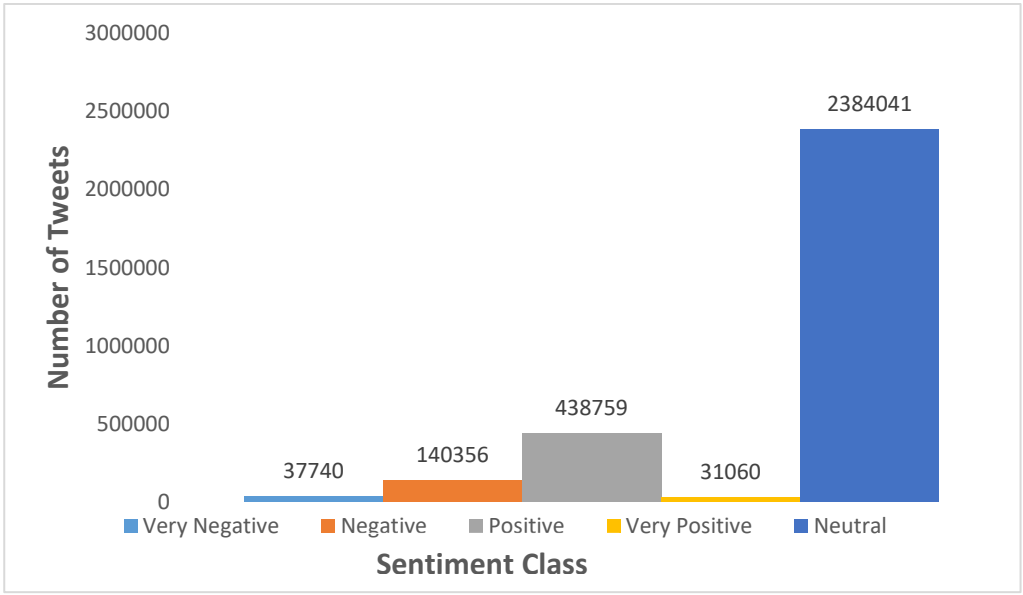

Figure 7.8: A Bar Graph Showing Sentiment Analysis Results

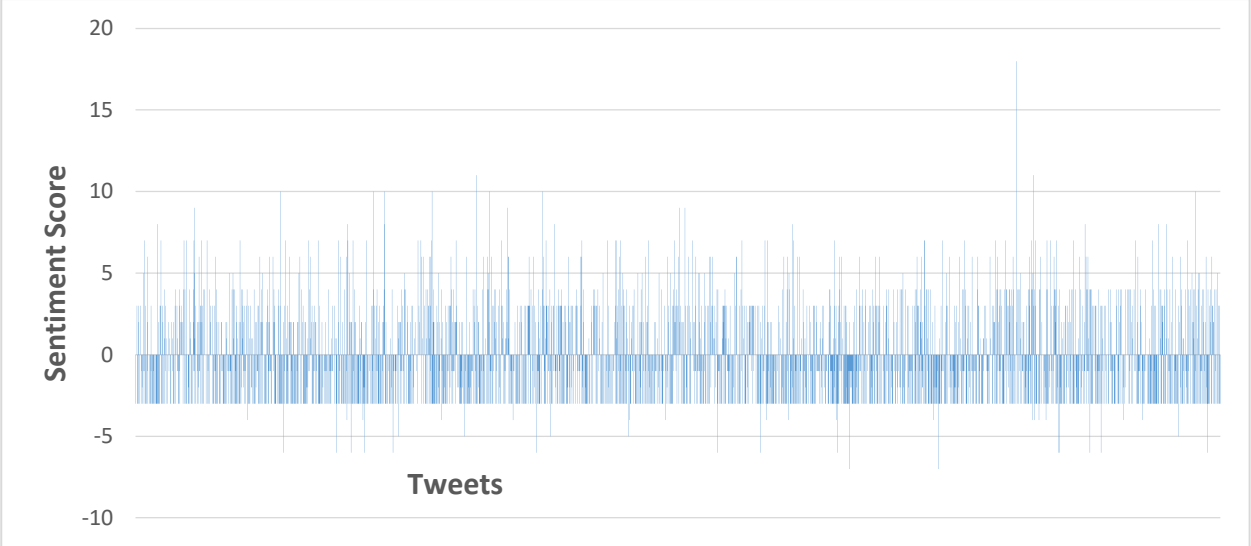

Figure 7.9: A Plot Graph Showing Sentiment Analysis Results

7.7 Discussion of the experimental results for sentiment analysis

MapReduce is a computational model and a software framework for writing applications which run on Hadoop and is capable of processing massive amounts of data in parallel on large clusters of computational nodes (Pol, 2016). In this study a single cluster with a single node was used to process and analyse big data from social media. The Hadoop architecture is highly scalable and is known for boosting the speed of data analysis by adding additional cluster nodes to increase throughput. As shown in Figure 6.8 (in bold text) above and Table 6.8 below, it took 12,335,000 milliseconds Central Processing Unit (CPU) time to pre-process and perform sentiment analysis of 3,031,956 tweets using a single cluster with one node. CPU time refers to the total cumulative CPU cycles spent for a task in milliseconds across the cluster (Brundesh, 2018). Based on these results the following estimates can be calculated just to see the impact of adding more nodes to the cluster:

Table 7.6: Estimated Impact of Scaling Hadoop Clusters

Time Unit	1 Node (Actual)	2 Nodes	10 Nodes	20 Nodes
Milliseconds	12,335,000	6,167,500	1,233,500	616,750
Seconds	12,335	6,167.5	1,233.5	616.75
Minutes	205.58333	102.79167	20.558333	10.27917
Hours	3.4263889	1.7131944	0.34263889	0.17132

As can be observed from Table 7.6, the more nodes added to the cluster the more tasks can be performed in shorter times in parallel. For example, it can be estimated that it could

have taken less than 2 hours for 2 nodes to process the same amount of data which took almost 4 hours to process using a single node. The experimental results demonstrated that the tweets were processed at a rate of 884,883.79121 tweets per hour which is equivalent to 245.80105 tweets per second as show in Table 7.7.

Table 7.7: Tweet processing rates

Number of Nodes	Number of tweets processed per:		
	Hour	**Minute**	**Second**
1	884,883.79121	14,748.06319	245.80105
2	1,769,767.58243	29,496.12637	491.60211
10	8,848,837.91212	147,480.63187	2,458.01053
20	17,697,675.82425	294,961.26374	4,916.02106

The 3,031,956 were collected over a period of 28 days, which amounts to about 108, 284.14 tweets per day. This figure can be converted to 4,511 tweets per hour, 75.20 tweets per minute and 1.25 tweets per second. Based on the estimates presented in Table 7.7, using a single cluster and a single node (with equal configuration) and downloading tweets at a rate of 1.25 tweets per second, it can be concluded that this data can be processed in real time.

7.8 Classifier Performance Evaluation

The current study adopts a lexicon-based method for sentiment analysis which utilises the AFINN-111 lexicon to classify Twitter data. In this section a proper evaluation of this classification method is conducted to demonstrate its accuracy and ultimately its effectiveness. According to Junker et al. (1999), the purpose of such an evaluation is in two folds: (1) The first goal is to demonstrate that the absolute effectiveness of the algorithm is acceptable for practical use (2) Lastly the evaluation can be used to demonstrate that the algorithm has a better or worse effectiveness than another competing algorithm.

In evaluating the model there is a need to define an evaluation metric in order to quantify the model's performance. One such commonly used metric for model evaluation is accuracy, which is simply the percentage of correctly classified predictions over the total number of instances (Williams et al., 2006). For example, a name gender classifier that predicts the

correct name 60 times in a test set containing 80 names would have an accuracy of 60/80 which is 75%. Informally, accuracy is the fraction of prediction our model got right.

Despite its wide adoption and its popularity, accuracy as a classification metric obscures two critical pieces of information. The first problem with accuracy is the underlying distribution of response value also known as the imbalance problem, which occurs when there are significantly fewer training instances of one class compared to another class and the second issue with accuracy is that it does not indicate the types of errors that the classifier is making class (Nguyen et al., 2009). A solution to these issues is the confusion matrix which was adopted in this study for performance evaluation.

A confusion matrix also known as an error matrix is essentially a table that records correctly and incorrectly recognized examples for each class on a set of test data for which the true values are known (Sokolova et al., 2006). The confusion matrix allows the visualization of the performance of an algorithm and it also enables easy identification of confusion between classes for example, one class is commonly mislabelled as the other. In a confusion matrix the number of correct and incorrect predictions is summarised with their count and broken down to their relevant classes, which enables us to identify the ways in which the classification model is confused when it makes predictions. This provides insight not only into the errors being made by the classifier, but also more importantly the types of the errors that the classifier is making. The rows of the confusion matrix represent the instances in the actual class and the columns shows the predicted values by the classifier. Tables 7.8 represents an example of a binary classifier with classes positive and negative:

Table 7.8: Binary Classifier Confusion Matrix

		Predicted	
		Positive	**Negative**
Actual	**Positive**	True Positives (TP)	False Positive (FP) or Type I errors
	Negative	False Negatives (FN) or Type II errors	True Negatives (TN)

The values in the confusion matrix in Table 7.8 are defined as follows (Bird et al., 2009):

1. True positives (TP): are relevant items that were correctly identified as relevant i.e. a positive observation is predicted to be positive.
2. True negatives (TN): are irrelevant items that were correctly identified as irrelevant i.e. a negative observation being identified as negative.
3. False positives (FP) or Type I errors: are irrelevant items that were incorrectly identified as relevant i.e. a negative observation predicted to be positive.
4. False negatives (FN) or Type II errors: are relevant items that were incorrectly identified as irrelevant i.e. a positive observation predicted to be negative.

The confusion matrix is not itself an evaluation metric, but there are many possible evaluation metrics that can be calculated from the confusion matrix for example, precision, recall and accuracy and several other metrics can be computed from the confusion matrix.

The confusion matrix can easily be adopted for cases where the data is classified into more than two classes as in the current study. In the current study, the tweets were classified into five classes i.e. very positive, positive, neutral, negative and very negative. One of the challenges in evaluating a lexicon based classifier is that there is no test data. In order to overcome this challenge, an approach was taken in the current study to generate test data manually by randomly selecting a sample of 2 390 tweets and manually assigning them into their relevant classes. In order to simplify this manual classification process, a decision was taken to consolidate the classes as follows: very positive and positive classes were consolidated into a single class i.e. positive and very negative and negative classes were also consolidated into a single class i.e. negative. The result of this process was the confusion matrix shown in Table 7.9.

Table 7.9: Twitter Data Confusion Matrix

n=2390	Predicted Class				**Total**
Actual Class		**Positive**	**Neutral**	**Negative**	
	Positive	**257**	49	6	312
	Neutral	75	**1736**	44	1855
	Negative	23	45	**155**	223
Total		355	1830	205	**2390**

Using the values in the confusion matrix above it gives us the ability to quantify and measure the various performance metrics such as accuracy, recall, precision and f-measure for comparison and evaluation purposes as discussed in the following sections.

7.8.1 Accuracy

Accuracy was defined by Kowcika et al. (2013) as the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications. Using the confusion matrix, accuracy is officially defined by Equation 7.1.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{7.1}$$

Using Equation 7.1, the accuracy of the classifier developed in this study can be calculated as follows:

$$\text{Accuracy} = \frac{257 + 1736 + 155}{2390}$$

$$\textit{0.8987} = \textit{89.87\%}$$

As shown in the calculation above, the accuracy of the lexicon classifier used to classify Twitter data in this study was 89.87%. This shows how often the classifier is correct; in this case it is generally correct 89.87% of the time. This metric is easy to interpret, but a high accuracy does not necessarily characterize a good classifier for reasons stated earlier. There are a number of evaluation criteria that can be utilised to evaluate a classifier depending on the evaluation goal and the type of data being analysed.

The Twitter data used in the experiments for the current study is an example of imbalanced data which was discussed above. Nguyen et al. (2009) argued that the measures that are most relevant to imbalanced data are precision, recall and F-measure. According to Nguyen et al. (2009) these metrics arise from the field of information retrieval and are used when performance of positive class (the minority class) is considered, since both precision and recall are defined with respect to the positive class. Table 7.10 shows a series of steps

followed to calculate the F-measure for our classifier. In order to calculate the F-measure we first have to calculate the recall and precision which are discussed in the next section.

7.8.2 Precision

The precision of a classifier is the percentage of positive predictions made by the classifier that are correct (Nguyen et al., 2009). Precision is given by Equation 7.2:

$$\text{Precision} = \frac{TP}{TP + FN} \tag{7.2}$$

Using Equation 7.2, the precision of the classifier developed in this study was calculated to be 0.809558 i.e. 81% as show in Table 7.10. A high precision indicates an example labelled as positive is indeed positive i.e. a small number of false positive. Precision helps us evaluate how many of the tweets were predicted correctly as belonging to a certain class out of all the tweets that were predicted. In other words, when the classifier predicts a given tweet to be positive, how often is it correct. In the calculations shown in Table 7.10, the results indicate that the classifier is correct 81.38% of the time.

7.8.3 Recall

The recall value of a classifier is the percentage of true positive patterns that are correctly predicted by the classifier (Nguyen et al., 2009). Recall is given by Equation 7.3:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7.3}$$

A recall of 0.818211 was calculated from the confusion matrix for our Twitter data as shown in Table 7.10. A high recall value indicates that the class is correctly recognized (i.e. small number of false negatives). The recall measures how many tweets were predicted correctly as belonging to a given class out of all the texts that's should have been predicted as belonging to that class.

7.8.4 F-measure

The F-measure (also known as the F-score or the F1 score) is the Harmonic Mean of the precision and recall (Fawcett, 2006). The F-measure will always be nearer to the smaller value of precision or recall as it punishes the extreme values more. The F-measure reaches its best value at 1 (i.e. perfect precision and recall) and worst at 0. The F-measure is given by Equation 7.4.

$$\mathrm{F-measure} = 2 * \frac{(Recall * Precision)}{Recall + Precision} \tag{7.4}$$

An F-measure of 0.8138618 was calculated for our classifier as indicated in Table 7.10. This value is closer to 1 (prefect precision and recall) than it is to 0. A high F-measure value signifies a high value for both precision and recall (Nguyen et al., 2009).

Table 7.10: Recall, Precision and F-Score Calculation

1.	Recall	$\frac{TP}{TP + FN}$
2.	Positive Recall	$\frac{257}{312} = 0.823717$
3.	Neutral Recall	$\frac{1736}{1855} = 0.935849$
4.	Negative Recall	$\frac{155}{223} = 0.695067$
5.	**Average Recall**	$\frac{(0.823717 + 0.935849 + 0.695067)}{3} = \mathbf{0.818211}$
6.	Precision	$\frac{TP}{TP + FP}$
7.	Positive Precision	$\frac{257}{355} = 0.72394$
8.	Neutral Precision	$\frac{1736}{1830} = 0.94863$
9	Negative Precision	$\frac{155}{205} = 0.756097$

9.	**Average Precision**	$\frac{(0.72394 + 0.94863 + 0.756097)}{3} = \mathbf{0.809558}$
10.	**F1-Score**	$2 * \frac{(\text{Recall} * \text{Precision})}{\text{Recall} + \text{Precision}}$ $2 * \frac{(0.818211 * 0.809558)}{0.818211 + 0.809558} = \mathbf{0.8138618}$

7.9 Conclusion

In the second experiment data was downloaded from Twitter using a Twitter application that utilizes the Twitter API. Apache Flume was used as a memory channel to filter and sink the generated Twitter data into HDFS. The downloaded Twitter data were pre-processed using text pre-processing methods which were proven to improve the accuracy of the results of sentiment analysis in the first experiment conducted in Chapter 6. Sentiment analysis was performed on this data to determine the sentiments about the movie reviews on Twitter using the AFINN-111 lexicon. In order to improve speed and efficiency a solution that utilises a combination of Apache Hadoop and the MapReduce programming paradigm was implemented in order to take advantage of this distributed parallel processing platform. The results demonstrate that social media data with big data properties such as Twitter data can be processed in real time using Hadoop and MapReduce.

In order to assess the lexicon classifier's performance, 2 390 tweets were manually labelled with classes and used as test data in order to assess classification performance. Performance metrics such as accuracy, precision, recall and F-measure were calculated in order to evaluate the classification model's performance. The performance of the lexicon classifier implemented in this study had a high accuracy, precision, recall and F-measure of more than 80%. The results of such an evaluation can be used to choose between two or more competing different models to be chosen for deployed into production or to adjust the performance of the model to suite a particular application purpose

CHAPTER 8: DISCUSSION

Social networking sites generate large volumes of opinionated data, in a variety of formats and this data arrives for processing at high velocity. Twitter, an online social networking service for example, as of the fourth quarter of 2018 had on average 321 million monthly active users, generating on average around 6,000 tweets every second. This is equivalent to generating 350,000 tweets per minute, 500 million tweets per day and around 200 billion tweets per year. This data if processed on time can be a valuable source of information for a business organisation. The challenge is that data of this sheer size are difficult to capture, store and analyse using traditional data management tools and technology (Zerhari et al., 2015). In this study a total of 3,031,956 tweets were downloaded over a period of 28 days from Twitter. Text pre-processing and sentiment analysis was performed on this data at a rate of 884,883.79 tweets per hour which is equivalent to 245.80 tweets per second. A single Hadoop cluster running a single node was cable of processing these massive amounts of data in real time.

The aim of this study was to research and answer the following main research question: "What technology and methods can be utilized to process social media text data fast and efficiently?" Since social media data such as Twitter data have the same characteristics as big data, this data also present the same challenges as big data. The solution in this study was to apply big data technologies to process data from Twitter a social networking site. Text pre-processing a method normally used in Natural Language Processing was also used to improve the efficiency and accuracy of data analysis.

Apache Hadoop an open source framework and its collective technology called the Hadoop Eco System offers a solution to processing big data, such as social networking data. Hadoop was designed for processing large data sets in parallel across cluster nodes in order to handle large and complex unstructured data sets that require a lot of processing. Hadoop provided a solution to collect, store and process the social networking data. To collect data which arrives at high velocity Apache Flume was used to stream the data from a Twitter source. Apache Flume which is part of the Hadoop Ecosystem was used for collecting, aggregating and moving large amounts of streaming data from Twitter and storing this data into the HDFS in real time. Tweets regarding the movie Avengers Endgame were filtered

from Twitter and were saved into the HDFS. This data was then accessed from the HDFS and processed using the MapReduce to perform sentiment analysis. MapReduce provided a computational model and framework to create an application which ran on Hadoop and performed sentiment analysis on Twitter data. The program was capable of processing massive amounts of data in parallel on large clusters of computational nodes. The sentiment analysis results from the MapReduce were exported to Microsoft Excel for visualization. A plot graph, a bar graph and a pie chart were generated from the sentiment analysis process with five sentiment classification categories: very negative, negative, neutral, good, very good.

The results of sentiment analysis of opinionated text data can be used by an organisation for social media monitoring, brand monitoring, voice of customer, customer service, workforce analytics and voice of employee, product analytics, market research and analysis. Organisations and individuals may no longer need to conduct surveys, or employ external consultants in order to find consumer opinions about their products/services and those of competitors because the information is already available in the form of user generated content on social networking sites such as Twitter and Facebook.

CHAPTER 9: CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

Online social networking sites, product reviews, forum post and blogs contain a huge amount of hidden unstructured and opinionated text which are difficult for a human reader to read, extract, summarize organize into usable forms. To add to this, traditional technologies which were mainly designed to work on structured transactional data are not capable of processing this massive amount of data at the speed necessary to generate value from this data. This study proposed a system that uses big data technology to perform sentiment analysis also known as opinion mining to perform automated opinion discovery and summarization using open source technology. The proposed method was able to stream data from Twitter in real time, and to store the data efficiently into the HDFS. The proposed system was also able to processes massive amounts of data in a single Hadoop cluster using MapReduce.

The results were then presented using Microsoft Excels in the form of 3 graphs backed by data. A pie chart, a bar graph and a plot graph were utilised to present the data from the results. This is similar to using Microsoft Excel dashboards which are normally used for decision making by managers and business leaders to track Key Performance Indicators (KPIs) and other metrics. These dashboards contain charts/tables/views which are backed by data.

9.2 Future Work

This study made some major advances in highlighting analysis methods and technologies that organisation can utilize in their efforts to convert online social media data into usable information. However there is still some more work to be done to improve the methods and technologies implemented in the current study:

(1) In this study Hadoop was configured to run on a single cluster. Using this architecture introduces a trade-off between reliability and writes/read bandwidth i.e. placing all replicas on a single node incurs the lowest write bandwidth penalty since the replication pipeline runs on a single node, but this offers no redundancy in which case if a the node fails , the data for that block is lost (White, 2012). In order to boost

speed of data analysis and to introduce redundancy future work could benefit from implementing a solution that implements a multi-node Hadoop cluster.

(2) The current study made use of the lexical approach to calculate the sentiment scores. This method not only simplifies the implementation of the MapReduce, but it also helps in cases where a sufficient corpus is not available to train a Machine learning algorithm, such as the Naïve Bayes, SVM, Linear regression, Deep learning and Neural networks. Using a Machine learning algorithm in the text classification process in future work could be done to compare the two approaches. This can be done through calculating and evaluating their performance using standard metrics such as precision, recall, F-measure and accuracy of the models. These metrics can be used to determine the most efficient one between the two methods.

(3) To create visualizations of the results of the sentiment analysis, the result CSV file had to be manually downloaded from the HDFS and exported to Microsoft Excel. This step could be automated and the results streamed constantly to a visualizing application that will draw and present the necessary visualizations automatically and in real time.

(4) The data used in this study was obtained from Twitter, but using Apache Flume, the same streaming data (log data) can be obtained from various web servers to HDFS. In future work data can be obtained from a different social networking site such as Facebook and analysed in the same way as described in the current study. The data ingestion mechanism can also be changed from using Apache Flume to using another different pipeline like Apache Kafka which is also another distributed, open source, high-throughput message bus that decouples data producers from consumers, but can support data streams for multiple applications, whereas Flume is specific for Hadoop and big data analysis.

REFERENCES

1. Abiteboul, S. (1997). Published. Querying semi-structured data. International Conference on Database Theory. Springer, 1-18.

2. Afli, H., Mcguire, S. & Way, A. (2017). Published. Sentiment translation for low resourced languages: Experiments on irish general election tweets. 18th International Conference on Computational Linguistics and Intelligent Text Processing.

3. Agarwal, A. (2016). *FrontPage - Hadoop Wiki* [Online]. Available: https://wiki.apache.org/hadoop [Accessed 12 January 2019 2019].

4. Agarwal, A., Xie, B., Vovsha, I., Rambow, O. & Passonneau, R. (2011). Published. Sentiment analysis of twitter data. Proceedings of the workshop on languages in social media. Association for Computational Linguistics, 30-38.

5. Aggarwal, C. C. (2011). An Introduction to Social Network Data Analytics. *In:* AGGARWAL, C. C. (ed.) *Social Network Data Analytics.* Boston, MA: Springer US.

6. Aggarwal, C. C. & Wang, H. (2011). Text mining in social networks. *Social network data analytics.* Springer.

7. Alhessi, Y. & Wicentowski, R. (2015). Published. Swatac: A sentiment analyzer using one-vs-rest logistic regression. Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015). 636-639.

8. Ali, S. M. & Tuteja, M. R. (2014). Data Mining Techniques.

9. Allahyari, M. & Kochut, K. (2015). Published. Automatic topic labeling using ontology-based topic models. Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on. IEEE, 259-264.

10. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B. & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.

11. Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F. & Manicardi, S. (2016). Published. A Comparison between Preprocessing Techniques for Sentiment Analysis in Twitter. KDWeb.

12. Apache Hadoop (2016). Welcome to apache hadoop. *Welcome to Apache Hadoop*.

13. Aung, K. Z. & Myo, N. N. (2017). Published. Sentiment analysis of students' comment using lexicon based approach. 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS). IEEE, 149-154.

14. Azevedo, A. I. R. L. & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*.

15. Baccianella, S., Esuli, A. & Sebastiani, F. (2010). Published. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. Lrec. 2200-2204.

16. Barbier, G. & Liu, H. (2011). Data mining in social media. *Social network data analytics.* Springer.

17. Benson, V., Filippaios, F. & Morgan, S. (2010). Online social networks: Changing the face of business education and career planning. *International Journal of e-Business Management,* 4**,** 20.

18. Bernard, J. J. & Sobel, M. K. (2009). Published. Mrico-blogging as online word of mouth branding. Proceeding of the 27thinternational conference extended abstracts on Human factors in computing systems. 19-21.

19. Best, B. & Thompson, M. (2018). *What is Twitter* [Online]. Available: http://tweeternet.com [Accessed 11 Oct 2018 2018].

20. Bird, S., Klein, E. & Loper, E. (2007). Introduction to Natural Language Processing. *University of Pennsylvania*.

21. Bird, S., Klein, E. & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*, " O'Reilly Media, Inc.".

22. Bird, S. & Loper, E. (2004). Published. NLTK: the natural language toolkit. Proceedings of the ACL 2004 on Interactive poster and demonstration sessions. Association for Computational Linguistics, 31.

23. Blumberg, R. & Atre, S. (2003). The problem with unstructured data. *Dm Review,* 13**,** 62.

24. Borgatti, S. P. (2009). 2-Mode concepts in social network analysis. *Encyclopedia of complexity and system science,* 6.

25. Borthakur, D. (2013). HDFS architecture guide. *Hadoop Apache Project,* 53.

26. Brady, D. A., Tzortzopoulos, P. & Rooke, J. (2013). Published. The development of an evaluation framework based on the design science approach. Proc. 21st. Ann. Conf. of the Int'l Group for Lean Construction.

27. Brundesh, R. (2018). *Counters in MapReduce* [Online]. Available: https://acadgild.com/blog/counters-in-mapreduce [Accessed February 24, 2018 2019].

28. Brynjolfsson, E. & Mitchell, T. (2017). What can machine learning do? Workforce implications. *Science,* 358**,** 1530-1534.

29. Cambria, E., Das, D., Bandyopadhyay, S. & Feraco, A. (2017). *A practical guide to sentiment analysis*, Springer.

30. Cambria, E., Olsher, D. & Rajagopal, D. (2014). Published. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. Twenty-eighth AAAI conference on artificial intelligence.

31. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide.

32. Chaturvedi, S., Bhirud, N. & Lowden, F. (2015). Solving Big Data Problem using Hadoop File System. *International Journal of Applied Information Systems (IJAIS)*, 23-28.

33. Chen, J., Chen, Y., Du, X., Li, C., Lu, J., Zhao, S. & Zhou, X. (2013). Big data challenge: a data management perspective. *Frontiers of Computer Science,* 7, 157-164.

34. Cherian, V. & Bindu, M. (2017). Heart disease prediction using Naive Bayes algorithm and Laplace Smoothing technique. *International Journal of Computer Science Trends and Technology (IJCST),* 5.

35. Christopher, C. (1998). Shilakes, Julie Tylman. Enterprise Information Portals. Merrill Lynch, Inc., New York, NY, November 16.

36. Cohn, M. (2018). *Social Media vs Social Networking* [Online]. CompuKol Communications. Available: https://www.compukol.com/social-media-vs-social-networking/ [Accessed 12 Oct 2018 2018].

37. Data, I. B. & Hub, A. (2015). The Four V's of Big Data. *IBM. Available online at www. ibmbigdatahub. com/infographic/four-vs-bigdata (last accessed February 29, 2016)*.

38. De Mauro, A., Greco, M. & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features. *Library Review,* 65, 122-135.

39. Dietrich, D., Heller, B. & Yang, B. (2015). Data Science & Big Data Analytics: Discovering. *Analyzing, Visualizing and Presenting Data*.

40. Dinucci, D. (1999). Fragmented future. *Print,* 53, 32-33.

41. Douglas, L. (2001). 3d data management: Controlling data volume, velocity and variety. *Gartner. Retrieved,* 6, 6.

42. Erickson, B. J., Korfiatis, P., Akkus, Z. & Kline, T. L. (2017). Machine learning for medical imaging. *Radiographics,* 37, 505-515.

43. Fan, W., Wallace, L., Rich, S. & Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM,* 49, 76-82.

44. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters,* 27, 861-874.

45. Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine,* 17, 37.

46. Feldman, R. & Sanger, J. (2007). *The text mining handbook: advanced approaches in analyzing unstructured data*, Cambridge university press.

47. Frijda, N. H. (1986). *The emotions*, Cambridge University Press.

48. Fu, G.-S., Levin-Schwartz, Y., Lin, Q.-H. & Zhang, D. (2019). Machine Learning for Medical Imaging. *Journal of healthcare engineering,* 2019.

49. Gandomi, A. & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management,* 35**,** 137-144.

50. García, A., Gaines, S. & Linaza, M. T. (2012). A lexicon based sentiment analysis retrieval system for tourism domain. *Expert Syst Appl Int J,* 39**,** 9166-9180.

51. Gartner. (2015). *Technology Research* [Online]. Available: http://www.gartner.com/technology/home.jsp [Accessed 02 November 18 2018].

52. Gimpel, K., Schneider, N., O'connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J. & Smith, N. A. (2010). Part-of-speech tagging for twitter: Annotation, features, and experiments. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.

53. Graham-Rowe, D., Goldston, D., Doctorow, C., Waldrop, M., Lynch, C., Frankel, F., Reid, R., Nelson, S., Howe, D. & Rhee, S. (2008). Big data: science in the petabyte era. *Nature,* 455**,** 8-9.

54. Gupta, V. & Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence,* 1**,** 60-76.

55. Haddi, E., Liu, X. & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science,* 17**,** 26-32.

56. Hansen, L. K., Arvidsson, A., Nielsen, F. Å., Colleoni, E. & Etter, M. (2011). Good friends, bad news-affect and virality in twitter. *Future information technology.* Springer.

57. Hatzivassiloglou, V. & Mckeown, K. R. (1997). Published. Predicting the semantic orientation of adjectives. Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics. Association for Computational Linguistics, 174-181.

58. He, W., Wu, H., Yan, G., Akula, V. & Shen, J. (2015). A novel social media competitive analytics framework with sentiment benchmarks. *Information & Management,* 52**,** 801-812.

59. Hearst, M. A. (1997). Published. Text data mining: Issues, techniques, and the relationship to information access. Presentation notes for UW/MS workshop on data mining. 112-117.

60. Hendrickx, I., Kozareva, Z., Nakov, P., Séaghdha, D. O., Szpakowicz, S. & Veale, T. (2013). Published. SemEval-2013 task 4: Free paraphrases of noun compounds. Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). 138-143.

61. Hltcoe, J. (2013). Semeval-2013 task 2: Sentiment analysis in Twitter. *Atlanta, Georgia, USA,* 312.

62. Hobbs, A. (2014). Social media and big data. *Houses of Parliament. Parliamentary Office of Science & Technology*.

63. Hortonworks. (2018). *Apache Hadoop* [Online]. Hortonworks Inc. Available: https://hortonworks.com/apache/hadoop/#section_1 [Accessed 30 June 2018 2018].

64. Hu, M. & Liu, B. (2004). Published. Mining and summarizing customer reviews. Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 168-177.

65. Hu, X. & Liu, H. (2012). Text analytics in social media. *Mining text data.* Springer.

66. Ibm. (2013). *Big Data Analytics* [Online]. IBM Corporation. Available: https://www-01.ibm.com/software/data/bigdata/what-is-big-data.html [Accessed 25 Oct 2018 2016].

67. Ismael, M. (2016). Solving Big Data Problems Using Hadoop and MapReduce. *In:* SYSTEMS, I. J. O. C. (ed.). Academia.edu.

68. Jain, N. & Srivastava, V. (2013). Data Mining techniques: A survey paper. *IJRET: International Journal of Research in Engineering and Technology,* 2**,** 2319-1163.

69. Java, A., Song, X., Finin, T. & Tseng, B. (2007). Published. Why we twitter: understanding microblogging usage and communities. Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis. ACM, 56-65.

70. Jianqiang, Z. & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access,* 5**,** 2870-2879.

71. Jivani, A. G. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl,* 2**,** 1930-1938.

72. Junker, M., Hoch, R. & Dengel, A. (1999). Published. On the evaluation of document analysis components by recall, precision, and accuracy. Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318). IEEE, 713-716.

73. Kaisler, S., Armour, F., Espinosa, J. A. & Money, W. (2013). Published. Big data: Issues and challenges moving forward. System sciences (HICSS), 2013 46th Hawaii international conference on. IEEE, 995-1004.

74. Kaplan, A. M. & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business horizons,* 53**,** 59-68.

75. Katal, A., Wazid, M. & Goudar, R. (2013). Published. Big data: issues, challenges, tools and good practices. Contemporary Computing (IC3), 2013 Sixth International Conference on. IEEE, 404-409.

76. Kaushik, C. & Mishra, A. (2014). A scalable, lexicon based technique for sentiment analysis. *arXiv preprint arXiv:1410.2265*.

77. Koto, F. & Adriani, M. (2015). Published. A comparative study on twitter sentiment analysis: Which features are good? International Conference on Applications of Natural Language to Information Systems. Springer, 453-457.

78. Kowcika, A., Gupta, A., Sondhi, K., Shivhre, N. & Kumar, R. (2013). Sentiment analysis for social media. *International journal of advanced research in computer science and software engineering*.

79. Kumari, S. (2016). impact of big data and social media on society. *Global Journal for research Analysis,* 5**,** 437-438.

80. Lee, I. (2018). Social media analytics for enterprises: Typology, methods, and processes. *Business Horizons,* 61**,** 199-210.

81. Liu, B. (2010). Sentiment Analysis and Subjectivity. *Handbook of natural language processing,* 2**,** 627-666.

82. Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies,* 5**,** 1-167.

83. Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*, Cambridge University Press.

84. Liu, B. & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. *Mining text data.* Springer.

85. Lock, M. (2016). THE HORSEPOWER OF HADOOP: FAST AND FLEXIBLE INSIGHT WITH RESULTS.

86. Lukka, K. (2003). The constructive research approach. *Case study research in logistics. Publications of the Turku School of Economics and Business Administration, Series B,* 1**,** 83-101.

87. Lutu, P. E. N. (2015). Web 2.0 computing and social media as solution enablers for economic development in Africa. *Computing in Research and Development in Africa.* Springer.

88. Ma, Y., Cambria, E. & Gao, S. (2016). Published. Label embedding for zero-shot fine-grained named entity typing. Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. 171-180.

89. Majumder, N., Poria, S., Gelbukh, A. & Cambria, E. (2017). Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems,* 32**,** 74-79.

90. Manning, C. D. & Schütze, H. (1999). *Foundations of statistical natural language processing*, MIT press.

91. March, S. T. & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems,* 15**,** 251-266.

92. Martin, J. H. & Jurafsky, D. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, Pearson/Prentice Hall Upper Saddle River.

93. Medhat, W., Hassan, A. & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal,* 5**,** 1093-1113.

94. Mesquita, A., Peres, P. & Oliveira, L. (2016). Published. Social Media as a Facilitator of Financial Literacy Competencies in e-Learning Courses: Contribution of the e-Finlit European Project. 3rd European Conference on Social M di R h Media Research EM Normandie, Caen, France. 232.

95. Minazzi, R. (2015). *Social media marketing in tourism and hospitality*, Springer.

96. Mittal, M., Balas, V. E., Hemanth, D. J. & Kumar, R. (2018). *Data Intensive Computing Applications for Big Data*, IOS Press.

97. Musser, J. & O'reilly, T. (2007). *Web 2.0: Principles and best practices*, O'Reilly Media.

98. Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F. & Stoyanov, V. (2016). Published. SemEval-2016 task 4: Sentiment analysis in Twitter. Proceedings of the 10th international workshop on semantic evaluation (semeval-2016). 1-18.

99. Narendra, B., Sai, K. U., Rajesh, G., Hemanth, K., Teja, M. C. & Kumar, K. D. (2016). Sentiment analysis on movie reviews: a comparative study of machine learning algorithms and open source technologies. *International Journal of Intelligent Systems and Applications,* 8**,** 66.

100. Neviarouskaya, A., Prendinger, H. & Ishizuka, M. (2010). Published. Recognition of affect, judgment, and appreciation in text. Proceedings of the 23rd international conference on computational linguistics. Association for Computational Linguistics, 806-814.

101. Nguyen, G. H., Bouzerdoum, A. & Phung, S. L. (2009). Learning pattern classification tasks with imbalanced data sets. *Pattern recognition.* IntechOpen.

102. Nhlabano, V. V. & Lutu, P. E. N. (2018). Published. Impact of Text Pre-Processing on the Performance of Sentiment Analysis Models for Social Media Data. 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD). IEEE, 1-6.

103. Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

104. Obar, J. A. & Wildman, S. S. (2015). Social media definition and the governance challenge: An introduction to the special issue.

105. Olson, M. (2010). Hadoop: Scalable, flexible data storage and analysis. *IQT Quart,* 1**,** 14-18.

106. Oussous, A., Benjelloun, F.-Z., Lahcen, A. A. & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences,* 30**,** 431-448.

107. Pak, A. & Paroubek, P. (2010). Published. Twitter as a corpus for sentiment analysis and opinion mining. LREc.

108. Pang, B. & Lee, L. (2004). Published. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 271.

109. Pang, B. & Lee, L. (2008). *Opinion mining and sentiment analysis*, Now Publishers Inc., Foundations trends in information retrieval, available at http://portal. acm. org/citation. cfm.

110. Pang, B., Lee, L. & Vaithyanathan, S. (2002). Published. Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 79-86.

111. Paul, M. J., Sarker, A., Brownstein, J. S., Nikfarjam, A., Scotch, M., Smith, K. L. & Gonzalez, G. (2016). Published. Social media mining for public health monitoring and surveillance. Biocomputing 2016: Proceedings of the Pacific symposium. World Scientific, 468-479.

112. Perkins, J. (2014). *Python 3 text processing with NLTK 3 cookbook*, Packt Publishing Ltd.

113. Pfeffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. & Bragge, J. (2006). Published. The design science research process: A model for producing and presenting information systems research. Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA, USA. 83-106.

114. Pol, U. R. (2016). Big Data Analysis Using Hadoop Mapreduce.

115. Pons, E., Braun, L. M., Hunink, M. M. & Kors, J. A. (2016). Natural language processing in radiology: a systematic review. *Radiology,* 279**,** 329-343.

116. Poria, S., Cambria, E., Hazarika, D., Majumder, N., Zadeh, A. & Morency, L.-P. (2017). Published. Context-dependent sentiment analysis in user-generated videos. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 873-883.

117. Poria, S., Cambria, E., Hazarika, D. & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. *arXiv preprint arXiv:1610.08815*.

118. Porter, M. F. (1980). An algorithm for suffix stripping. *Program,* 14**,** 130-137.

119. Prudhvi, D., Jaswitha, D., Mounika, B. & Bagal, M. (2015). A Survey on Big Data.

120. Python. (2017). *Download python 2.7.14* [Online]. Python Software Foundation. Available: https://www.python.org/downloads/ [Accessed 27/11/2017 2017].

121. Rajagopal, D., Cambria, E., Olsher, D. & Kwok, K. (2013). Published. A graph-based approach to commonsense concept extraction and semantic similarity detection. Proceedings of the 22nd International Conference on World Wide Web. ACM, 565-570.

122. Rajaraman, A. & Ullman, J. D. (2011). *Mining of massive datasets*, Cambridge University Press.

123. Rajput, R. & Solanki, A. K. (2016). Review of Sentimental Analysis Methods using Lexicon Based Approach. *IJCSMC,* 5**,** 159-166.

124. Robert, C. (2014). Machine learning, a probabilistic perspective. Taylor & Francis.

125. Rodrigues, A. P., Chiplunkar, N. N. & Rao, A. (2016). Sentiment Analysis of Social Media Data using Hadoop Framework: A Survey. *International Journal of Computer Applications,* 151.

126. Rosenthal, S., Farra, N. & Nakov, P. (2017). Published. SemEval-2017 task 4: Sentiment analysis in Twitter. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). 502-518.

127. Rosenthal, S., Nakov, P., Kiritchenko, S., Mohammad, S., Ritter, A. & Stoyanov, V. (2015). Published. Semeval-2015 task 10: Sentiment analysis in twitter. Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015). 451-463.

128. Russom, P. (2011). Big data analytics. *TDWI best practices report, fourth quarter,* 19**,** 1-34.

129. Sarker, A., Ginn, R., Nikfarjam, A., O'connor, K., Smith, K., Jayaraman, S., Upadhaya, T. & Gonzalez, G. (2015). Utilizing social media data for pharmacovigilance: a review. *Journal of biomedical informatics,* 54**,** 202-212.

130. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR),* 34**,** 1-47.

131. Selvi, S. T., Karthikeyan, P., Vincent, A., Abinaya, V., Neeraja, G. & Deepika, R. (2017). Published. Text categorization using Rocchio algorithm and random forest algorithm. Advanced Computing (ICoAC), 2016 Eighth International Conference on. IEEE, 7-12.

132. Seyede, Z. E. (2017). SOCIAL NETWORK SITES: ORAL PERFORMANCE OF EFL LEARNERS. *Theoretical & Applied Science***,** 116-121.

133. Sheela, L. J. (2016). A review of sentiment analysis in twitter data using Hadoop. *International Journal of Database Theory and Application,* 9**,** 77-86.

134. Siegel, C. F. (2006). *Internet Marketing: Foundations & Applications*, South Western Educational Publishing.

135. Simon, H. A. (1996). *The sciences of the artificial*, MIT press.

136. Simoudis, E. (1996). Reality check for data mining. *IEEE Intelligent Systems***,** 26-33.

137. Sinha, S. (2019). *Install Hadoop: Setting up a Single Node Hadoop Cluster* [Online]. Available: https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster [Accessed 2 June 2019 2019].

138. Sivarajah, U., Kamal, M. M., Irani, Z. & Weerakkody, V. (2017). Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research,* 70**,** 263-286.

139. Smith, K. (2019). *122 Amazing Social Media Statistics and Facts* [Online]. Available: https://www.brandwatch.com/blog/amazing-social-media-statistics-and-facts/ [Accessed 10 February 2019 2019].

140. Sokolova, M., Japkowicz, N. & Szpakowicz, S. (2006). Published. Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. Australasian joint conference on artificial intelligence. Springer, 1015-1021.

141. Statista. (2019). *Most popular social networks worldwide as of January 2019, ranked by number of active users (in millions)* [Online]. Available: https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/ [Accessed].

142. Stelzner, M. (2011). 2011 Social Media Marketing Industry Report. *Social media examiner***,** 1-52.

143. Stone, P. J., Dunphy, D. C. & Smith, M. S. (1966). The general inquirer: A computer approach to content analysis.

144. Stoneman, E. (2017). *Hadoop Succinctly*, CreateSpace Independent Publishing Platform.

145. Subramaniyaswamy, V., Vijayakumar, V., Logesh, R. & Indragandhi, V. (2015). Unstructured data analysis on big data using map reduce. *Procedia Computer Science,* 50**,** 456-465.

146. Taboada, M., Anthony, C. & Voll, K. D. (2006). Published. Methods for Creating Semantic Orientation Dictionaries. LREC. 427-432.

147. Taboada, M., Brooke, J., Tofiloski, M., Voll, K. & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics,* 37**,** 267-307.

148. Tan, A.-H. (1999). Published. Text mining: The state of the art and the challenges. Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases. sn, 65-70.

149. Tan, S., Cheng, X., Wang, Y. & Xu, H. (2009). Published. Adapting naive bayes to domain adaptation for sentiment analysis. European Conference on Information Retrieval. Springer, 337-349.

150. The Apache Software Foundation. (2019). *Welcome to Apache Flume — Apache Flume* [Online]. Available: https://flume.apache.org/ [Accessed 12 January 2019 2019].

151. Thelwall, M., Buckley, K. & Paltoglou, G. (2011). Sentiment in Twitter events. *Journal of the American Society for Information Science and Technology,* 62**,** 406-418.

152. Thelwall, M., Buckley, K. & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology,* 63**,** 163-173.

153. Tong, S. & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of machine learning research,* 2**,** 45-66.

154. Turney, P. D. (2002). Published. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 417-424.

155. Turney, P. D. & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS),* 21**,** 315-346.

156. Twitter. (2018). *Twitter* [Online]. Available: https://twitter.com/ [Accessed 10 February 2019 2019].

157. Vaishnavi, V. & Kuechler, B. (2013). 10-23. Design Science Research in Information Systems. Hämtad.

158. Vergeer, M. & Hermans, L. (2013). Campaigning on Twitter: Microblogging and online social networking as campaign tools in the 2010 general elections in the Netherlands. *Journal of Computer-Mediated Communication,* 18**,** 399-419.

159. Vijayarani, S., Ilamathi, M. J. & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks,* 5**,** 7-16.

160. Vijyalaxmi, M., Chopra, S., Oswal, S. & Chaturvedi, M. D. (2013). The How, When and Why of Sentiment Analysis. *International Journal of Computer Technology and Applications,* 4**,** 660.

161. Vinodhini, G. & Chandrasekaran, R. (2012). Sentiment analysis and opinion mining: a survey. *International Journal,* 2**,** 282-292.

162. Wakade, S., Shekar, C., Liszka, K. J. & Chan, C.-C. (2012). Published. Text mining for sentiment analysis of Twitter data. Proceedings of the International Conference on Information and Knowledge Engineering (IKE). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 1.

163. Ward, J. S. & Barker, A. (2013). Undefined by data: a survey of big data definitions. *arXiv preprint arXiv:1309.5821*.

164. White, T. (2012). *Hadoop: The definitive guide*, " O'Reilly Media, Inc.".

165. Williams, N., Zander, S. & Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review,* 36**,** 5-16.

166. Witten, I. H., Frank, E., Hall, M. A. & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann.

167. Wu, X., Zhu, X., Wu, G.-Q. & Ding, W. (2014). Data mining with big data. *IEEE transactions on knowledge and data engineering,* 26**,** 97-107.

168. Yadav, V. & Elchuri, H. (2013). Published. Serendio: Simple and Practical lexicon based approach to Sentiment Analysis. Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). 543-548.

169. Yang, Y. & Pedersen, J. O. (1997a). A comparative study on feature selection in text categorization. *In Icml,* 97**,** 412-420.

170. Yang, Y. & Pedersen, J. O. (1997b). Published. A comparative study on feature selection in text categorization. Icml. 412-420.

171. Younis, E. M. (2015). Sentiment analysis and text mining for social media microblogs using open source tools: an empirical study. *International Journal of Computer Applications,* 112.

172. Zeng, D., Chen, H., Lusch, R. & Li, S.-H. (2010). Social media analytics and intelligence. *IEEE Intelligent Systems,* 25**,** 13-16.

173. Zerhari, B., Lahcen, A. A. & Mouline, S. (2015). Published. Big data clustering: Algorithms and challenges. Proc. of Int. Conf. on Big Data, Cloud and Applications (BDCA'15).

174. Zhang, L. & Liu, B. (2016). Sentiment analysis and opinion mining. *Encyclopedia of Machine Learning and Data Mining***,** 1-10.

175. Zhu, X. (2007). *Knowledge Discovery and Data Mining: Challenges and Realities: Challenges and Realities*, Igi Global.

176. Zohuri, B. & Moghaddam, M. (2017). *Business Resilience System (BRS): Driven Through Boolean, Fuzzy Logics and Cloud Computation*, Springer.

Appendix A: Installing Java and Hadoop on Windows 7

In this study Hadoop 2.9.2 was installed on a 64 bit Windows 7 machine, which has an Intel i5 processor and 8 GB of RAM. To setup Apache Hadoop the first step was to install Java. The steps described in the Table A.1 below were followed:

Table A.1: Java and Hadoop installation steps

Step Number	Step Description	What was done
1	Installing Java	The first step was to download and install the latest java JDK version if you do not already have it and setup the **JAVA_HOME** path as your java installation path. **Java 1.8.0** was installed for this study. Java is available to download for free from: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html. In this study a Windows 64 bit version.
2	Configure Environment Variables	Add a new user variable and call it **JAVA_HOME**. Set its value to **C:\Java\jdk1.8.0_191**. Append the same value to the **PATH** variable as well. To verify if Java has been installed and setup properly, enter: **java –version** command on the command prompt and press enter. The system will respond with the currently installed java version. In this study the system responded with the below information: **java version "1.8.0_191"** **Java(TM) SE Runtime Environment (build 1.8.0_191-b12)** **Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)**
3	Install Apache Hadoop on the cluster	**Hadoop 2.9.2** was installed for this research; it is available to download for free from https://archive.apache.org/dist/hadoop/core/hadoop-2.9.2/

		After the download is completed, extract the **hadoop-2.9.2.tar.gz** and copy the extracted folder to this location C:\hadoop-2.9.2.
4	Configure Hadoop Environment variables	Add a new **user** variable **HADOOP_HOME** and set its value to: **C:\hadoop-2.9.2\bin** and append the same value to the end of the **PATH variable**. Similarly append the **sbin path** to the **PATH variable**. The sbin path points to important executables and command files for windows.
5	Add datanode and namenode	Create a new folder and name it data in **C:\hadoop-2.9.2**. Inside that folder add two more subfolders and call them **datanode** and **namenode**
6	Modify xml files	After the above configurations are done, an important step is to edit and modify the following four xml files: core-site.xml, hdfs.xml, mapred.xml and yarn.xml which are found in this path: **C:\hadoop-2.9.2\etc\hadoop**: **i) Core-site.xml** Edit core-site.xml file with notepad and provide values for the default file system and access port number as shown below: **<configuration>** **<property>** **<name>fs.defaultFS</name>** **<value>hdfs://localhost:9000</value>** **</property>** **</configuration>** **ii) Hdfs.xml** Edit hdfs-site.xml file with notepad and configure the number of time the data has to be replicated and also to configure datanode and namenode locations as show below:

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>C:\hadoop-2.9.2\data\namenode</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>C:\hadoop-2.9.2\data\datanode</value>
  </property>
</configuration>
```

iii) Mapred.xml

Make copy of the mapred-site.xml.template file and paste it in the same location and rename the copied file to mapred-site.xml. Open the mapred-site.xml with Notepad and add the below configuration properties. The configuration properties configure where to run MapReduce applications on Hadoop.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
</configuration>
```

iv) Yarn.xml

Open the yarn-site.xml file with Notepad to edit it. This file is used to configure namenode to get which aux-services wants to be used and also to recognize which class to be used for shuffling when

		aux-services are set. Change the configurations as shown below: **<configuration> <property> <name>yarn.nodemanager.aux-services</name> <value>mapreduce_shuffle</value> </property> <property> <name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name> <value>org.apache.hadoop.mapred.ShuffleHandler</value> </property> </configuration>**
7	Format the namenode	Open the Windows command prompt with administrator privileges and enter the following command and press enter to format the namenode: **hadoop namenode –format**
8	Start Hadoop	If all the above steps where followed correctly, the setup is now complete and Hadoop can now be started by entering the following command and pressing enter on the command prompt: **start-all.cmd** Windows will respond with four command prompt screens showing the **namenode, datanode, node manager and resourcemanager**
9	Testing the artefact	The setup can be verified by checking which services are running in the system after Hadoop was started. Enter the following command on the windows command prompt and press enter: **jps**

		Windows will respond with the command window showing the services currently running in the system as follows: **C:\Users\Valentino>jps** **12880 Jps** **15616 NodeManager** **8960** **21284 ResourceManager** **21624 DataNode** **20460 NameNode** Another way to also see the services is to browse to the following URLs: The **Resourcemanager** can be viewed by browsing to the following URL: http://localhost:8088/ The Web UI of the **Namenode** daemon can be viewed by browsing to the following URL: http://localhost:50070/

Appendix B: Installing Apache Flume

In order to download and setup Apache Flume for this study the steps shown in the Table B.1 were followed:

Table B.2: Apache Flume installation steps

Step Number	Step Description	What was done
1	Download Apache Flume	The first step is to download the latest version. Apache Flume is available to download for free from their website: http://flume.apache.org/download.html. In order to avoid having to recompile the source code, for this study the binary version was downloaded and installed (i.e**. apache-flume-1.8.0-bin.tar.gz**)
2	Extract the downloaded Flume files	Extract the **apache-flume-1.8.0-bin.tar.gz** with any file archiver utility such as **WinRAR** or **WinZip**.
3	Copy files	Create a new directory and call it **apache-flume-1.8.0-bin** in this location: **C:\apache-flume-1.8.0-bin**. **Copy** all the files from the extracted files apache-flume-1.8.0-bin.tar.gz to this new folder
4	Configure Apache Flume Environment variables	The following three Environment variables for Flume needs to be set as shown below: **4.1 CLASSPATH : %FLUME_HOME%\lib*** **4.2 FLUME_CONF : %FLUME_HOME%\conf** **4.3 FLUME_HOME : C:\apache-flume-1.8.0-bin** Append the path of the Apache Flume bin to the **PATH variable** as follows: **;C:\apache-flume-1.8.0-bin\bin;**

5	Creating folders in HDFS	This step was done at this stage to prepare for the next step which is configuring Apache Flume. The folders created in Hadoop at this stage are used to configure Apache Flume in the next step. For the experiments conducted in this study, the following folders were created in HDFS using the commands shown below: **5.1 C:\Users\Valentino>hadoop fs -mkdir /afinn_dir** **5.2 C:\Users\Valentino>hadoop fs C:/Experiments/AFINN-111.txt /affin_dir** **5.3 C:\Users\Valentino>hadoop fs -mkdir /input_dir** **5.4 C:\Users\Valentino>hadoop fs -mkdir /output_dir**
6	Configuring Apache Flume	Navigate to the following folder: **C:\apache-flume-1.8.0-bin\conf** and do the following: **6.1** Copy and paste the **flume-env.ps1.template** file in the same location. Rename the copy to **flume-env.ps1**. Edit this file with Notepad and change the following values: $JAVA_OPTS="-Xms500m -Xmx1000m -Dcom.sun.management.jmxremote" $FLUME_CLASSPATH="" # Example: "path1;path2;path3" **6.2** Copy and paste the **flume-conf.properties.template** file in the same location. Rename the copy to **flume-conf.properties**. Edit this file with Notepad and change its contents to the following values as indicated below:

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS

#TwitterAgent.sources.Twitter.type = org.apache.flume.source.twitter.TwitterSource
TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
```

TwitterAgent.sources.Twitter.channels = MemChannel TwitterAgent.sources.Twitter.consumerKey = dTDmMrN5XNmBynOIQzyZj1mmn TwitterAgent.sources.Twitter.consumerSecret = UyQ6WHd7ARlVig9SdvBsDTntZLXWxHdOFW8O0QLc4MKp0OEU38 TwitterAgent.sources.Twitter.accessToken = 2493822120-T1EBfSWIzBvCstWLoAF78zbe6y1nsUTcRfit8Ss TwitterAgent.sources.Twitter.accessTokenSecret=VNUomCbmthjpdthpoI2BVU43AQXCN0Rd8NZDNrQSh8pfL TwitterAgent.sources.Twitter.keywords = **Avengers Endgame** TwitterAgent.sinks.HDFS.channel = MemChannel TwitterAgent.sinks.HDFS.type = hdfs TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/input_dir TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text TwitterAgent.sinks.sink1.HDFS.rollInterval=900 TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000 TwitterAgent.sinks.HDFS.hdfs.rollSize = 0 TwitterAgent.sinks.HDFS.hdfs.rollCount = 1000000 TwitterAgent.channels.MemChannel.type = memory TwitterAgent.channels.MemChannel.capacity = 1000000 TwitterAgent.channels.MemChannel.transactionCapacity = 100		
7	Start Apache Flume	To start Apache Flume, first start Hadoop by entering the following command on the command prompt and press enter: **start-all.cmd** Once Hadoop starts running enter the following command to start Apache Flume and press enter: **flume-ng agent -name TwitterAgent -conf-file %FLUME_HOME%/conf/flume-conf.properties** The system will respond with a screen showing the data being streamed by Apache Flume into the HDFS. Since the input_dir was set to be the HDFS path to which the sink delivers data, the data from Flume will be saved in this folder.

This flume-conf.properties file described in Step 6 connects the HDFS and the Twitter Application created earlier. Basically this file contains key-value pair settings that configure the following components of Apache Flume:

- Name the components of the current agent
- Describe/Configure the source
- Describe/Configure the sink
- Describe/Configure the channel
- Bind the source and the sink to the channel

Naming the components – the configuration file defines the sources, the channels and the sinks and these are defined per agent. There can be multiple agents in Flume and these can be differentiated using a unique name and each agent has to be configured individually using this unique name. In this study a Twitter agent was used and the following values were set in the configuration file, which indicates that Twitter data is being transferred using a Twitter source through a memory channel to a HDFS:

TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS

A source is a component of a Flume agent that receives data from the data generators and transfers that data to one or more channels in the form of Flume event. In this study the source is the Twitter application created earlier, so its consumer key, consumer secret, access token and access token secrete where used as indicated below:

TwitterAgent.sources.Twitter.type = com.cloudera.flume.source.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey = dTDmMrN5XNmBynOlQzyZj1mmn
TwitterAgent.sources.Twitter.consumerSecret = UyQ6WHd7ARlVig9SdvBsDTntZLXWxHdOFW8O0QLc4MKp0OEU38
TwitterAgent.sources.Twitter.accessToken = 2493822120-T1EBfSWlzBvCstWLoAF78zbe6y1nsUTcRfit8Ss
TwitterAgent.sources.Twitter.accessTokenSecret=VNUomCbmthjpdthpoI2BVU43AQXCN0Rd8NZDNrQSh8pfL
TwitterAgent.sources.Twitter.keywords = Avengers Endgame

It is important to note that this is where the keywords used to filter data from Twitter are set. For this study data regarding the movie Avengers Endgame was collected as indicated above. Configuring the sink – a sink consumes data from the channels and delivers it to its

destination which might be another agent or a central store such as HDFS or HBase. In this study, data was stored in the HDFS; hence the sink type was set to HDFS as show below:

TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
TwitterAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/input_dir
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.sink1.HDFS.rollInterval=900
TwitterAgent.sinks.HDFS.hdfs.batchSize = 1000
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
TwitterAgent.sinks.HDFS.hdfs.rollCount = 1000000

It is also important to note that the input folder where the Twitter data was stored in HDFS is also set above. This property was set so that data is saved in a directory called input_dir in the HDFS as indicated above. Bind the source and the sink to the channel – various channels are provided in Flume to transfer data between sources and sinks. The below properties were used to describe the memory channel which was used in this study:

TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 1000000
TwitterAgent.channels.MemChannel.transactionCapacity = 100

Appendix C: The Twitter application

In order to collect Twitter data, a Twitter application is required to be created and registered on Twitter website: https://apps.twitter.com/. The application shown below was created for this research on Twitter for streaming data used in the experiments.

App details

Details and URLs

App icon

App icon is default, click edit to upload.

App Name
UP.Val.SentimentAnalyser

Description
SentimentAnalyser is a Sentiment Analysis application for Twitter data created by Valentine Nhlabano for Research Purposes at the University of Pretoria

Website URL
https://www..SentimentAnalyser.Home

The Consumer API key, Consumer secret key, Access token and Access token secret for the above applications are shown below. These key are required when configuring the agent in Flume.

Keys and tokens

Keys, secret keys and access tokens management.

Consumer API keys
dTDmMrN5XNmBynOIQzyZj1mmn (API key)

UyQ6WHd7ARlVig9SdvBsDTntZLXWxHdOFW8O0QLc4MKp0OEU38 (API secret key)

Access token & access token secret
2493822120-T1EBfSWIzBvCstWLoAF78zbe6y1nsUTcRfit8Ss (Access token)

VNUomCbmthjpdthpoI2BVU43AQXCN0Rd8NZDNrQSh8pfL (Access token secret)

Read, write, and direct messages (Access level)

Appendix D: MapReduce Algorithm

The Java MapReduce program which was used to perform sentiment analysis in this study is shown below. It consists of a map function, the reduce function and a main function:

```
public class twitter_mapper extends Mapper<LongWritable, Text, Text, Text> {
  private URI[] files;
  private HashMap<String, String> AFINN_map = new HashMap<String, String>();
  private List<String> StopWords_map = new ArrayList<String>();
  private final static String URL_REGEX = "((www\\.[\\s]+)|(https?://[^\\s]+))";
  private final static String CONSECUTIVE_CHARS = "([a-z])\\1{1,}";
  private final static String STARTS_WITH_NUMBER = "[1-9]\\s*(\\w+)";

  public void map(LongWritable key, Text value, Context context) throws IOException,
  InterruptedException {
    String twt;
    String processed_twt;
    String line = value.toString();
    String[] tuple = line.split("\\n");
    JSONParser jsonParser = new JSONParser();

    try {
        for (int i = 0; i < tuple.length; i++) {
            JSONObject obj = (JSONObject) jsonParser.parse(line);
            System.out.println(line);

            String tweet_id = (String) obj.get("id_str");
            twt = (String) obj.get("text");
            processed_twt = preprocess(twt);

            Twokenizer twokenizer = new Twokenizer();
            List<String> tokens = twokenizer.twokenize(processed_twt);

            int sentiment_sum = 0;
            for (String word : tokens) {
                  if (AFINN_map.containsKey(word)) {
                    Integer x = new Integer(AFINN_map.get(word));
                    sentiment_sum += x;
                  }
            }

            if(sentiment_sum > 4) {
                  System.out.println(twt);
                  System.out.println(processed_twt + " " + sentiment_sum);
            } else if (sentiment_sum < -3) {
                  System.out.println(twt);
                  System.out.println(processed_twt + " " + sentiment_sum);
            }

            context.write(new Text(tweet_id.trim() + ", " ), new
            Text(Integer.toString(sentiment_sum).trim()));
            }
      } catch (Exception e) {
            e.printStackTrace();
      }
}
```

```
public class twitter_reducer extends Reducer<Text, Text, Text, Text> {

  public void reduce(Text key,Text value, Context context)
                  throws IOException, InterruptedException {
              // process values
              context.write(key,value);
  }
}

public class twitter_driver implements Tool  {

  public static void main(String[] args) throws Exception {
  {
         try {
              Configuration conf = new Configuration();
              conf.addResource(new Path("/etc/hadoop/conf/core-site.xml"));
              conf.addResource(new Path("/etc/hadoop/conf/hdfs-site.xml"));
              conf.set("fs.defaultFS", "hdfs://localhost:9000");
              conf.set("mapreduce.jobtracker.address", "localhost:9000");
              conf.set("mapred.textoutputformat.separatorText", ",");

              Job job =Job.getInstance(conf,"Sentiment Analysis");
              job.addCacheFile(new URI("hdfs://localhost:9000/afinn_dir/AFINN-111.txt"));
              job.setJarByClass(twitter_driver.class);
              job.setMapperClass(twitter_mapper.class);
              job.setReducerClass(twitter_reducer.class);
              job.setMapOutputKeyClass(Text.class);
              job.setMapOutputValueClass(Text.class);
              job.setOutputKeyClass(NullWritable.class);
              job.setOutputValueClass(Text.class);
              job.setInputFormatClass(TextInputFormat.class);
              job.setOutputFormatClass(TextOutputFormat.class);

              FileSystem fs = FileSystem.get(conf);
              RemoteIterator<LocatedFileStatus> fileStatusListIterator = fs.listFiles(
                                  new Path("hdfs://localhost:9000/input_dir/"), true);
              while(fileStatusListIterator.hasNext()){
                     LocatedFileStatus fileStatus = fileStatusListIterator.next();
                     FileInputFormat.addInputPath(job, fileStatus.getPath());
              }
              FileOutputFormat.setOutputPath(job, new
              Path("hdfs://localhost:9000/output_dir/Output.csv"));

              System.exit(job.waitForCompletion(true) ? 0 : 1);
              }catch(Exception ex) {
                     System.out.print(ex.toString());
              }
  }
}
```

Appendix E: Pre-processing Implementation

In order to clean up the Twitter data the below Text pre-processing functions were implemented as part of the MapReduce program. The complete implementation is given below:

```
private void setupStopWord(Context context) throws IOException {
        FileSystem fs = FileSystem.get(context.getConfiguration());
                                FSDataInputStream in = fs.open(new
                Path("hdfs://localhost:9000/stopwords_dir/english"));
        try {
                BufferedReader reader = new BufferedReader(new InputStreamReader(in));
                String line;
                while ((line = reader.readLine()) != null) {
                        StopWords_map.add(line);
                }
                reader.close();
                in.close();
        } catch (Exception e) {
                e.printStackTrace();
        }
}

private static String StermWord(String word) {
        PorterStemmer stemmer = new PorterStemmer();
        stemmer.setCurrent(word); //set string you need to stem
        stemmer.stem();  //stem the word
        return stemmer.getCurrent(); //return the stemmed word
}

protected String preprocess(String tweet) {
        // Remove stop words
        String[] splits = tweet.toString().split(" ");
        for (String word : splits)
                if (StopWords_map.contains(word.toLowerCase())) {
                        tweet = tweet.replaceAll("\\b"+word+"\\b", "");
                }
        // remove urls
        tweet = tweet.replaceAll(URL_REGEX, "");
        // remove @username
        tweet = tweet.replaceAll("@([^\\s]+)", "");
        // remove character repetition
        tweet = tweet.replaceAll(CONSECUTIVE_CHARS, "$1$1");
        // remove words starting with a number
        tweet = tweet.replaceAll(STARTS_WITH_NUMBER, "");
        // escape HTML
        tweet = tweet.replaceAll("&", "&");
        tweet = StringEscapeUtils.unescapeHtml(tweet);
        // stem each word in the tweet using the PorterStemmer
        String result = "";
        String[] words = tweet.split("\\s+");
        for (int i = 0; i < words.length; i++) {
                result = result + " " + StermWord(words[i]);
        }
        // Remove repeating white spaces
        tweet = result.replaceAll("\\s+", " ");
        return tweet;
}
```

Appendix F: A tweet

A tweet is a massive JSON object with a lot of key value properties. An example of a tweet and its associated structure is given below:


```
object{29}
        in_reply_to_status_id_str:null
        in_reply_to_status_id:null
        created_at:Wed Apr 24 08:51:24 +0000 2019
        in_reply_to_user_id_str:null
        source:<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">
               Twitter for iPhone</a>
        retweeted_status{29}
        retweet_count:0
        retweeted: false
        geo:null
        filter_level:low
        in_reply_to_screen_name:null
        is_quote_status: false
        id_str:1120973502511833088
        in_reply_to_user_id:null
        favorite_count:0
        id:1120973502511833100
        text:RT @Simplenewsuk: Emilia Clarke: Game of Thrones' Daenerys star puts
             on saucy display at Time 100 Gala https://t.co/YXc2STmLrY https://t.co…
        place:null
        lang:en
        quote_count:0
        favorited: false
        possibly_sensitive: false
        coordinates:null
        truncated: false
        timestamp_ms:1556095884336
        reply_count:0
        entities{4}
        contributors:null
        user{39}
```

Appendix G: Snowball Porter Stemmer

The complete Porter Stemmer implementation in Snowball is given below. The implementation below uses the exact algorithm as described in the (Porter, 1980)

```
integers ( p1 p2 )
booleans ( Y_found )

routines (
   shortv
   R1 R2
   Step_1a Step_1b Step_1c Step_2 Step_3 Step_4 Step_5a Step_5b
)

externals ( stem )
groupings ( v v_WXY )

define v        'aeiouy'
define v_WXY    v + 'wxY'

backwardmode (

    define shortv as ( non-v_WXY v non-v )
    define R1 as $p1 <= cursor
    define R2 as $p2 <= cursor

    define Step_1a as (
        [substring] among (
            'sses' (<-'ss')
            'ies'  (<-'i')
            'ss'   ()
            's'    (delete)
        )
    )

    define Step_1b as (
        [substring] among (
            'eed'  (R1 <-'ee')
            'ed'
            'ing' (
                test gopast v  delete
                test substring among(
                    'at' 'bl' 'iz'
                        (<+ 'e')
                    'bb' 'dd' 'ff' 'gg' 'mm' 'nn' 'pp' 'rr' 'tt'
                    // ignoring double c, h, j, k, q, v, w, and x
                        ([next]  delete)
                    ''   (atmark p1  test shortv  <+ 'e')
                )
            )
        )
    )

define Step_1c as (
```

```
        ['y' or 'Y']
        gopast v
        <-'i'
    )

    define Step_2 as (
        [substring] R1 among (
            'tional'  (<-'tion')
            'enci'    (<-'ence')
            'anci'    (<-'ance')
            'abli'    (<-'able')
            'entli'   (<-'ent')
            'eli'     (<-'e')
            'izer' 'ization'
                      (<-'ize')
            'ational' 'ation' 'ator'
                      (<-'ate')
            'alli'    (<-'al')
            'alism' 'aliti'
                      (<-'al')
            'fulness' (<-'ful')
            'ousli' 'ousness'
                      (<-'ous')
            'iveness' 'iviti'
                      (<-'ive')
            'biliti'  (<-'ble')
        )
    )

    define Step_3 as (
        [substring] R1 among (
            'alize'   (<-'al')
            'icate' 'iciti' 'ical'
                      (<-'ic')
            'ative' 'ful' 'ness'
                      (delete)
        )
    )

    define Step_4 as (
        [substring] R2 among (
            'al' 'ance' 'ence' 'er' 'ic' 'able' 'ible' 'ant' 'ement'
            'ment' 'ent' 'ou' 'ism' 'ate' 'iti' 'ous' 'ive' 'ize'
                      (delete)
            'ion'     ('s' or 't' delete)
        )
    )

    define Step_5a as (
        ['e']
        R2 or (R1 not shortv)
        delete
    )

    define Step_5b as (
        ['l']
```

```
        R2 'l'
        delete
    )
)

define stem as (

    unset Y_found
    do ( ['y'] <-'Y' set Y_found)
    do repeat(goto (v ['y']) <-'Y' set Y_found)

    $p1 = limit
    $p2 = limit
    do(
        gopast v  gopast non-v  setmark p1
        gopast v  gopast non-v  setmark p2
    )

    backwards (
        do Step_1a
        do Step_1b
        do Step_1c
        do Step_2
        do Step_3
        do Step_4
        do Step_5a
        do Step_5b
    )

    do(Y_found  repeat(goto (['Y']) <-'y'))

)
```

Appendix H: Publications and Conference Presentations

NHLABANO, V.V. & LUTU, P.E.N. Impact of Text Pre-Processing on the Performance of Sentiment Analysis Models for Social Media Data. 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD), 2018. IEEE, 1-6.