SENTIMENT ANALYSIS SYSTEM IN BIG DATA ENVIRONMENT

WINT NYEIN CHAN

UNIVERSITY OF COMPUTER STUDIES, YANGON

JANUARY, 2019

Sentiment Analysis System in Big Data Environment

Wint Nyein Chan

University of Computer Studies, Yangon

A thesis submitted to the University of Computer Studies, Yangon in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

January, 2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

…..……. ……..…….....……

Date Wint Nyein Chan

ACKNOWLEDGEMENTS

First of all, I would like to thank the Union Minister, the Ministry of Education for giving the opportunity to attend this course leading to the doctoral degree courses at the University of Computer Studies, Yangon and giving provision to finish this research.

I would like to thank Dr. Mie Mie Thet Thwin, the Rector of the University of Computer Studies, Yangon, for supporting to this research.

I would like to express my deepest gratitude to my supervisor, Dr. Thandar Thein, the Rector (Acting) of the University of Computer Studies, Maubin, for providing me with an excellent atmosphere in doing research. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. Her patience and support helped me overcome many crisis situations within research and finish this dissertation. I appreciate her endless patience, positive outlook, ability to provide advice.

I wish to extend special thanks to Dr. Khine Moe Nwe, Professor and Dean of the Ph.D Courses, for her general guidance and encouragement.

I would like to express my respectful gratitude to Daw Ni Ni San, Lecturer, the Language Department, the University of Computer Studies, Yangon, for commenting my dissertation even at hardship. She has read and commented upon everything I have written, and pointed out the correct usage in my dissertation.

I also would like to thank a lot all my teachers for mentoring, encouraging, and recommending the thesis. And also, sincere thanks to all my friends for their motivating encouragement, for the stimulating discussions about research and for our fun time together. Most of the appreciations show to my friend Myat Nandar Oo for her great deal of time.

Last but not least, I would like to gratefully thank my late father who has been my constant source of inspiration and my mother and family who specifically offered strong moral and physical support, care and kindness, during the years of my Ph.D study. Without their full support, my dissertation would not have been possible.

ABSTRACT

Nowadays, Big Data, a large volume of both structured and unstructured data, is generated from Social Media. Social Media are powerful marketing tools and Social Big Data can offer the business insights. The major challenge facing Social Big Data is attaining efficient techniques to collect a large volume of social data and extract insights from the huge amount of collected data. Sentiment Analysis of Social Big Data can provide business insights by extracting the public opinions. The traditional analytic platforms need to be scaled up for analyzing a large volume of Social Big Data. Social data are by nature shorter and generally not constructed with proper grammatical rules and hence difficult to achieve high reliable result in Sentiment Analysis. Acquiring effective training data is a challenge, although learning based approaches are good for sentiment classification. Manual Labeling for training data is time and labor consuming. Sentiment analysis based on multiclass classification scheme is oriented towards classification of text into more detailed sentiment labels. However, multiclass classification with Single-tier architecture where single model is developed and entire labeled data is trained may decrease the classification accuracy. The presence of sarcasm, an interfering factor that can flip the sentiment of the given text, is one of the challenges of Sentiment Analysis. Real-time tracking and analytics is important for Social Big Data because the speed may indeed be the most important competitive business profits. Compared to batch processing of Sentiment Analysis on Big Data Analytics platform, Real-time analytic is data intensive in nature and require to efficiently collect and process large volume and high velocity of data.

In this research, proposed Sentiment Analysis system is implemented with different architectures on different platforms to provide valuable information by analyzing large scale social data in an efficient and timely manner. Firstly, Sentiment Analysis is implemented on traditional analytics platform by performing model selection which is evaluated by comparing the performance of three different machine learning algorithm (Naïve Bayes, Random Forest and Linear Regression). For developing scalable and high performance Sentiment Analysis system, Sentiment Analysis is implemented on Big Data Analytics Platform (Hadoop MapReduce). The system enables high-level performance of sentiment classification while taking advantage of combining lexicon-based classifier's effortless setup process and

learning based classifier. Multi-tier Sentiment Analysis system on Big Data Analytics Platform (MSABDP) is developed for achieving high level performance of multiclass classification. This system is implemented by combining lexicon and learning based classification scheme with Multi-tier architecture. Multi-tier Sentiment Analysis system with sarcasm detection on Hadoop (MSASDH) is proposed to achieve high-level performance of sentiment classification. MSASDH identifies sarcasm and sentiment-emotion by conducting rule based sarcasm-sentiment detection scheme and sentiment classification with Multi-tier architecture. Real-time Multi-tier Sentiment Analysis system (RMSA) is implemented to achieve high level performance of multi-class classification in Real-time manner. To improve the classification accuracy, the suitable classifier is selected by comparing the accuracy of three different learning based multiclass classification techniques: Naïve Bayes, Linear SVC and Logistic Regression.

On the traditional analytics platform, Naïve Bayes classifier is better and the proposed system can achieved the promising accuracy. The evaluation result shows that the proposed system on Big Data Analytics Platform has enabled to achieve the promising accuracy by 84.2% and is able to scale up to analyze the large scale data by decreasing the running time when adding more nodes in the cluster. The evaluation results show that the proposed MSABDP is able to significantly improve the classification accuracy over multi-class classification based on Single-tier architecture by 7%. The evaluation results show that detecting sarcasm can enhance the accuracy of Sentiment Analysis. The evaluation results show that Real-time Multi-tier Sentiment Analysis achieves the promising accuracy and Linear SVC is better than other techniques for Real-time Multi-tier Sentiment Analysis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

As the rapid growth of the Internet and online activity, many services such as blogging, podcasting, social networking and bookmarking are popular. These services allow users to create and share information within open and closed communities and contribute to the volumes of the data. According to IBM reports everyday "2.5 quintillion bytes of data" is created and data are increasing each year. In Social Networking, Twitter [110] has 320 million monthly active users and they posts 500 million tweets every day; Facebook has 936 million daily and 1,440 million monthly active users as of December, 2015. These factors are reasons of a rise of Big Data [99]. Big Data is characterized by the volume, velocity, veracity, variety, value and volatility of data.

With the advent of Big Data, the distillation and analysis of Big Data can offer a more thorough and business insights of enterprises, which can lead to enhanced productivity, stronger competitive position and greater innovation. In accordance with the potential that Big Data facilitate [81], an increasingly interest in studies of techniques for analyzing new and diverse digital data streams to reveal new sources of economic value, provide fresh insights into customer behavior and identify market trends in advance. The major challenge facing Big Data is attaining efficient techniques to collect a large volume of Big Data and extract insights from Big Data. The traditional analytic platforms need to be scaled up for analyzing a large volume of Big Data. Big Data Analytics [113] has become popular for analyzing and managing large volume of the structure and unstructured data. Hadoop is a good framework for Big Data Analytics as it provides scalability, cost-effective, flexible, fast, and secure and authentication, parallel processing, Availability and resilient nature. It is an open-source software framework comprises of two parts: storage part and processing part. The storage part is called the Hadoop Distributed File System (HDFS) and the processing part is called MapReduce.

Sentiment Analysis (SA) is one of the main agenda in Big Data that focus on various ways to analyze Big Data to identify patterns and relationships [86], make informed predictions, deliver actionable intelligence and gain business insight from this steady influx of information. SA is the process of using text analytics to mine

various sources of data for opinions. There are two main approaches [64] to the problem of SA: lexical approach and machine learning approach. In the lexical approach, the meaning of sentiment depends on the analysis of each word and / or phrase, often use emotional dictionaries: a list of emotional vocabulary dictionaries being searched in the text, weight calculations, feelings, and total weight functions are used. This technique is governed by the use of a dictionary consisting pretagged lexicons. The classification of text depends on the total score achieved. Machine learning based SA, it considers a general explication of text classification and can be solved by classifiers to compile a labeled text collection. The machine learning approach applicable to SA mostly belongs to supervised classification. A number of machine learning techniques have been adopted to identify opinions.

1.1 Big Data

Big Data can be classified as machine-generated, which refers to data that is created by a machine without human intervention, or as human-generated, which refers to data that humans, in interaction with computers, supply. The former refers to audio, music, image, speech and video data, to sensor data, such as RFID tags used to track locations, to Intelligent Lighting Control (ILC) sensors used to identify the location and conditions of goods on the supply chain for example, and to smart meter, medical device or Global Positioning System (GPS) data. The latter instead refers to social media posts, clickstream data or web contents.

The importance of Big Data is demonstrated by the fact that data are produced extensively every day in many forms and from many different sources. For example, more than 98,000 tweets are written every sixty seconds, 695,000 status updates are posted on Facebook, 11 million instant messages are written, 685,445 Google searches are lunched, more than 169 million emails are sent, more than 1820 TB of data are created, and there are 217 new mobile web users.

The Big Data has numerous advantages on society, science and technology. Some of the advantages [23] are described below:

- **Better decision-making:** Analytics of Big Data can provide business decision-makers the data-driven insights that is needed to be compete and growth of organization.
- **Increased productivity:** The insights gained from analytics often allow organizations to increase productivity.

- **Reduce costs:** Big Data tools help for increasing operational efficiency and reducing costs.
- **Improved customer service:** Social media, customer relationship management (CRM) systems and other points of customer contact give today's enterprises a wealth of information about customers, and this information can be used to better serve the customers.
- **Fraud detection:** One of the big advantages of Big Data analytics systems that rely on machine learning is that they are excellent at detecting patterns and anomalies.
- **Increased revenue:** Big Data tools help the organization to increase revenue and accelerate growth based on better insights.
- **Increased agility:** Many organizations use Big Data for better align IT and business efforts, and Big Data analytics support faster and more frequent changes to business strategies and tactics.
- **Greater innovation:** Innovation is another common benefit of Big Data because it can generate glean insights that the competitors don't have. The organization may be able to get out ahead of the rest of the market with new products and services.
- **Faster speed to market:** Big Data can be used to provide faster time-to-market.

1.2 Sentiment Analysis (SA)

In most circumstances, the principle goal of SA is to uncover individuals' opinions to achieve meaningful insight about products or services. Its point is to show valuable information to both customers and manufacturers. Instead of detailed reviews, it is established that both manufacturer and customers look upon summarized opinions. So the sentiments that are classified on positive, negative, or neutral sentiments are valuable for both parties in making the correct call.

Regardless of the substantial number of concentrates on opinion mining and SA techniques, the effect they have on individuals has been less investigated. There has been incredible emphasis on the methods utilized and less on how individuals can profit by the findings. Consequently, this examination expects to explore the human

component in opinion mining and SA research. To accomplish this point, we will deliberately audit the pertinent written works that have utilized the two methodologies. The examination offers a few commitments. The first and headmost significance of this study is to concentrate on both technical and nontechnical challenges of opinion mining and SA. Secondly, it establish emphasis on the areas of opportunity by regarding at the trends of application scope that would offer some conceivable areas for research.

1.3 Motivation of the Thesis

Social Media are popular as these services allow users to share information and express opinions on specific topics. All these constantly accumulating actions on social media generate large volumes and high velocity of Social Big Data (SBD). Therefore, Social Media are one of generating sources of Big Data and analyzing SBD can provide the valuable information. Figure 1.1 presents the different types of data sources Big Data researchers have used. According to the survey results [57], Administrative data were the most widely used: 1690 respondents (55 percent) used this type of data in their most recent research involving Big Data. Administrative data includes data collected by government departments and can include health, educational, and income data. Twenty-nine percent (927 respondents) have done research using some kind of social media data (including Facebook, Twitter, and other social media). The third most commonly used data type was commercial or proprietary data with 697 respondents (23 percent).

Reduction of storage costs and processing power is one of the main factors leading to the booming of Big Data. Because of the large volumes and high velocity of SBD, the problem of how to store large datasets and improve the performance of calculating has become significant and cannot be ignored. The efficient technique is necessary to extract the valuable information from Social Big Data. SA is one of the main agenda in Big Data that focuses on various ways to analyze Big Data to extract valuable information. A scalable SA is needed to analyze high volume and high velocity of SBD.

With more advanced Big Data technologies, SA can easily capture, quantify, retrieve and analyze consumers more effectively. Achieving high level performance of SA is important because SA can provide valuable insights and thus help organizations to formulate effective business strategies. It can help firms to monitor

brand and product performances, handle customer grievances, get in-depth information for strategic analysis. SA can help to track and come up with effective marketing campaigns. SA has become the backbone of digital strategies for most firms today. The velocity aspect is closely related in SA because social media is actively used by the users and real-time streaming data is generated. Therefore Real-time SA is needed to facilitate ad hoc valuable information.

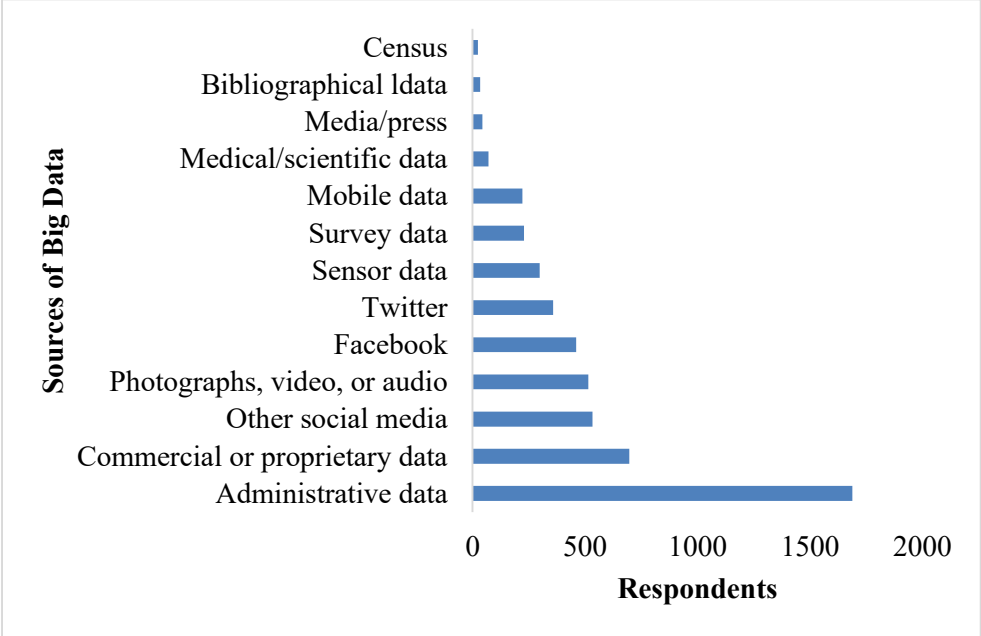


Figure 1.1 Percentage of Respondent in Sources of Big Data

1.4 Problem Statement

With the rapid growth of Social Networks, blogs, podcast and other types of online communication media, the way of expressions about people's opinions and feelings has significantly changed in recent years. These new tools can facilitate the establishment of original content, feelings, opinions, and ideas, connect millions of people over the Internet. A large volume and high velocity of SBD is generated from Social Media in every time. Social Media are powerful marketing tools and SBD can offer the business insights.

The major issue facing SBD is obtaining efficient techniques to extract insights from the large volumes and high velocity of data. SA of SBD can provide business insights by extracting the public opinions. The traditional analytic platforms need to be scaled up for analyzing a large volume of SBD. The first issue, developing scalable

and high performance SA for extracting valuable information from SBD in an efficient and timely manner, is the main issue of this thesis.

The second issue is posed by collecting high volume, velocity and variety of SBD. This has required to collect large amounts of data, and develop Big Data collection technology. Storing SBD on traditional way is problematic and traditional data protection is not efficient for large scale storage. In addition, the velocity of SBD requires the storage systems to be able to scale up quickly which is difficult to achieve with traditional storage systems.

And the third one is acquiring high-quality labeled dataset for learning based classifier. Manual Labeling of training data is time and labor consuming. Labeling is an essential stage of data classification in supervised learning. Historical data with predefined target classes is used for this model training style. Therefore, Labeling has become one of the important issues as each mistake or inaccuracy negatively affects a dataset's quality and the overall performance of a predictive model.

The fourth one is misclassification error in multi-class SA. This issue become more complicated when the number of classes is high. In multiclass classification, the classification performance tends to decrease when multiclass classification with Single-tier architecture, where single model is developed and entire labeled data is trained.

The fifth issue is detecting sarcasm that can degrade the performance of SA. In general, the current SA systems only consider the emotion of each word of the sentences. Thus, it is difficult to correctly judge the emotion of expressions such as sarcastic sentences that does not directly express their intention. Sarcasm is a special type of sentiment which plays a role as an interfering factor that can flip the polarity of the given text. Given an example of tweets: "I love amazing new iphone because it runs awfully". This example uses a word "love" to express the positive sentiment in a negative context. Therefore, the tweet is classified as sarcastic. Unlike a simple negation, sarcastic tweets contain positive words or even intensified positive words to convey a negative opinion or vice versa. As the previous example, sarcastic texts affect the classification accuracy of the SA.

The last one is developing high level performance of Real-time SA. Currently, SBD require real-time tracking and analytics because faster decisions can be provided with the analytics that better business profits. But the real-time analytics for SBD faces the velocity challenge because of the high velocity of streaming data.

1.5 Objectives of the Thesis

The main objective of this research is scaling up the traditional analytics platform to perform SA system for extracting valuable information from large scale social data in an efficient and timely manner. The goals of this research are the following:

(i) To develop high performance and scalable SA System by implementing SA on Big Data Analytics platform (Hadoop MapReduce)

(ii) To achieve high level performance of multiclass SA by implementing SA with different architectures (Single tier & Multi-tier)

(iii) To improve accuracy of SA via sarcasm and sentiment detection

(iv) To offer the optimal analysis results by analyzing the system with different Machine Learning algorithms

(v) To provide Real-time SA System by developing SA on Big Data Analytics platform (Spark)

This research develops high performance and scalable SA System which can provide valuable information that is significant for use of opinions in the practical decision making process. Moreover it can provide real-time analysis with the promising accuracy.

1.6 Contributions and Research Direction

The main issue of this thesis is developing scalable and high performance SA by scaling up the traditional analytics platform. To solve this issue, SA on Big Data Analytics platform is proposed to extract valuable information from large scale social data in an efficient and timely manner. The second issue is collecting and storing large volumes of tweet stream data. Apache Flume is the solution of data collection and HDFS is the solution of large volumes of data storage. The third issue is the necessity of effective training data for learning based classifier. In order to get effective training data for learning based classifier, lexicon-based approach is adopted which can reduce time and labor consuming. The fourth issue is misclassification error in Single-tier multiclass SA. To solve this issue, Multi-tier multiclass SA is proposed. The fifth issue is Performance degradation due to the interfere factor (sarcasm). Therefore, Multi-tier SA system with sarcasm detection on Hadoop is develop in order to achieve high-level performance of sentiment classification by classifying sentiment and sarcasm that can flip the sentiment of the given text.

Finally, to solve the real time issue, the proposed SA is implemented on Spark platform.

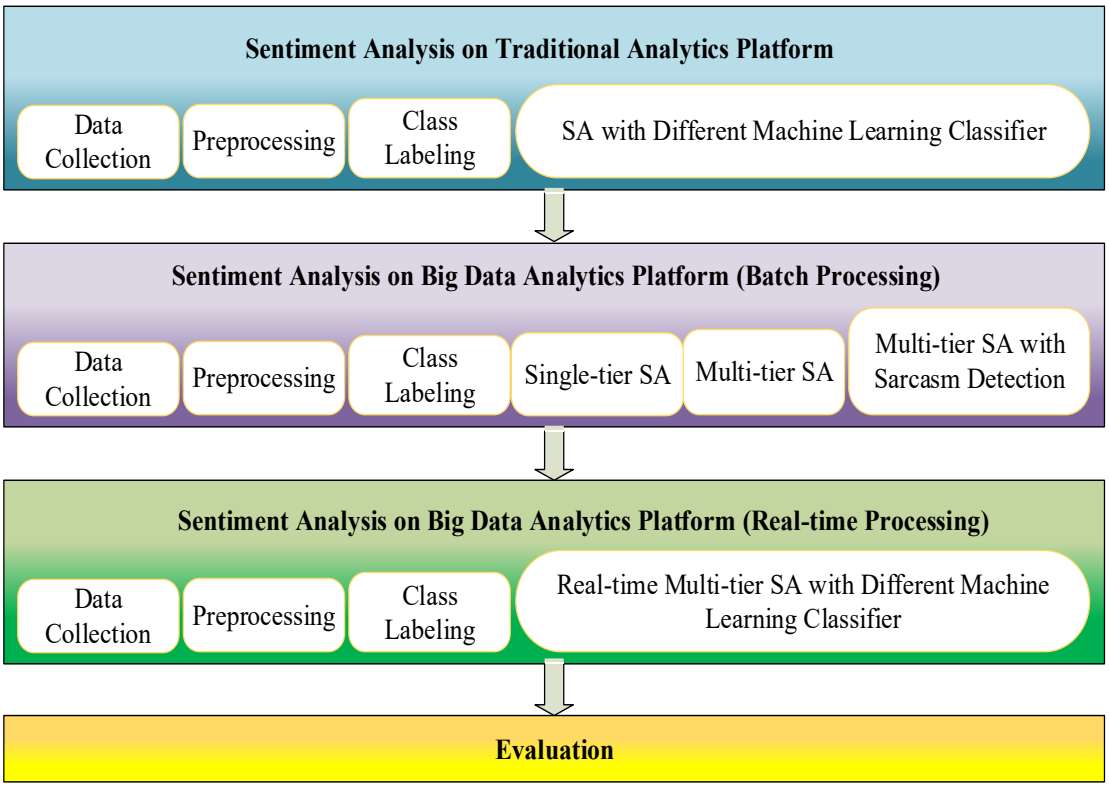


Figure 1.2 Research Direction

The processes of the research direction are illustrated in Figure 1.2. The first stage is implementing SA on traditional analytics platform. Model selection is evaluated by comparing three different machine learning algorithm (Naïve Bayes, Random Forest and Linear Regression. The experiment is implemented with Java. As a result, Naïve Bayes Classifier has been selected the (approximately) best classifier.

The second stage is developing SA on Big Data Analytics platform (Hadoop MapReduce). The Big Data Analytics Platform is implemented for analyzing large scale social data by using Apache Flume, HDFS, MapReduce [45] and Mahout Machine Learning Library [3]. In this platform, the SBD is collected by Apache Flume and the collected raw data is preprocessed to remove noisy data. Single-tier SA is developed on this platform by using lexicon and Naïve Bayes classifier. In this case SentiStrength lexicon based classifier [103] runs on distributed platform for providing training data of learning based classifier. Naïve Bayes classifier [60] (in Mahout Library) is used for developing prediction model and the incoming data are predicted by the model for providing the sentiment values. The evaluation result shows that the

proposed system has enable to achieve the promising accuracy by 84.2% and is able to scale up to analyze the large scale data by decreasing the running time when adding more nodes in the cluster.

SA based on multiclass classification with Single-tier architecture, where single model is developed and entire labeled data is trained, may decrease the classification accuracy. Therefore, Multi-tier SA system on Big Data Analytics Platform (MSABDP) is proposed to achieve high level performance of multiclass classification. The MSABDP efficiently analyze large scale data set by adopting distributed processing environment since they have been implemented using a MapReduce framework and a Hadoop distributed storage (HDFS). This system is implemented by combining lexicon and learning based classification scheme with Multi-tier architecture. The evaluation results show that the proposed MSABDP is able to significantly improve the classification accuracy over multiclass classification based on Single-tier architecture by 7%.

The presence of sarcasm, an interfering factor that can flip the sentiment of the given text, is one of the challenges of SA in social Media text, especially tweets. Therefore Multi-tier SA system with sarcasm detection on Hadoop (MSASDH) is proposed to extract the opinion from large volumes of tweets. To achieve high-level performance of sentiment classification, MSASDH identifies sarcasm and sentiment-emotion by conducting rule based sarcasm-sentiment detection scheme and sentiment classification with Multi-tier architecture. The evaluation results show that detecting sarcasm can enhance the accuracy of SA. The work performed in this area has resulted in the publication.

In microblogging environment the real-time interaction is a key feature and thus the ability to automatically analyze information and predict user sentiments as discussions develop is a challenging issue. For that reason, the third stage is developing Real-time Multi-tier SA system (RMSA) on Big Data Analytics Platform (Spark). The Platform is implemented by using Apache Flume, Hadoop Distributed File System (HDFS) [79], Spark streaming [105] and Spark Machine Learning Library (Spark MLlib) [104]. To improve the classification accuracy, the suitable classifier is selected by comparing the accuracy of three different learning based multiclass classification techniques: Naïve Bayes, Linear SVC and Logistic Regression. The evaluation results show that Real-time Multi-tier SA will achieve the

promising accuracy and Linear SVC is better than other techniques for Real-time Multi-tier SA. The work performed in this area has resulted in the publication.

The work performed SA on Traditional Analytics platform has been published in [p1]. The works related to SA on Big Data Analytics platform (Hadoop) have been published in [p2, p3 and p4]. The work related to Real-time SA on Big Data Analytics platform (Spark) has been published in [p5] of Author's Publication section.

1.7 Organization of the Thesis

The first chapter outlines the study areas and defines the goals and aims of the study. The research issues is presented and the underlying hypotheses are stated. An overview of the methodology is presented.

The second chapter reviews the literature to build the theoretical foundation of the study and gives an overview of research related to the area of this thesis that has been conducted thus far. It identifies key issues which form the basis of the research.

The third chapter highlights the traditional analytics platform and Big Data Analytics Platform for analyzing SBD.

The fourth chapter provides SA System based on Single tier and Multi-tier architecture for analyzing Multi-class SBD.

The fifth chapter presents a detailed description of proposed Real-time SA System on Spark. The experimental results for the classification model and scalability are put together in their related chapters.

The sixth chapter concludes the whole research work and the effectiveness of the research by the result discussion, the scope and limitations of the research and finally points out with future research directions.

1.8 Chapter Summary

This chapter presents the introduction of Big Data and evolution of SBD. As the blooming of Big Data, the enterprises face the challenges about what to do with the data and how to extract information from this data. Analytics is the process of collecting, managing and analyzing large amount of data that is important for the business. The process of analyzing and processing this huge amount of data is called Big Data Analytics. The volume, variety and velocity of Big Data cause performance problems when processed using traditional data processing techniques. Therefore the role of Big Data technologies such as Big Data platforms and Big Data Analytics

techniques are important and summarily discussed in this chapter. As the evolution of SBD, this data can give businesses valuable insight into how consumers observe their brand, and allow them to actively make business decisions to continue their trade. Hence, it becomes essential for the enterprises to sentiment social media data (SBD) to make predictions. However, to process SBD requires open-source Big Data technologies and SA techniques. Consequently the various SA techniques are also introduced in this chapter. Moreover, the motivation and problem statement of the thesis are also described in this chapter. This chapter briefly explains the objectives, contributions, and overview of the thesis. Detail explanations of this thesis will show in next chapters.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORK

Sentiment Analysis (SA) is one of the fastest growing research areas in computer science, making it challenging to keep track of all the activities in the area. The extensive use of technologies and astounding flow of data over the years has also aided in the escalation of Big Data business analytics. At the age of Big Data, Big Data Analytics has significantly emerged as a widely accepted computing techniques and the research on Big Data Analytics is still at an early stage. SA is one of the categories of Big Data Analytics and SA on Big Data suffers from different challenging issues related to data collection, preprocessing, and scalable distributed processing. Multi-class classification has become a significant issues of SA on SBD. SA with sarcasm detection has raised growing interests for researchers. Real-time SA for analyzing streaming data is one of the challenges within Big Data processing. This chapter reviews the current literature upon which the theoretical basis of this thesis is built. Based on the interdisciplinary nature of the research there are several related areas covered in this review.

2.1 SA on Conventional Computing Platform

SA is a challenge of the Natural Language Processing (NLP), text analytics and computational linguistics. Fundamentally, SA determines the opinion regarding the object/subject in discussion. Its initial use was made to analyze sentiment based on long texts such as letters, emails and so on. At previous time, most of the applications work enough with traditional SA techniques which run on conventional computing platform. In this case, it is important to choose the right automated SA method to obtain high accuracy in content classification. The well-known approaches are existed to categorized text employed for SA: lexicon-based, machine learning and hybrid approaches. Basically, all of the approaches of sentiment classification classify any given text into positive, negative or neutral sentiment according to the polarity of the contents. Some of the research works to deal with each of the three different SA techniques on conventional computing platform are presented in this subsection.

In [93], the authors found that there is no generalized solution for which technique is best suitable although there are many of machine learning approaches that can be applied to some business domain. People in different organizations try one type of approach and then follow it again and again. There was no standard approach for business prediction to be followed. So there is need to dug up deep for some generalized standard machine learning approaches for different business domains. X. Zhang et al. [119] performed the analysis of the emotional polarity of text as two-classification problems. The VSM model was used to represent a text, and then applied SVM and ELM with kernels to give out the result of classification. Operations to the data set were cleaning, word segmentation, removing stop words, feature selection and classification are the main task of their system. The experiment results showed that ELM with kernels method of emotional polarity analysis of Chinese text is more effective. In [56], the authors exploited four machine learning classifiers for SA using three manually annotated datasets. The classifiers are Naive Bayes, J48, BFTree and OneR. The efficacies of these four classification techniques are examined and compared. They found the Naive Bayes is quite fast in learning whereas OneR seems more promising in generating the accuracy of 91.3% in precision, 97% in F-measure and 92.34% in correctly classified instances.

In [64], L. Zhang et al. presented another variant of this approach: a new entity-level SA method for Twitter is presented. They adopted a lexicon based approach to perform entity-level SA. To improve recall, additional tweets that are likely to be opinionated are identified automatically by exploiting the information in the result of the lexicon-based method. A classifier is then trained to assign polarities to the entities in the newly identified tweets. Instead of being labeled manually, the training examples are given by the lexicon-based approach. Experimental results showed that the proposed method dramatically improved the recall and the F-score, and outperforms the state-of-the-art baselines.

2.2 SA Techniques on Hadoop MapReduce Platform

Today, social media platforms are popular vehicles to study consumer sentiment in a broad and natural way. Due to freely shared of online conversations expressing consumers' thoughts, feelings and opinions about brands and products. Analysis of sentiment in text content is often depended on simple sentiment annotation task while annotators must consider that the sentence is a positive, negative or neutral sentence.

Due to the large amount of social media content, manual sentiment annotation is impractical. Supported by most automated text classification tools, SA, marketers often use it regularly to get fast, scalable and effective way of assessing consumer sentiment. Automated SA received increased attention from both educational institutions and industries, and has become one of the key techniques for managing a huge amount of social media data. In general, automatic SA techniques are used to classify any text documents into predefined categories reflecting the polarity of manual and automated analysis. In this work, SA techniques on Hadoop MapReduce are discussed.

The proposed approach in [123] utilized dictionary based technique for handling SA on MapReduce framework. The approach analyzed the sentiment in the tweets by enriching the AFINN dictionary with semantics using the WordNet. The used semantic relation was the synonymy. Each word in the tweet was assigned a weight and a threshold on the sum of weights was used to determine the polarity of the tweet. The approach achieved good results of classification rate and error rate. In [92], a dictionary based SA technique was proposed based on the Hadoop MapReduce framework. The used dictionary contained sentiment-bearing words. The technique was evaluated using accuracy and execution time. The focus of the research was to develop a scalable technique to handle large datasets efficiently. They concluded that machine-learning techniques provided more accurate results than dictionary-based techniques but machine-learning techniques were not used since they require more time in training and model building.

This paper [83] proposed a MapReduce based Naive Bayes classifier to classify tweets according to their sentiment polarity. The proposed method proves to be highly scalable and gives more accuracy with new large size training dataset we collected, 5% more accuracy than small size bench mark datasets. The MapReduce based training algorithm is all in one job solution to train Naive Bayes classifier which makes it 1.5-1.7 times faster, than the previous works [128]. It also used lesser number of MapReduce jobs to train Naive Bayes as compared to Apache Mahout Naive Bayes, making it and 2.7 to 3.4 times faster than Mahout's Naive Bayes classifier.

2.3 Multi-class SA

As the reimbursed investigation of social media platforms, an ever increasing number of individuals post online messages on various stages to express their sentiments on social issues and offer their feelings on product and services. These expanding quantities of online writings contain countless, which is profitable for governments, organizations and purchasers to settle on alluring choices. For this, numerous examinations on opinion mining and SA have been directed, in which supposition characterization is viewed as a critical research point concentrating on consequently distinguishing the assumption classes of online writings. It is important to bring up that, in the vast majority of the current examinations on slant arrangement, just two sorts of notion classes are recognized, i.e., the estimation class of every content is distinguished as either positive or negative feeling introduction. A few researchers have called attention to that it is over the top straightforwardness that just positive and negative supposition introductions of online writings can be recognized. In the event that numerous conclusion classes of the online writings can be distinguished, they would be profitable for directing progressively exact and increasingly powerful investigation for supporting the choices of governments, organizations and customers. A few examinations on multi-class classification can be discovered, which will be presented in the following subsection.

In paper [66], Multi-tier classification architecture was developed for multi-class SA. To achieve high performance, features were selected by applying various feature selection techniques. 150,000 movie reviews posted on social media were used to train and test the performance of the system. The proposed system runs on traditional analytics platform and classifies the movie reviews into multi-class by applying only supervised machine learning classifiers. Four supervised classifiers (Naive Bayes, SVM, Random Forest, and SGD) were used in the experiments. The result showed that the performance of proposed architecture was significantly improved prediction accuracy over the simple Single-tier model by more than 10%. The authors [71] presented an approach that relies on writing patterns and special unigrams for multi-class SA of Twitter data. They extracted patterns that rely on PoS -Tag of words and then calculate the resemblance degree res (p, t) of each pattern in the training set p to the tweet t. For each tweet, they extracted a 4 set of features; referred to the training set and use Random Forest machine learning algorithms to perform the classification. Training data set contains 21000 tweets which had been manually classified the class.

Since the training data are manually classified the class, no more methods were required to develop the training data sets. In the experiment, the accuracy of the multi-class SA is 56.9%.

2.4 SA with Sarcasm Detection

Sarcasm is a sophisticated form of irony widely used in social networks and microblogging websites. It is usually used to convey implicit information within the message a person transmits. Sarcasm might be used for different purposes such as criticism or mockery. However, it is hard even for humans to recognize. That being the case, the state of the art approaches of SA and opinion mining tend to have lower performances when analyzing data collected from microblogging websites. Maynard et al. [28] showed that SA performance might be highly enhanced when sarcasm within the sarcastic statements is identified. Therefore, the need for an efficient way to detect sarcasm arises. Therefore, recognizing sarcastic statements can be very useful to improve automatic SA of data collected from microblogging websites or social networks. SA refers to the identification and aggregation of attitudes and opinions expressed by Internet users towards a specific topic. Researchers had made a few experiments on sarcasm detection. In this section, the use of sarcasm in tweets are reviewed, and in particular their effect on SA.

S. K. Bharti et al. [96] proposed a Hadoop-based framework that allows the user to acquire and store tweets in a distributed environment and process them for detecting sarcastic content in real time using the MapReduce and Hive. They proposed six algorithms to detect sarcasm in tweets collected from Twitter. The processing time under the Hadoop framework with data nodes reduced up to 66% on 1.45 million tweets. In paper [102], sarcasm-emotion detection method was implemented to classify correctly the emotion of the sentence. They classify phrases in the sentences into the proposed phrase based on the sequence of part-of-speech [59]. The emotion of the propose phrases is determined by the number of words with the emotion included in the phrases. The sarcastic sentiment is determined by judging the emotion of the phrases. Review texts of computer games are used to evaluate the system and this method can determine sarcastic sentences with the precision of 0.79 and the recall of 0.56.

2.5 Real-time SA

As Twitter is one of the most used social media site with huge volumes of data since its start in 2006 with its strength in real-time data, it is considered for SA. Capturing and analyzing the sentiments of these rich tweets with real time data provides a huge opportunity for various businesses and organizations by providing a platform to interact with customers and can yield high benefits in all the fields of research. Real-time analytics has become essential in this digital evolution and it is vital for taking actions or decisions in almost all the sectors. Faster it is available, faster a decision can be made and in some cases, it has the potential to save the lives and prevent the loss of lives. For example, Twitter data is extensively used in Japan on research of earthquakes, Paris attacks, played a major role in US presidential elections etc. and many more. There have been many research works that are being carried out on SA over a decade but real-time SA research are being conducted till now. This section discusses some of the recent Real-time SA techniques.

Apache Spark, one of the large scale data processing frameworks, can offer real time monitoring and analysis. A.Assiri et al. [1] described a distributed solution using Big Data techniques (Spark, Flume) to process the real-time SA using only lexicon based approach. The flume was used for listening Twitter stream data with certain hashtags. When the new batch of tweet arrives at Flume sink, spark streaming consumes the data and loads into memory as RDD objects spread across multiple nodes based on the cluster size. Two types of real data set [2, 8] are used to implement the proposed solution. In order to test the performance of the lexicon-based algorithm, they implemented it in two ways: One way in Java and the second way in Spark. The results showed that the two implementing ways of lexicon-based algorithm achieved the same accuracy. But significant performance improvements in running time of lexicon based algorithm in Spark compared to the implementation of the lexicon based algorithm in Java.

AirSent [29] was a complete Real Time SA system and abled to provide interactive sentiment representations within seconds. The R-based application was effectively used to measure passengers' satisfaction when traveling with an airline, or monitor Twitter users' response to a recent change or event. They used the SMOTE (SyntheticMinorityOver-Sampling) [4] technique on the train set to construct synthetic samples for the minority class (positive) and achieved an observation split of 47:53%.Their system presented the case study of American airline carriers. Moreover,

the proposed architecture could be used in other domains as well, provided there was a training data set available. For the future work, they presented usage of multiple computational units to perform the analysis for reducing the response time of the application, resulting in creating opportunities for commercial use.

2.6 Differences between Existing and Proposed System

The aim of this subsection is to compare some of the existing solutions and proposed system. The comparison can be classified into different categories based on different platforms, strategies and methodologies. For different platforms, the comparison of existing and proposed SA system on conventional computing platform, Big Data Analytics Platform (Hadoop MapReduce) and real-time analytics platform (Spark) are presented. For different strategies and methodologies, the comparison of existing and proposed SA system for multi-class classification and sarcasm detection are presented.

2.6.1 Differences between Existing and Proposed System on Conventional Computing Platform

Most of researchers attempt to create their own lexicon for addressing the coverage of the various domain and build the generalized machine learning model for classification by comparing the different machine learning classifier. Hybrid approaches are developed by exhibiting the accuracy of a machine learning approach and the speed of lexical approach.

The proposed system are developed by the combination of lexicon and machine learning approaches. Existing lexicon based approaches: SentiStrength are used for providing training data set and suitable machine learning classifier is selected by comparing three different machine learning techniques. The selected classifier is used for building the model and predicting new incoming unlabeled data.

2.6.2 Differences between Existing and Proposed System on Hadoop Mapreduce Platform

According to the literature review, the already labeled dataset were used for system evaluation and no more methods were required to develop the training data set in the most of the previous SA system on Hadoop MapReduce framework.

In this work, SA is performed on Hadoop MapReduce by the use of Hybrid approaches. It consists of four modules: Data Collection, Data Preprocessing, Class

Labeling and Sentiment Classification, implemented on four different layers: Data Ingestion Layer, Storage Layer, Data Processing Layer and Analytics Layer. For Data Collection, Real time Twitter data is collected by Apache Flume and filtered the data with keyword: “iphone”. Compared with Kafka, flume is a part of Hadoop ecosystem, which is used for efficiently collecting, aggregating, and moving large amounts of data from many different sources to a centralized data store, such as HDFS or HBase. It is more tightly integrated with Hadoop ecosystem. Kafka is just a general purpose publish-subscribe model messaging system and It is not specifically designed for Hadoop as hadoop ecosystem just acts as one of its possible consumer. In preprocessing step, Selecting tweet text, Removing noisy data such as website links with URL, @username, punctuation, additional white space, hash tags, Removing character repetitions and negation handling and Removing stop words are computed. In Class Labeling module, SentiStrength lexicon based approach implemented with Map Reduce framework, is applied for annotating training data. Mahout Naive Bayes classifier is used for building the classification model and classifying the new incoming unlabeled dataset. For scalability, the evaluation results show that the running time of the system with different volumes of data decreases when adding more nodes into the cluster.

2.6.3 Differences between Existing and Proposed System for Multi-class Classification

Most of Multi-class classification is addressed by using binarization strategies and the approaches are implemented with traditional computing paradigm.

In the proposed system, Multi-tier SA system is developed on Distributed platform (Hadoop MapReduce & Spark) and classified the five classes: positive, negative, strongly_positive, strongly_negative and neutral. Machine learning with Multi-tier architecture is performed in order to address the low performance (in terms of accuracy) of multi-class classification. For performing Machine based classification with multi-tier architecture, three classification models are developed For multi-tier architecture, three prediction models are generated and each model inherits the same configuration as the first model. The comparative results of multi-class classification with single-tier (where single model is developed by training with entire labeled data) and Multi-tier architecture are evaluated. The evaluation results

show that the multi-class classification with Multi-tier architecture improves the classification accuracy.

2.6.4 Differences between Existing and Proposed System for Sarcasm Detection

The previous researches of Sarcasm detection only focus on classifying two classes: sarcasm or non-sarcasm.

The proposed Sarcasm detection approach classified the data into sarcasm and sentiment values. The proposed system is developed by combining rule-based sarcasm detection and machine learning based sarcasm sentiment classification approaches and implemented on Hadoop Mapreduce Platform. Before rule based sarcasm-sentiment detection, POS tagging, dependency parsing, developing sentiment and situation phrase, and deciding sentiment-emotion on the developed phrase are operated. The training data for learning based classifier is generated by applying Rule-based sarcasm detection approach. The classification model is developed with Multi-tier architecture and the new data (unlabeled data) are classified by the developed models. The results show that the proposed system can enhance the performance of SA and opinion mining by detecting sarcastic statements.

2.6.5 Differences between Existing and Proposed System for Real-time SA

Most of the previous Real time SA classified the streaming data by applying only lexicon or machine learning based classifiers. In [47], hybrid approach is implemented on Storm and other types of lexicon and learning based classifiers are performed.

In the proposed system, Real time SA is implemented on Spark by applying hybrid approach and it works with two phrases: Offline training and Online prediction. At the first phrase, the data is collected with batch mode and stored in HDFS. The collected data is preprocessed and then a large amount of training data is provided by applying SentiStrength lexicon based classifier. The last process of Offline training phrase id developing classification model with Multi-tier architecture. At the second phrase, the streaming data is collected by Apache Flume and received and processed the streaming data by Spark Streaming. The new incoming streaming data is classified by using the developed classification models of off-line training. Three different machine learning classifiers (Linear SVC, Naive Bayes and Logistic Regression) are applicable for comparing the performance of the classifiers.

2.7 Chapter Summary

This chapter analyzes the current research works in the area of traditional SA with conventional computing paradigm, SA with parallel computing paradigm: Hadoop MapReduce Platform and Real-time SA. Moreover, multi-class Sentiment Classification techniques and Sarcasm detection methods are reviewed. The aim of this chapter is to highlight the actual requirements and efforts of the research area. This research emphasizes only on developing the scalable multi-class SA system in Big Data environment, and thus various SA approaches: lexicon based approaches, machine learning based approaches and hybrid approaches in existing trends are investigated to justify the design and develop effectively the proposed SA system in Big Data environment.

CHAPTER 3

SENTIMENT ANALYSIS ON TRADITIONAL ANALYTICS PLATFORM AND BIG DATA ANALYTICS PLATFORM

Over the previous decade, the usage of social media (SM) such as Facebook, Twitter, and Tumblr has energetically increased. With SM, people share their opinions on products, services, they rate movies, restaurants or vacation destinations. Social Media such as Facebook or Twitter has made it easier than ever for users to share their views and make it accessible for anybody on the Web. Social media has empowered businesses across the globe to obtain the customers' feedback and design the solutions for their problems in due course of time. Businesses can discover valuable information about their products and services by analyzing social data. With the rapid increase in the amount of social data produced and that is available, the increasing demand of the processing power for solving computational problems has been compelling for innovative ways coping with the need, beyond the level of conventional computing. To accurately analyze social data, the latest trends for researchers from Social Sciences is to understand the potential of Big Data in supplementing traditional research methods and their value in making decisions. In fact, Big Data require review of data analysis techniques with basic methods at every step from data collection and storage to data transformation and interpretation. In particular, the task of collecting and analyzing data, which is at the heart of Big Data Analytics Big Data analytics pipelines, has been challenged in the domain of Social Science. In this chapter, SA is implemented on both traditional analytics platform (conventional computing) and Big Data Analytics Platform (distributed computing).

3.1 Study of Traditional Analytics Platforms

The traditional analytics platforms of the past two decades have chiefly succeeded in providing comprehensive historical reporting and analysis tools. The availability of this functionality is largely due to the underlying data architecture, which consists of a centralized data storage solution [98]. Data is then standardized, cleansed, and transformed in the Enterprise Data Warehouse (EDW) before being pulled into various reports and dashboards to display historical business information, such as quarterly sales or weekly revenue metrics. Traditional computation refers to the

execution of program instructions sequentially, one after the other, on a single processor [12].

The ever increasing demand of the processing power for solving computational problems has been compelling for innovative ways coping with the need, beyond the level of conventional computing. Traditional computing requires complex and expensive hardware and software in order to manage large amount of data. Also moving the data from one system to another requires more number of hardware and software resources which increases the cost significantly. And these computing was not designed to address a number of today's data challenges. The cost, required speed, and complexity of using these traditional computing to address these new data challenges would be extremely high

3.2 Proposed SA System on Traditional Analytics Platform

Traditional applications are designed, developed, and built to perform host-based and fully managed locations by internal IT organizations. In general, the application run on serial computing. The application is covered with state, data and interface references, which can affect the flexibility and agility. In this work, Data Collection and SA is implemented with monolithic design. It consists of four main modules: Data Collection, Preprocessing, Class Labeling by SentiStrength, Learning based Classification Model Development and Tweets classification by using the developed classifier model.

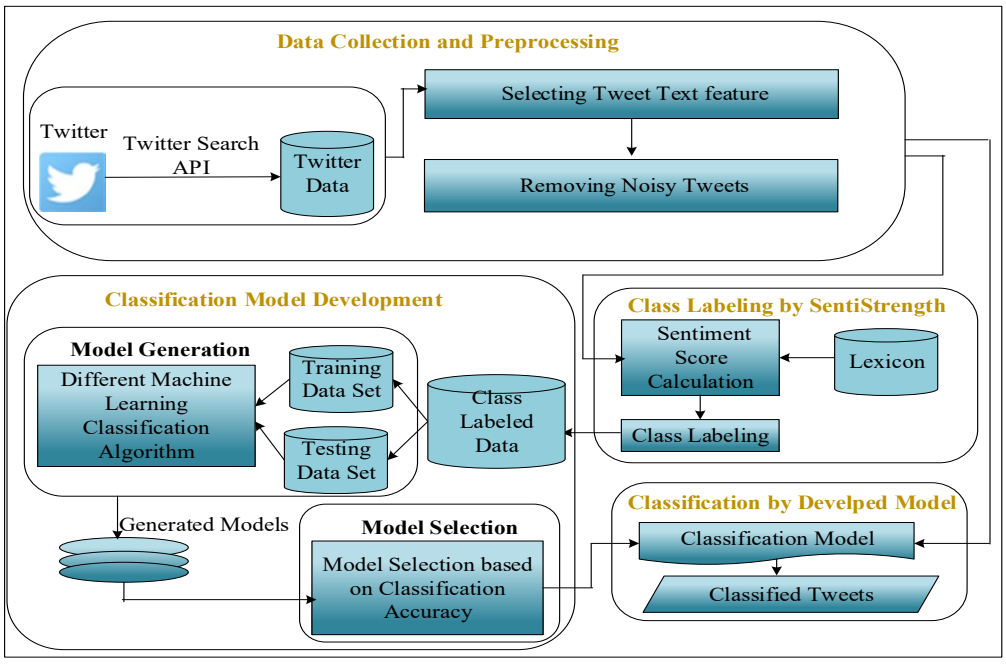


Figure 3.1 Process Flow Diagram of Proposed SA on Traditional Analytics Platform

Figure 3.1 illustrates the processes of SA on traditional analytics platform and the works in this flow are as follows. Apache Flume collects the Tweet stream data through the Twitter streaming API [17]. Among the collected tweets data, tweet text feature with English language is selected. And then Duplicate tweets (Retweets) and noisy data are removed. The preprocessed data is used to calculate sentiment score by using SentiStrength (Lexicon based approach) in order to produce class labeled data. The class label data is used to develop the classification model. To develop the effective classification model, the best model is selected among other different models. The developed model is used to classify the newly incoming raw tweets. Detail processes are described the next sections.

3.2.1 Data Collection and Preprocessing

In order to collect tweets, an application is created and registered for interacting with Twitter API. The application is created by link https: //dev.twitter.com/apps/new. The application name and description are required to create the Twitter application. The Twitter API key, access token, and secret key etc. are created by launching the Twitter application. To connect the application to Twitter, copy the authorized key into the OAuth [14] settings and enable the connection. OAuth is an authorization tool that authorizes third-party applications to access Twitter account without sharing Twitter password. Then the tweet data is collected by using the Twitter API via OAuth setting. In this proposed system, the english tweets are collected and the data is filtered by the keyword “iphone”.

The data collected from Twitter API is tweet stream in JSON (JavaScript Object Notation) format. Table 3.1 shows the sample one record of tweet stream data in JSON format. JSON uses commas to separate attributes, colon to separate the attribute name from the attribute value. There are many tweet attributes and values in one record of tweet stream data. According to the table 3.1, “user location, user name, text”, etc. are attributes and colon separated string and number are its related values. Preprocessing is performed to select values of “text” attribute (tweet texts) and remove noisy characters in tweet texts. The preprocessing process not only simplify the classification task but it also serves to greatly decrease processing cost in the training phase.

Table 3.1 Sample One Tweet in JSON Format

Sample One Tweet in JSON Format
{"filter_level":"low","retweeted":false,"in_reply_to_screen_name":null,"truncated":false, "lang":"en","in_reply_to_status_id_str":null,"id":783222637170262018,"in_reply_to_user_id _str":null,"timestamp_ms":"1475569802931","in_reply_to_status_id":null,"created_at":"Tue Oct 04 08:30:02 +0000 2018","favorite_count":0,"place":null,"coordinates":null,"text":"RT @hankypanty: Updated software on my old iPhone.\nIt's now slow/hanging.\nYes, @Apple - I fell for your scam of forcing me to buy a new pho\u2026","contributors":null,"retweeted_status":{"filter_level":"low","contributors":null,"t ext":"Updated software on my old iPhone.\nIt's now slow/hanging.\nYes, @Apple - I fell for your scam of forcing me to buy a new phone.\n\nAn Android.", "geo":null,"retweeted":false,"in_reply_to_screen_name":null,"truncated":false,"lang":"en ","name":"Apple","indices":[63,69],"screen_name":"Apple","id_str":"380749300"}]},"retwee t_count":2,"in_reply_to_user_id":null,"favorite_count":22,"id_str":"783220031928823808"," place":null,"user":{"location":"Mumbai","default_profile":false,"profile_background_tile":fal se,"statuses_count":34921,"lang":"en","profile_link_color":"0084B4","following":null,"prote cted":false,"favourites_count":2467,"description":"Comedian.Author of 'Under Delhi'. Founder of East India Comedy. Yet another Feminist Indian Male :).\n\nOnly for bookings: pantonfirecomedy@gmail.com.(Direct.)","contributors_enabled":false,"profile_sidebar_borde r_color":"A8C7F7","name":"SorabhPant","profile_background_color":"022330","created_at" :"Sat Dec 05 11:31:12+0000 2009","default_profile_image":false,"followers_count":427379,geo_enabled":true,"follow_re quest_sent":null,"url":http://www.facebook.com/SorabhPant,retweet_count":0,"id_str":"7832 22637170262018","user":{"location":"Mumbai,India","default_profile":true,"profile_backgro und_tile":false,"statuses_count":8246,"lang":"en","favourites_count":0,"profile_text_color":" 333333","verified":false,"description":"A devil's mind with an idle workshop. Coffee connoisseur. Tweets intended for humor, to be taken with a pinch of salt, offensive only to those looking for one.","follow_request_sent":null,"url":null,"utc_offset":19800,"time_zone":"NewDelhi","noti fications":null,"profile_use_background_image":true,"friends_count":134,"profile_sidebar_fil l_color":"DDEEF6","screen_name":"salilmp","id_str":"100968460","listed_count":8,"is_tran slator":false}}

3.2.1.1 Selecting Values of Tweet Text Attribute for SA

There are many tweet attributes (user location, user name, text etc.) in one record of tweet stream data. Among them, the values “text” attribute only expresses twitter users’ feeling and opinion. Therefore only the values of “text” attribute (tweet texts) are selected for developing the proposed system because SA is a type analysis for tracking the mood of the public about a particular product. And the text attribute with English language is focused on the proposed SA. To select values of “text” attribute, the values of “lang” attribute is extracted and checked whether English language or other. If the values of “lang” attribute is English, tweet texts are selected for analysis. Detail procedures of selecting tweet texts are presented in Figure 3.2.

Procedure: Selecting_Tweet_Texts_on _Traditional_Analytics_Platform

Input: Raw Tweets

Output: Tweet Texts

```
1.  Begin
2.  Tweettextfeature ← null
3.  Tweetsindex ← 0
4.  Tweets ← total number of tweets
5.  Tweetslength ← length of the total number of tweets
6.  while(tweetsindex < tweetslength)
7.  Tweetslanguage←tweets[tweetsindex] ['lang']
8.  If(tweetslanguage= ='en')
9.    Tweetsdata←tweets[tweetsindex]
10.   Tweettext ←tweetsdata['text']
11.   Append tweettext to tweettextfeature
12.   Tweetsindex← Tweetsindex+1
13.  endif
14.  endwhile
15.  end
```

Figure 3.2 Procedure of Selecting Values of Tweet Text Attribute on Traditional Analytics Platform

After selecting tweet texts, the texts contains website links with URL and special characters. The sample tweet texts have shown in Table 3.2.

Table 3.2 Sample Tweet Texts

Sample Tweet Texts
If i put my iPhone on "Do Not Disturb" WHHHYYY am I still getting phone calls? I've recorded this... https://t.co/mEcIBmjLN2
Woooohooo my father is going to gift me an iPhone for the success of my research #GreatNewsIsHere
I liked a @YouTube video from @booredatwork https://t.co/lNPwtDWD2z iPhone 7 Review: hmmmmm.....???

3.2.1.2 Removing Noisy Data on Traditional Analytics Platform

The term noisy data is used to describe any piece of information within the tweet that will not be useful while classifying tweet. The noisy data such as character repetitions, website links with URL, @username, punctuation and additional white space may be included in tweet text data. In order to remove two or more repetitions of character, whether the given string contains repetitive letters are needed to be checked. If the repetitive characters are found, those characters are replaced the character itself. Hash tags are replaced with the same word without the hash. For example, #fun is replaced with fun. Website links with URL that start with www.* or http are replaced with "urlinksymbol". In order to easily identify that a user is being referenced, @username is replaced with "usermentionsymbol". Non alphabets are

replaced with space. After removing noisy data of the sample raw Tweets in Table 3.2 the output cleaned data can be seen in Table 3.3.

Table 3.3 Cleaned Tweets by Removing Noisy Data

Cleaned Tweets by Removing Noisy Data
If I put my iPhone on Do not Disturb WHHYY am I still getting phone calls i have recorded this urlinksymbol
Woohoo my father is going to gift me an iPhone for the success of my research GreatNewsIsHere
I liked a usermentionsymbol video from usermentionsymbol urlinksymbol iPhone 7 Review hmm

3.2.2 Class Labeling on Traditional Analytics Platform

Hug amount of twitter data is collected by Apache Flume and manual labeling for those huge amount data is time and labor consuming. Instead of manual labeling, the collected data is labeled the class by SentiStrength. It is a lexicon-based classifier that uses additional (non-lexical) linguistic information and rules to detect sentiment strength in short informal English text. The implementation of the method can be freely used for academic purposes and is available for download in [103].

Procedure: Class_Labeling_ on_ Traditional_ Analytics_Platform

Input: Removed Retweet Texts
Output: Class Labeled Data

1. Begin
2. Perform tokenization
3. Calculate the sentiment strength by using SentiStrength lexicon-based classifier
4. Calculate the total Sentiment Score by combining the strength of positive sentiment(1 to 5) and negative sentiment(-1 to 5)
5. If (score > 1) then print "positive"
6. Else if (score < -1) then Print" negative"
7. Else print "neutral"
8. End

Figure 3.3 Procedure of Class Labeling on Traditional Analytics Platform

In this classifier, a sentiment word list with human polarity and strength judgments is used for classification. It consists of a total number of 58,119 sentiment words. SentiStrength considers linguistic aspects of the passage such as a negating word list, an emoticon list with polarities and slang word list. Emoticon word list consists of 235 emoticons and Slang word list consists of 358 slang words.

For each text, the SentiStrength output is two integers: 1 to 5 for positive sentiment strength and a separate score of 1 to 5 for negative sentiment strength. Here, 1 signifies no sentiment and 5 signify strong sentiment of each type. For example, a message with a score of 3, 5 will have a moderate positive confidence and a good negative confidence. Neutral messages are written as "1, 1". Two scales are used because even a short message can have both positivity and negativity, and the goal is to examine sentiment rather than overall electricity [74]. Finally, the positive and negative strength are combined to classify the sentiment values. Procedures for SentiStrength are presented in Figure 3.3. The sample class labeled data is shown in Table 3.4.

Table 3.4 Sample Class Labeled Data (Ternary Class)

Tweet Texts	Classes
Woohoo my father is going to gift me an iPhone for the success of my research GreatNewsIsHere	positive
I got the iPhone 7 but I can not connect it because I do not have wifi at home and my stupid carrier stopped letting me use personal hotspot	negative
When your boyfriend buys you an iPhone 7 on NationalBoyfriendDay he is a keeper	Neutral

3.2.3 SA with Different Machine Learning Classifiers

To implement SA on traditional analytics platform, classification model is developed by applying three different machine learning algorithm (Naïve Bayes, Random Forest and Linear Regression). To develop the classification model, class labeled data set is divided into two disjoint parts: training and testing data set. The two data set is used to generate (or fit) the model. To develop the effective classification model, the suitable model is selected by comparing accuracy of the generated different classification models. In this system, Naïve Bayes, Linear Regression and Random Forest classifier are used for model generation. Naïve Bayes classifiers are statistical classifiers and can predict class membership probabilities such as probability that a given sample belongs to a particular class [39]. Linear regression is a pretty well-behaved classification algorithm that can be trained as long expected features to be roughly linear and the problem to be linearly separable [4]. Linear regression can also be used in Big Data scenarios since it is efficient and can be distributed. The Random Forest is appropriate for high dimensional data modeling

because it can handle missing values and can handle continuous, categorical and binary data. The bootstrapping and ensemble scheme makes Random Forest strong enough to overcome the problems of over fitting and hence there is no need to prune the trees. Besides high prediction accuracy, Random Forest is efficient, interpretable and non-parametric for various types of datasets [124]. After selecting the best classification Model, it is used to classify newly identified preprocessed Tweets.

3.2.4 Experiments and Results of SA on Traditional Analytics Platform

The performance comparison of three different scalable machine learning techniques are evaluated to select the high performance of learning based technique for the proposed SA. For evaluating the performance, the required dataset are presented in this section. In addition, explanations about evaluation results are also described.

3.2.4.1 Data Set of SA on Traditional Analytics Platform

In the experiments, the collected data from Twitter API is used to evaluate the analysis results. In order to test the functionality of the proposed system, tweets stream data related with iphone product is examined. The data are collected for two months from January to February in 2017. 10000 tweets are utilized as the training datasets and 500 new batch of tweets are applied as the test set for evaluation of the performance of sentiment classification.

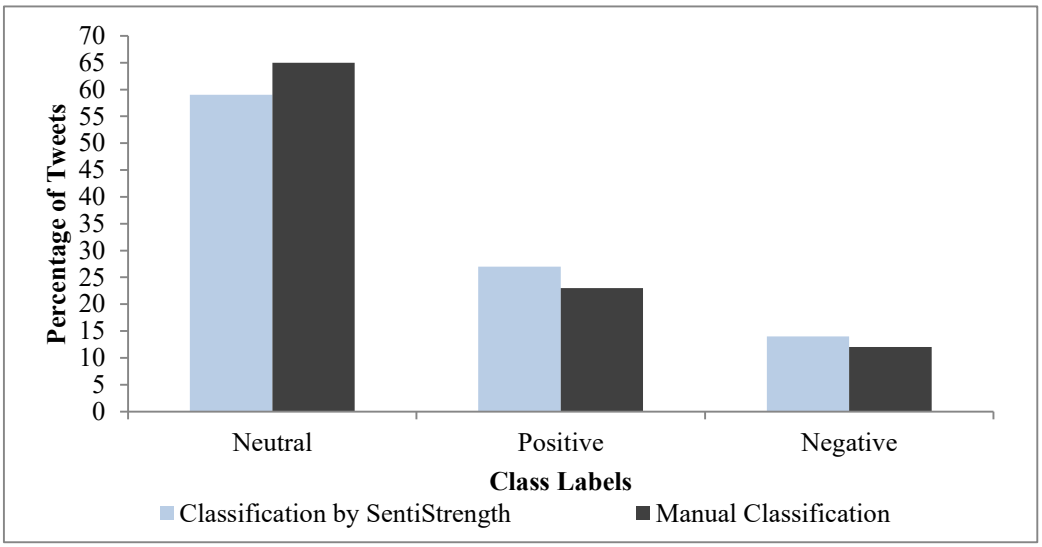
3.2.4.2 Results Discussions of SA on Traditional Analytics Platform

To establish the ground truth, the performance of lexicon based classification is compared with manual classification. 10000 tweets are labeled by manually reading the tweets. Then the same data set is labeled with SentiStrength.jar [103]. To evaluate the performance of Lexicon based classifier (SentiStrength), the classification result from SentiStrength is compared with manual classification on the same data set. Figure 3.4 shows the tweets percentage of SentiStrength lexicon-based classification and manual classification for each class labels. The tweets percentage of classification by SentiStrength for Neutral, Positive and Negative class label are 59%, 27% and 14%. And manual classified Tweets percentage for Neutral, Positive and Negative class label are 65%, 23% and 12%. Therefore, error rate for Lexicon based classifier

is 6% in Neutral class, 4% in Positive class and 2% in Negative class Label. The overall accuracy rate is 88% and error rate is 12%.

Figure 3.4 Tweets Percentage of SentiStrength Lexicon Based Classification and Manual Classification for Each Class Labels

To develop the classification model, class labeled data set is divided into two



disjoint parts: Training and testing data set. The two data set is used to generate (or fit) the model. The training data set size is 10000 records and testing data set contains 500 records. Model selection is evaluated by comparing three different machine learning algorithm (Naïve Bayes, Random Forest and Linear Regression). The experiment is implemented with Java.

Table 3.5 Comparative Results of Three Different Classifiers (Ternary Class)

ML Classifier	Accuracy (%)	Precision (%)	Recall (%)	F-measure (%)
Naïve Bayes	93.4	0.99	0.94	0.95
Random Forest	89.1	0.86	0.89	0.84
LinearRegression	87.3	0.83	0.86	0.82

Table 3.5 shows the accuracy, Precision, Recall and F-measure of each classifier by analyzing the same dataset. As a result, Naïve Bayes Classifier has been selected the (approximately) best classifier with accuracy of 93.4%. Random Forest Linear Regression have been explored with the accuracy of 89.1% and 87.3% respectively. As the Linear Regression is sensitive to outline, the accuracy is decreased. The

accuracy of Random Forest can be affected by the number of decision trees. In this work, the number of decision trees is 100.

The selected model (Naïve Bayes) is used to classify the newly collected Tweets. 500 newly collected tweets are randomly selected and it is used for evaluation. To establish the ground truth, the classified results are compared with manually classified with the same data.

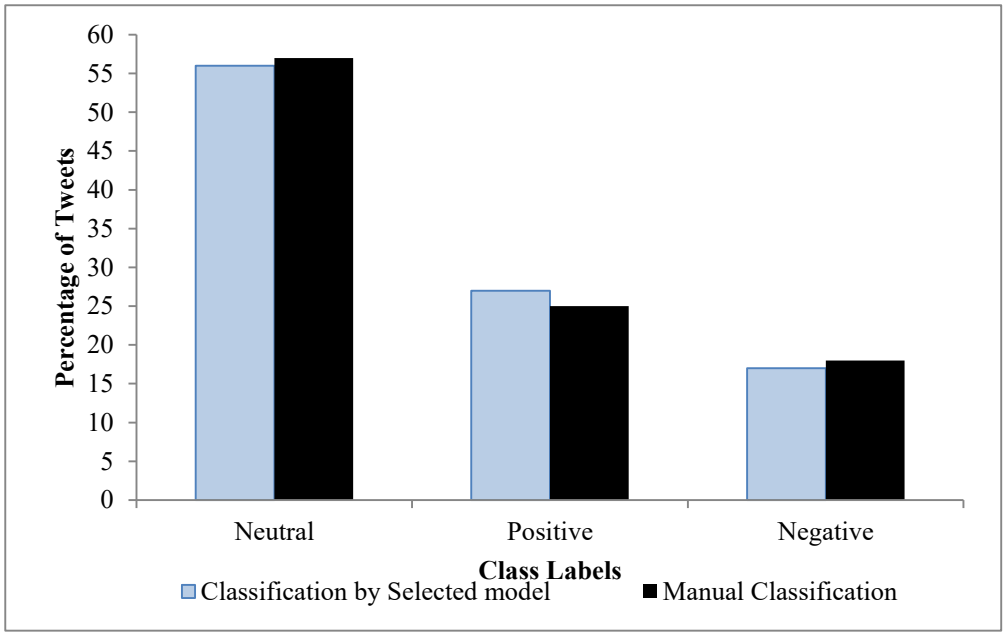


Figure 3.5 Tweets Percentages of Selected Model Classification and Manual Classification for Each Class Labels

Figure 3.5 shows the comparison of selected model classification and manual classification result. According to the results, Neutral, Positive and Negative tweets percentage of selected model classification are 56%, 28% and 16% respectively. Manual classified tweets percentage for Neutral, Positive and Negative class are 57%, 25% and 18%. Therefore, error rate of selected model classifier for Neutral, Positive and Negative class label is 1%, 3% and 2% respectively. Overall accuracy rate is 94% and error rate is 6%. According to the evaluation results, the performance level of proposed system is achieved the promising accuracy.

3.2.5 Limitations of Traditional Analytics Platforms

While users have been able to gain tremendous value from traditional platforms for historical reporting capabilities, more users now require data analyses techniques that require direct access to data without relying on IT specialists. The following

challenges associated with traditional solutions have been highlighted by federal agencies in the analytics space:

- Lack of On-Demand Analysis Capabilities – Today's advanced BI users don't want to wait to get answers to their most complex business problems. More users require self-service capabilities in to relate and analyze specific data sets based on their own understanding, at any time, for any purpose.
- Need for Predictive Analyses – Historical reporting capabilities only provide one piece of the puzzle: insight into what happened in the past. To truly become data driven and forward thinking, businesses are looking to predictive analytics or insight into the future. With predictive models, businesses can use patterns and forecasting to gain actionable next steps based on their data.
- Analysis of Mixed Data Types – Traditional Platforms have largely been focused on structured data, but today, users need the ability to also view and analyze semi-structured, unstructured data, and third party data. The sheer amount of information produced has skyrocketed in recent years, in part due to the Internet of Things (IoT), new data mining techniques, and the proliferation of sensors and other automated data collection tools. Data scientists and advanced BI users now require access to untapped data in various formats, where they have the ability to create their own algorithms and blend data types, and where insights are available on demand for rapid and accurate decision-making.

3.3 SA of Social Big Data on Big Data Analytics Platform

In the proposed SA system, Big Data Analytics Platform is developed to scale up the traditional analytics platform for analyzing large scale social data by using Apache Flume [13], HDFS, MapReduce [45] and Mahout machine learning library [3]. SA is implemented on Big Data Analytics platform (SABDP) and high level architecture of the proposed SA system is illustrated in Figure 3.6. It consists of four processes: data collection, data cleaning and preprocessing, sentiment classification are executed at four layers: Data Ingestion Layer, Storage Layer, Processing Layer, and Analytics Layer.

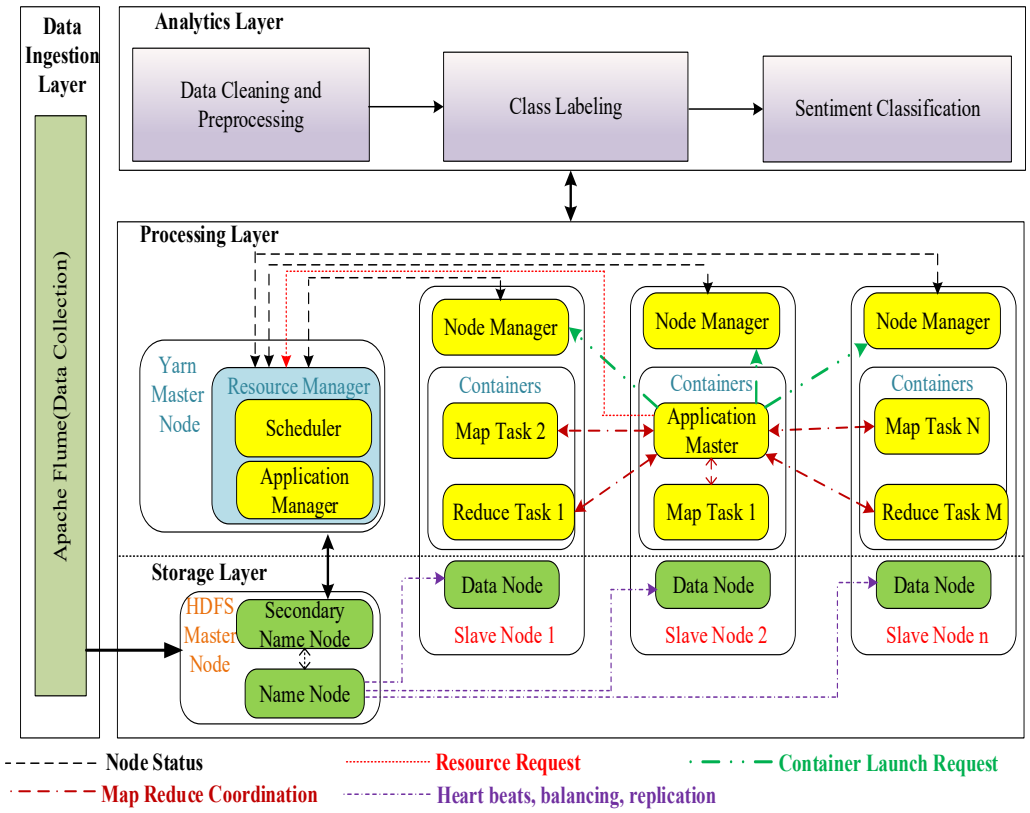


Figure 3.6 High Level Architecture of SA on Big Data Platform (Hadoop MapReduce)

Data collection is performed in the data ingestion layer and the collected data is stored in HDFS. Data cleaning, class labeling and sentiment classification is conducted in the analytics layer. All of the processes from Analytics Layer are executed in distributed manner by using HDFS (storage layer) and MapReduce (processing layer). The process flow of four processes are illustrated in Figure 3.7 and the detail descriptions of each layer are presented the following subsections.

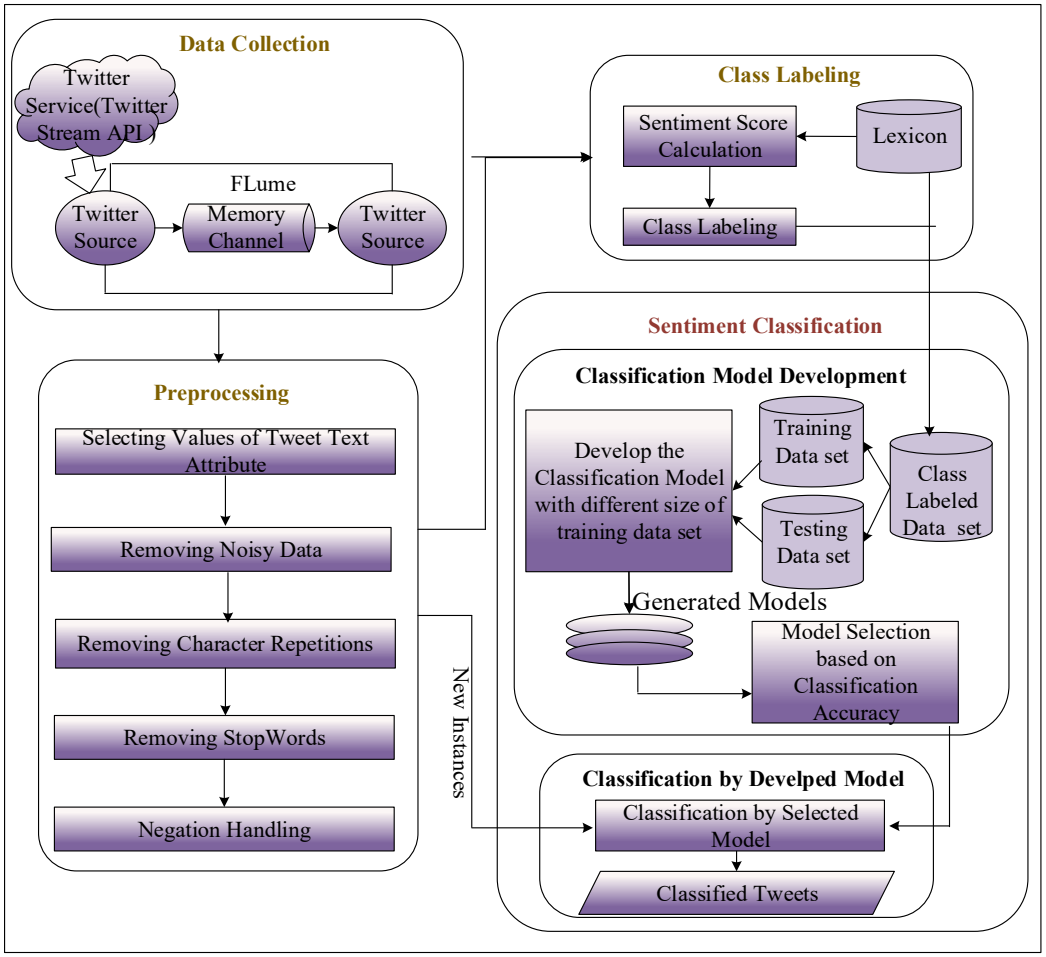


Figure 3.7 Process Flow of SA on Big Data Platform (Hadoop MapReduce)

3.3.1 Data Ingestion Layer

In this layer, tweet stream data is collected and the collected data are ingested to HDFS through the memory channel by using Apache Flume. It has a simple and flexible architecture based on data streaming. It is durable and faulty, with adjustable reliability mechanisms and numerous recovery mechanisms and failures. In the proposed SA system, a large volume of Twitter stream data that is generated from Twitter services is collected by Apache Flume. Twitter services and Flume are vital role of data collection process and the brief explanations are described as follow.

3.3.1.1 Twitter Service

To access Twitter data, there are three different ways: Twitter's Search API; Twitter's Stream API; Twitter's Firehose [17]. In this work, Twitter's Stream API is used for collecting Twitter stream data. It is a push of data as tweets happen in near real-time. With Twitter's Streaming API, users register a set of criteria (keywords,

usernames, location, etc.) and tweets match the criterias are pushed directly to the user. But, it is heavily based on the criteria users request and the current traffic.

3.3.1.2 Apache Flume

Apache Flume is a distributed, reliable, and available service for efficiently capturing, aggregating, and moving large amounts of log data. In this work, Flume is deployed by Twitter Agent in order to ingest Twitter stream data from the Twitter Service (Twitter Stream API), and forward it to HDFS through MemoryChannel. A Twitter Agent is an independent process that receives the data from its source and transports it to its destination. In reality, [54] the process could be more complex where there might be multiple Agents getting data from the same source or an Agent getting data from the output of another Agent. Twitter Agent has three main components – a TwitterSource, a MemoryChannel and a HDFS Sink.

3.3.1.3 Twitter Source

The sources operate events and transport the stream data is to a Memory channel. It is operated by collecting various pieces of data and then the collected data is converted into individual events. Then the events is proposed by Memory channel once at a time, or as a batch. In this work, cloudera Twittersource is used as a data source. Data is limited by key words. The source comes as an event-driven source. Event-driven sources often receive events via mechanisms such as call back or call RPC. The twitter4j library is used in TwitterSource for keeping access from the Twitter Streaming API. Some of the application particular secrets like Consumer Key and Consumer Secret, Access Token, Access Secret are needed to be used for accessing the API. To obtain the application specific secrets, it is needed to create a Twitter application. The Twitter application is created by the following link https://apps.twitter.com/. It generates Consumer key, Consumer secret, Access token, and Access token secret.

3.3.1.4 Memory Channel

The channel operates as a pathway between the TwitterSource and HDFS Sink. Events are added to the channel by TwitterSource, and later removed from the Channel by HDFS Sink. It uses as an in-memory queue to store events until they're

ready to be written to a sink. As the channel holds all events in memory, the channel's capacity and transaction capacity is limited by the "capacity" and "transaction Capacity" parameters in the configuration file. In this work, the channel's capacity is set up to 10000 and the transaction capacity is set up to 100. The "capacity" parameter defines as the maximum number of events stored in the channel at any given time. Channel capacity must be resized to be large enough to store as many events as it is added to the upstream agent. The "transaction Capacity" parameter is defined as the maximum number of events the channel will take from a source or give to a sink per transaction. The number of events vary depending on how many tiers of agents/collectors have been setups. In general though this should probably be equal to whatever you have the batch size set to in the client. This parameter is also a good protection against rogue customers, pushing many events to the source, causing the agent to run out of memory. This parameter forces batches to be limited in size, therefore limiting the number of events per RPC call and preventing simple denial of service (DoS) attacks.

3.3.1.5 HDFS Sink

HDFS Sink, which writes events to the location defined in HDFS in the HDFS Sink configuration, determines the size of the file with roll count parameters and sets a maximum value of 10,000, so each file ends with 10,000 tweets. It also retains the original data format, by setting the file Type to DataStream and setting writing Format to Text. This is done instead of storing the data as a Sequence File or some other format. The file path is defined as that the files will end up in a series of directories for the year, month, day and hour during which the events occur. For example, an event that comes in at 1/2/2017 3:00PM will end up in HDFS at hdfs://hadoop1:8020/user/flume/tweets/2017/1/2/17/. The timestamp is set to true in configuration and which is used by Flume to determine the timestamp of the event, and is used to resolve the full path where the event should end up.

3.3.2 Storage Layer

In Storage Layer, HDFS is used to provide scalable and reliable data storage. HDFS serves master/slave architecture and single NameNode serve as a master server. Name Node performs the functions of the file system namespace, such as opening, closing and renaming files and directories. In addition, it defines the block mapping to DataNodes. DataNodes used to store real data in HDFS. The input file is

divided into one or more blocks and these blocks are stored in a set of DataNodes. Each block size is 64 MB. DataNodes are Responsible for providing service for reading and writing requests from customers. DataNodes also creates blocks, deletions, and replicas according to commands from NameNode.

3.3.3 Processing Layer

In this system, Yarn and MapReduce-2 [32] are located in the processing Layer to process vast amounts of data in-parallel on clusters of commodity hardware in a reliable, fault-tolerant manner. The MapReduce job splits the input data set into independent chunks which are processed by the map tasks in a completely parallel manner. The framework will sort the results of the map which will be entered with the reduction task. Both the input and output of the job are stored in the HDFS framework. The task scheduler supervises them and performs the failed task again. The MapReduce framework contains the ResourceManager. One single core, NodeManager, Slavic per cluster, nodes and MRAppMaster per application. The application identifies input / output positions and provides maps and reduces functions through the use of interface and abstract classes. Input / output positions and other required job parameters include configuration, client tasks, Hadoop jobs, job assignments, and configurations to ResourceManager which is responsible for distributing software and configurations to slaves, scheduling tasks and checking them, providing status and diagnostic information to job-client.

3.3.4 Analytics Layer

Data cleaning and preprocessing, class labeling and sentiment classification are performed in the analytics layer. Sentiment classification is implemented by combining lexicon and learning based approaches. To be scalable classification, the lexicon and learning based classifiers are performed in distributed manner. SentiStrength is used for lexicon based classification and Mahout machine learning library is used for learning based classification. The Mahout machine learning library is specifically designed to use Hadoop for enabling scalable processing of huge data sets. Once the data is stored on the HDFS, Mahout provides the data science tools to automatically find meaningful patterns in big data sets.

3.3.4.1 Data Cleaning on Big Data Analytics Platform

As flume ingests the raw Twitter stream data in nested JSON format and the raw data may contain irrelevant and duplicated data, data cleaning need to be performed. For cleaning the raw data, selecting tweet text features, removing duplicate tweets, removing noisy data are executed. Removing character repetitions, removing stopwords and negation handling are performed during the preprocessing. The preprocessing process not only simplifies the classification task, but also serves to greatly decrease the processing cost in the training phase.

(a) Selecting Values of Tweet Text Attribute

Many tweet attributes are included in one record of Twitter stream data and the proposed SA system focus on tweet texts with English language. For SA, tweet texts is selected among other feature because it expresses twitter users' feeling and opinion. Tweet id feature is also selected to assign as a key and tweet text feature is assigned as a value in Map Reduce process.

Procedure: Selecting_Tweet_Texts_On_Big_Data_Analytics_Platform

```
1.  Input:  rawtweets // rawtweets is JSON file;
2.  Output: tweet_text_features
     1. Begin
     2.   tweetsarray[ ] ← rawtweets.split(" \\n ");
     3.   int i ← 0
     4.   while(i < tweetsarray.length( ))
     5.   Create JSON Object " obj"  of tweetsarray[i]
     6.    tweet_lang ←  String.valueOf (obj.get("lang"))
     7.    if (Tweet_lang.equals("en"))
     8.        Create JSON Object "JSONobj" of tweetsarray[i]
     9.         tweets_id  ←  String.valueOf (JSONobj.get("id"))
     10.        if(!tweets_idno.contains(tweets_id))
     11.         Add tweets_id into tweets_idno
     12.         tweets_text ← String.valueOf(JSONobj.get("text"))
     13.         Convert all of the tweets text to lower case
     14.          if(!tweets_text_feature.contains(tweets_text))
     15.          Add tweets_text into tweet_text_features
     16.              i ← i+1
     17.           endif
     18.                 endif
     19.                   else i=i+1
     20.                     endif
     21.         endwhile
     22.     end
```

Figure 3.8 Procedure of Selecting Values of Tweet Text Attribute on Big Data Analytics Platform

To select the values of tweet_id and tweet_text attributes, the collected tweet data in JSON format is fetched as a JSON object using JSON parser. And “tweets_id”, “tweets_text” and “tweet_lang” are extracted as a string from JSON object. “tweet_lang” is checked whether English or other. If “tweet_lang” is english, tweets_text and tweets_id will be checked as the duplicate or not. If the duplication is detected in tweets, the system will remove the duplicate tweets. If “tweet_lang” is not English language, the system will move to the next line. Detail procedures of selecting values of tweet text attribute are presented in Figure 3.8.

(b) Removing Duplicate Tweets

Twitter Stream data may include many duplicate tweets. To remove duplicate tweets, the extracted “tweets_id” and “tweets_text” is checked whether already stored or not. If the extracted features have not already stored, they are added as the list of tweet data and then analyze the data. The selected sample tweet text and tweet id are shown in Table 3.6.

Table 3.6 Sample Tweet Texts and Tweets Id

Tweets_id	Tweets_text
88841	I liked a @YouTube video from @booredatwork https://t.co/lNPwtDWD2z iPhone 7 Review: hmmmmm.....???

(c) Removing Noisy Data

The term noisy data is used to describe any piece of information within the tweet that will not be useful for the machine learning algorithm to assign a class to that tweet. The noisy data such as character repetitions, website links with URL, @username, punctuation additional white space Replace hash tags with the same word without the hash tags may be included in tweet text. For example, #fun is replaced with fun. Replace the website link with the URL, so link to a website that begins with www. * Or http. Convert @username to "usermentionsymbol by replacing the @username instance found in the tweet with "usermentionsymbol" for the classifier. Non-letters will be replaced with spaces. After removing the noisy data, a sample tweet message can be found in Table 3.7.

Table 3.7 Sample Tweet Texts and Tweet Id that is Removed Noisy Data

Tweets_id	Tweets_text
88841	I liked a usermentionsymbol video from usermentionsymbol urlinksymbol iPhone 7 Review hmm

(d) Removing Character Repetitions

Tweets may contain a character that is repeating more than two times, like the word 'greeeeat'. It is important to replace these words with the original words in order to be able to merge. Otherwise the classifier will assume that they are different words and may be ignored because of the low frequency that occurs.

Procedure : Removing_Character_Repetitions

Input: rawtweets
Output: tweets which is removed two or more character repetitions

```
1. Begin
2. current_tweets ← rawtweets
3. temp ← current_tweets.replaceAll( " ^A-Za-z ", " ")
4. if(temp.length() > 0 && containsRepetitions(temp))
5.       temp ←  replaceRepetitions(temp)
6.       return temp
7. else return current_tweets
8. endif
9. end
```

Figure 3.9 Procedure of Removing Character Repetitions

Figure 3.9 shows the procedure which is the main function for removing character repetitions. This procedure called the other procedure i.e. "containsRepetitions" in order to check whether the repetitions contains or not. If the repetitive characters are found and the character count is more than 2, the procedure: "replaceRepetitions" is invoked for replacing the character itself. Finally the result is returned in this procedure. After removing character repetitions, the sample result can be seen in Table 3.8. Figure 3.10 presents the procedure for checking whether the repetitive characters contain or not in the input data. Firstly, the former and pervious characters are compared by indexing word in one record of tweet. If the previous and current characters are the same, the character is counted. If the quantity of character is two or more, the procedure returns as true. Otherwise, it returns false. Detail procedures of replace repetitions are presented in Figure 3.11. If more than two repetitive characters are found, the procedure replaces the character by deleting the repetitive characters with substring function.

Procedure: Checking_Repetitions

Input: rawtweets
Output: checking results of whether repetitions contains or not//true or false

```
1.  Begin
2.      tweets ← rawtweets
3.      previous_tweets ← tweets.charAt(0)
4.      count ← 0
5.      for(index=1; index← tweets.length(); index++)
6.          current_tweets ← tweets.charAt(i)
7.          if(current_tweets == previous_tweets)then
8.              count ← count+1
9.              if (count > 2)
10.                 return true
11.             else count ← 0
12.             previous_tweets ← tweets.charAt(index)
13.         return false
14.         endif
15.     endif
16.   endfor
17. end
```

Figure 3.10 Procedure of Checking Repetitions

Procedure: Replace_Repetitions

Input: rawtweets
Output: replaced repetitions tweets

```
1.  Begin
2.    tweets ← rawtweets
3.    output_tweets← tweets.substring(0,1)
4.    previous_tweets ← tweets.charAt(0)
5.    found ← false
6.    for(index=1;index<tweets.length( ); index++)
7.      current_tweet ← tweets.charAt(i)
8.      output_tweets ← output_tweets + current_tweets
9.      if (currenttweet == previous_tweets)
10.     if (found==true)
11.       output_tweets ← output_tweets.substring(0,toreturn.length()-1)
12.     else found ← true
13.     endif
14.     else if (found==true)
15.   found ← false
16.   previous_tweets ← tweets.charAt(index)
17.   endfor
18. return output_tweets
19. end
```

Figure 3.11 Procedure of Replacing Repetitions

Table 3.8 Sample Raw Tweets and Cleaned Tweets by Removing Character Repetitions

Raw Tweets	Cleaned Tweets by Removing Character Repetitions
If I put my iPhone on "Do Not Disturb" WHHHYYY am I still getting phone calls? I've recorded this... https://t.co/mEcIBmjLN2	If I put my iPhone on Do not Disturb WHHYY am I still getting phone calls i have recorded this urlinksymbol
RT @ThvGuySpvzz: iPhone 7 Camera soooooo clear you can really see mf's souls lifting outta their body from receiving that bomb head	RT usermentionsymbol iPhone 7 Camera soo clear you can really see mf s souls lifting outta their body from receiving that bomb head

(e) Removing Stopwords

When working with text classification methods, removal of stopwords is a common approach to reduce noise in the data. In this work, not only common stopwords but also stopwords based on classification domain are considered by manually examining the data. For example, domain stopwords contain iphone, apple, mobile, etc.

(f) Negation Handling

Negation handling is one of the factors that significantly affect the accuracy of learning based classifier. For example: the word “good” in the phrase “not good” will be contributing to positive sentiment rather that negative sentiment as the presence of “not” before it is not taken into account. To solve this problem, the simple procedure for handing negations is described in Figure 3.12. It converts words followed by the word "not_" + word.

Procedure: Negation_Handling

Input: tweets

Output: negation handled tweets

```
1. Begin
2.     negated ← false
3.     if the tweets contains negated words:
4.         negated ← true
5.         for each words in tweets:
6.             Replace word to “not_” + word
7.         endfor
8.     endif
9. end
```

Figure 3.12 Negation Handling Procedure

3.3.4.2 Class Labeling on Big Data Analytics Platform (for Ternary Class)

Instead of manually labeling the class, SentiStrength lexicon based classifier is used for the task of annotating the training data for the learning-based classifier. SentiStrength, a lexicon-based classifier, uses additional (non-lexical) linguistic information and rules to detect the sentiment strength in short informal English text. The contextual valence shifter: negation and intensifier are used to evaluate the context sentiment and to solve the context dependent problem by applying NegationWordList and BoosterWordList. There are consists of eight dictionaries: BoosterWordList, EmoticonLookupTable, EmotionLookupTable, EnglishWordList, IdionLookupTable, NegationWordList, QuestionWords and slangLookupTable.

Figure 3.13 shows the procedure for calculating the sentiment score by applying SentiStrength. The procedure is performed in a distributed manner using Map Reduce function. At the Mapper stage, the collected raw data is parsed with the JSONParser in order to select the tweets and tweets_id. After cleaning the data, the total polarity strength is calculated for each sentence by using SentiStrength_Data. If the sentiment score is equal and greater than 1, the output label is “positive. If the sentiment score is equal with 0, the output label is “neutral”. If the score is equal and less than -1 is negative. The Sample class labeled training data is shown in Table 3.3.

Procedure: Class_Labeling_Job (Ternary Class)

Input: k: tweets_idno, v: tweet_text_feature //k:key, v:value
Output: k:tweets_indo, v: class labeled data

```
1.  Function MAPPER(key, values)
2.  while(value € values):
3.      Perform Data Cleaning
4.      Calculate the polarity strength for each sentence
5.      Calculate the total score by combining the strength of sentiment
6.      if (score > 0) then print   “positive”
7.     else if (score < 0) then print ”negative”
8.     else  print   ”neutral”
9.     endif
10.    Emit(tweet & tweet_id, class label)
11. endwhile
```

Figure 3.13 Procedure of Class Labeling on Hadoop MapReduce (for Ternary Class)

3.3.4.3 Sentiment Classification with Scalable Machine Learning Approach

Mahout naive Bayes classifier [3], scalable machine learning algorithm, is conducted to develop the classification model. Naïve Bayes is a learning algorithm

that is frequently employed to tackle text classification problems. Figure 3.14 illustrates the procedure of classification model development. The class labeled data is used as the input data and the input data is preprocessed by applying preprocessing steps. The preprocessed class labeled data set is split into training and testing datasets in order to build the classification model. And then the training and testing data are transformed into the sequence file. As this sequence file consists of key values pairs, class category and tweet_id are set to key and tweets text are set to value. Then Feature generation is performed by using the sparse vector function. In feature generation, TFIDF feature vectors are generated for improving the performance of classification model. The TFIDF vectors are used to train the classification model. Different classification models are developed by applying different size of the training data set. For each developed model, classification accuracy is calculated to select suitable model. The suitable model is selected by comparing the accuracy of classification models with different size of training datasets.

Procedure : Classification_Model_Development (Ternary Class)

Input: class labeled data
Output: classification model

1. Begin
2. Perform data preprocessing
3. Split the input data into training and testing datasets
4. Transform training and testing datasets into sequence file
5. Convert sequence file to TFIDF feature vector
6. Train the classification model using the vector
7. Develop the classification models with different size of training data set
8. Calculate classification accuracy for each developed model
9. Select model based on classification accuracy
10. end

Figure 3.14 Procedure of Classification Model Development (for Ternary Class)

The newly incoming tweets are classified by using selected model. Figure 3.15 shows the procedure for sentiment classification of new instances with distributed manner using MapReduce function. At the Mapper stage, the newly incoming tweets need to be performed data cleaning and preprocessing to be effective classification. Word id and tfidf weights are used to create vectors of the new tweets. With the classifier, the vector score is calculated by applying vector and developed model. To calculate the output results (class category), the "bestscore" is set to "-Double. MAX_VALUE", the "bestcategory_id" is set to "-1" and "category_id" is set to index of vector score. If the vector score is greater than bestscore, bestcategory_id is replaced with category_id. If the "bestcategory_id" is equal with "0", the classifier

classify as “positive”. If the “bestcategory_id” is equal with “1”, the classifier classify as “neutral”. If the “bestcategory_id” is equal with “2”, the classifier classify as “negative”. The naïve bayes algorithm is considered naive because it assumes that the value of a particular feature is independent of the value of any other feature, given the class variable. The Reduce stage outputs the results obtained by the Mapper.

Procedure : Classification_Job (Ternary Class)

Input: k1: tweet_id, v1: Newly Collected Tweet Stream Data //k, k1, k2 : key, v, v1, v2 : values
Output: k: tweets, tweet_id, v: class category //class category : positive || negative || neutral

```
1.  Function Mapper(k1, v1)
2.  while(value Є values)
3.      Performed Data cleaning
4.      Applied the Preprocessing steps for classification
5.      Create vector by using word-id, tfidf value
6.      Calculate vector_score by applying vector and developed model
7.      Assign bestscore to “-Double. MAX_VALUE”, “bestcategory-Id” to
        “-1” and “category-Id” to index of vector_score
8.      if (vector_score > bestscore)
9.          Replace bestcategory-Id to category-Id
10.     if (bestcategory-Id == 0)
11.         Print class category as “ positive”
12.     else if (bestcategory-Id == 1)
13.         Print class category as “ neutral”
14.     else if (bestcategory-Id == 2)
15.         Print class category as “ negative”
16.      end if
17.         Emit(tweets & tweet_id, class category)
18. End while
19. End function
```

Figure 3.15 Procedure of Classification by Developed Model (for Ternary Class)

3.3.5 Evaluation Results of SA on Big Data Analytics Platform (for Ternary class)

To evaluate the performance of the proposed system, beginning from evaluating the performance of lexicon based classifier (SentiStrength). The classification accuracy of Mahout Naïve Bayes classifier is evaluated. In order to test the scalability of this system, the processing time of proposed SA system are measured on different Hadoop cluster nodes. For evaluation, the required system specification and the dataset are presented in this section. In addition, explanations about evaluation results are also described.

3.3.5.1. Experiment Environment of SA on Big Data Analytics Platform (for Ternary Class)

In the experiment, the cluster is composed of 4 computing nodes (VMs). The specifications of devices and necessary software components of this experiment are described in Table 3.9.

Table 3.9 Testing System Specification

Parameters	Specification
Server/Client OS	Ubuntu 14.04 LTS
Host Specification	Intel ® Core i7-3770 CPU @ 3.40GHz, 8GB Memory, 1TB Hard Disk
VMs Specification	4GB RAM, 100 GB Hard Disk
Software Component	- Hadoop 2.7.1 - Flume 1.6 -SentiStrength2 - Mahout 0.10.0

3.3.5.2 Data Set of SA on Big Data Analytics Platform (for Ternary Class)

In order to test the functionality of the proposed system, tweets stream data related with iphone product is examined. The data are collected for two months from January to February in 2017. 200,000 tweet data are utilized as the training datasets and 50000 new batch of tweets are applied as the test set for evaluation of the performance of sentiment classification.

3.3.5.3 Results Discussions of SA on Big Data Analytics Platform (for Ternary Class)

To evaluate the performance of the proposed system, beginning from evaluating the performance of lexicon based classifier (SentiStrength). To establish the ground truth, the evaluation result of lexicon based classifier is compared with manual classification. The comparative results for the performance of lexicon based classifier and manual classification are illustrated in Figure 3.16. In this work, 10,000 tweets are randomly selected from training data for evaluation of lexicon based classifier. Tweets Percentages of classification by SentiStrength for neutral, positive and negative class label are 39, 30 and 31. Manually classified Tweets percentages for neutral, positive and negative class label are 52, 23 and 25. Therefore, error rate for

Lexicon based classifier is 13% in neutral class, 7% in Positive class and 6% in negative class label. The overall accuracy rate is 74% and error rate is 26%.

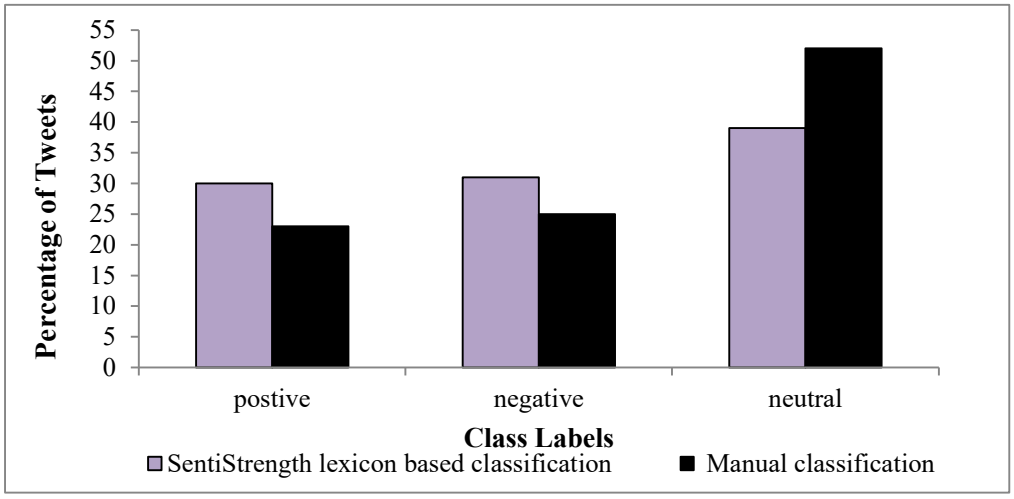


Figure 3.16 Percentage of Tweets on Lexicon based classification and Manual classification (for Ternary Class)

The preprocessing is performed for improving the performance of the classifier. The classification accuracy of Mahout Naïve Bayes classifier using the preprocessing steps is presented in Table 3.10. The results show that the preprocessing can be able to improve the classifier.

Table 3.10 Classification Accuracy of Mahout Naïve Bayes Classifier

Feature	Accuracy(%)
Character Repetition Removal	78.16
Character Repetition Removal + Stopwords Removal	79.38
Character Repetition Removal+ Stopwords Removal + Negation Handling	82.56

For building the classification model, the preprocessed class labeled dataset is divided into two disjoint parts: training and testing dataset. These two datasets is used to generate (or fit) the model. Different training and test dataset can affect the accuracy of classification model. In Figure 3.17 shows the classification accuracy changes while varying the training and testing dataset sizes. The difference between the minimum and maximum classification accuracies when varying the training set size is little about 8%. The highest accuracy of the classification model is 82.56 while training set is 80% and the model is selected for classifying new instances.

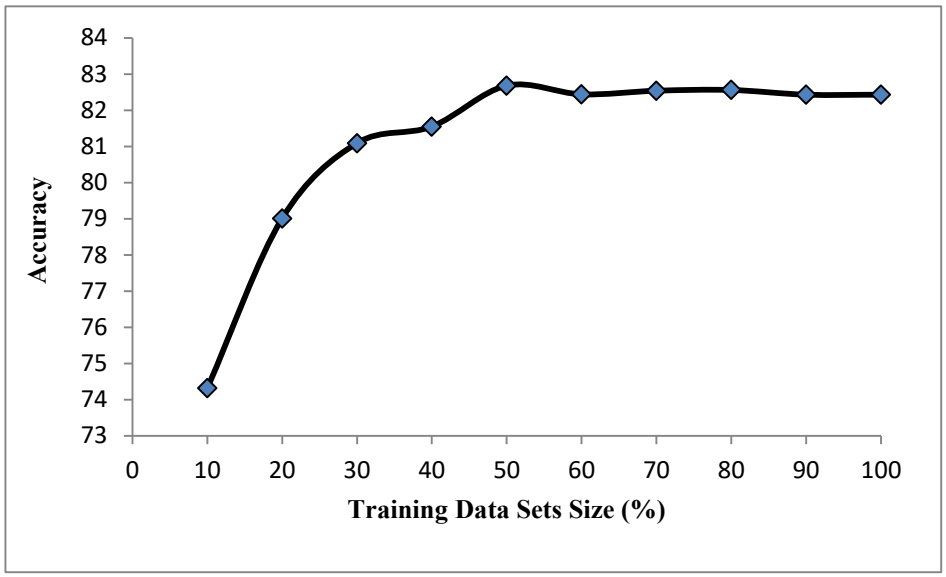


Figure 3.17 Classification Accuracy for Different size of Training Dataset

After selecting the classification model, new test sets are classified by applying the model. Table 3.11 shows the accuracy and F-Measure of the proposed SA system and the overall evaluation results show that the proposed system achieves the promising accuracy by 84.2%.

Table 3.11 Classification Accuracy and F-Measure of Proposed SA system

	Accuracy (%)	F-Measure (%)
Positive	78.5	81.5
Negative	85.3	84.2
Neutral	88.7	83.4
Overall	**84.2**	**83.0**

In order to test the scalability of this system, the proposed SA system runs the job with different number of tweets on different nodes. Each node is developed on each machine. Figure 3.18 shows the processing time of the proposed SA system. In general, the processing time of the system with different volumes of data decreases when adding more nodes into the cluster. In particular, the processing time of data analysis are significantly decrease when adding one node to three nodes and the processing time of data analysis are hardly decrease when adding three nodes to four nodes. According to the results, the processing time is not proportional to the number of nodes due to the latency of IO performance of Hadoop cluster with default configurations.

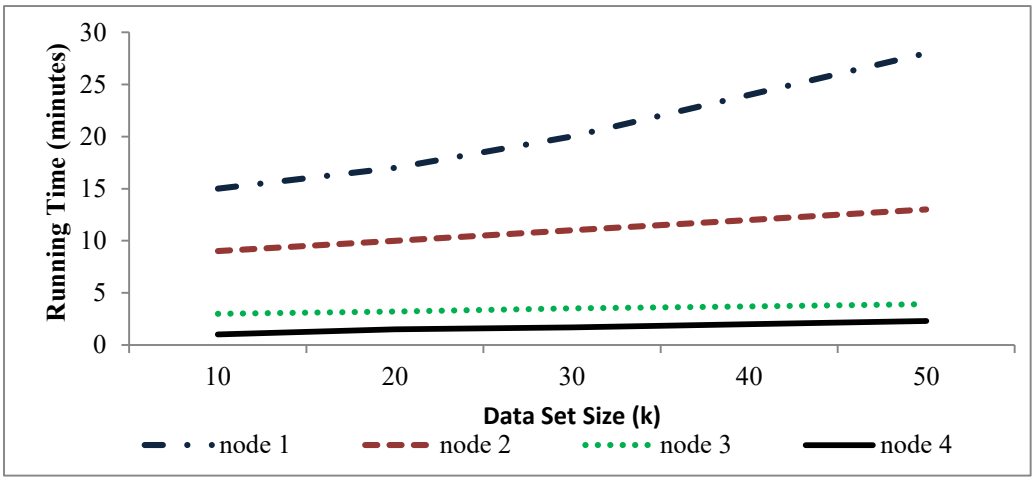


Figure 3.18 Processing Time of the Proposed SA System

3.4 Chapter Summary

The role of SA system on different platforms can be studied in this chapter. Firstly, SA system is implemented on traditional analytics platform. The evaluation results show that the proposed system achieved the promising accuracy. The next one, SA system is implemented on Big Data Analytics Platform as the cost, required speed, and complexity of using these traditional systems. Data collection, data cleaning and preprocessing, sentiment classification are executed at four layers: Data Ingestion Layer, Storage Layer, Processing Layer, and Analytics Layer. The overall accuracy of the proposed SA system are 84.2. For scalability, the evaluation results show that the running time of the system with different volumes of data decreases when adding more nodes into the cluster.

CHAPTER 4

SENTIMENT ANALYSIS WITH SINGLE-TIER AND MULTI-TIER ARCHTECHURE ON BDAP

SA and opinion mining in social networks present nowadays a hot topic of research. However, most of the state of the art works and researches on the automatic SA and opinion mining of texts collected from social networks and microblogging websites are oriented toward the binary classification (i.e., classification into “positive” and “negative”) or the ternary classification (i.e., classification into “positive”, “negative” and “neutral”) of texts. However, to study the opinion of a user, it would be more interesting to go deeper in the classification, and detect the sentiment hidden behind the post. Alternatively, SA based on multi-class classification scheme is oriented towards classification of text into more detailed sentiment labels. Multi-class classification with single-tier architecture where single model is developed and entire labeled data is trained may decrease the classification accuracy. In this work, multi-class classification with multi-tier architecture is developed for increasing performance (in term of accuracy) of multi-class SA problem.

4.1 Multi-class Classification

In machine learning, multi-class or multi-class classification is the problem of classifying instances into one of three classes or more. While some classification algorithms allow more than two classes to be used naturally. However, these things can become polynomial classifiers with a variety of strategies. Multi-class classification should not be confused with multi-label classification, which must predict multiple labels for each instance. Many existing classification techniques can be divided into (i) flat classification and (ii) hierarchical classification.

4.1.1 Flat Classification

The flat classification approach [11], which is the most straightforward one to manage various hierarchical classification problems, comprises of totally overlooking the class pecking order, commonly foreseeing just classes at the leaf nodes. This methodology carries on like a conventional classification algorithm while training and

testing. In any case, it gives a roundabout answer for the issue of various hierarchical classification, since, when a leaf class is appointed to a model, one can think about that all its ancestor classes are additionally verifiably allocated to that example. Be that as it may, this straightforward methodology has the genuine impediment of structure a classifier to segregate among a substantial number of classes (all leaf classes), without investigating data about parent-child class connections present in the class. With more than two classes, this is a standard multi-class classification issue and this methodology is represented in Figure 4.1.

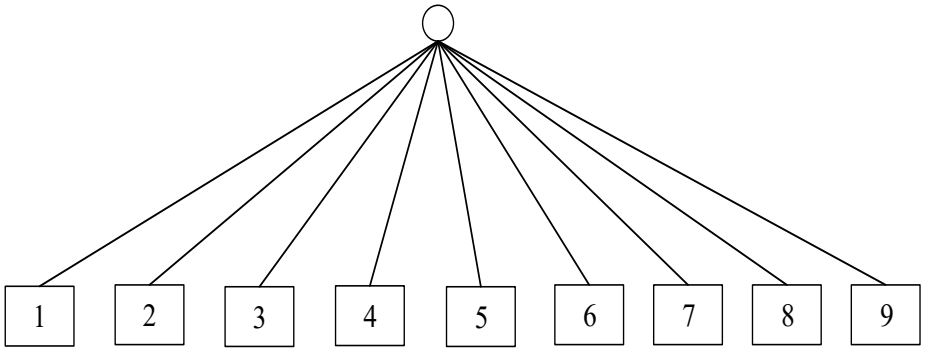


Figure 4.1 Example of a Flat Multi-class Classification Problem

4.1.2 Hierarchical Classification

A hierarchical classification problem can be seen as an issue including huge amount of classes, where a few subsets of classes are more firmly related than others [11]. Distinctive analysts have created diverse strategies to manage multi-class arrangement issues. The most widely recognized are the (I) One-Against-One and the (ii) One-Against-All plans. A less realized methodology comprises of partitioning the issue in a various leveled way where classes which are progressively like each other are gathered into meta-classes, bringing about a binary hierarchical classifier for instance, consider the various leveled order issue in Figure 4.2. The issue comprises of three superclasses, a, b, and c. Every one of these three superclasses contains three subclasses. The given chain of command expresses that the subclasses from one superclass are more unequivocally identified with one another than to subclasses of different superclasses. For example, class 1 is identified with class 2 however not to 4. A class related with an inward hub of a class progressive system speaks to the arrangement of all leaf classes in the tree beneath. Figure 4.3 shows a system of nested dichotomies that is consistent with the n-array hierarchy in Figure 4.2. An

outfit of these trees are found out for expectation on the grounds that the given tree isn't the main conceivable binarization: the first n-array class hierarchy of command can be served to by other binary trees in a legitimate way.

A hierarchical classification problem can be viewed as a problem involving a large number of classes, where some subsets of classes are more closely related than others [11]. Different researchers have developed different methods to deal with multi-class classification problems. The most common are the (i) One-Against-One and the (ii) One-Against-All schemes. A less known approach consists of dividing the problem in a hierarchical way where classes which are more similar to one another are grouped together into meta-classes, resulting in a Binary Hierarchical Classifier As an example, consider the hierarchical classification problem in Figure 4.2. The problem consists of three superclasses, a, b, and c. Each of these three superclasses contains three subclasses. The given hierarchy states that the subclasses from one superclass are more strongly related to each other than to subclasses of other superclasses. For instance, class 1 is related to class 2 but not to 4. A class associated with an internal node of a class hierarchy represents the set of all leaf classes in the tree below. Figure 4.3 shows a system of nested dichotomies that is consistent with the n-array hierarchy in Figure 4.2. An ensemble of these trees are learned for prediction because the given tree is not the only possible binarization: the original n- array class hierarchy can be represented by other binary trees in a valid manner.

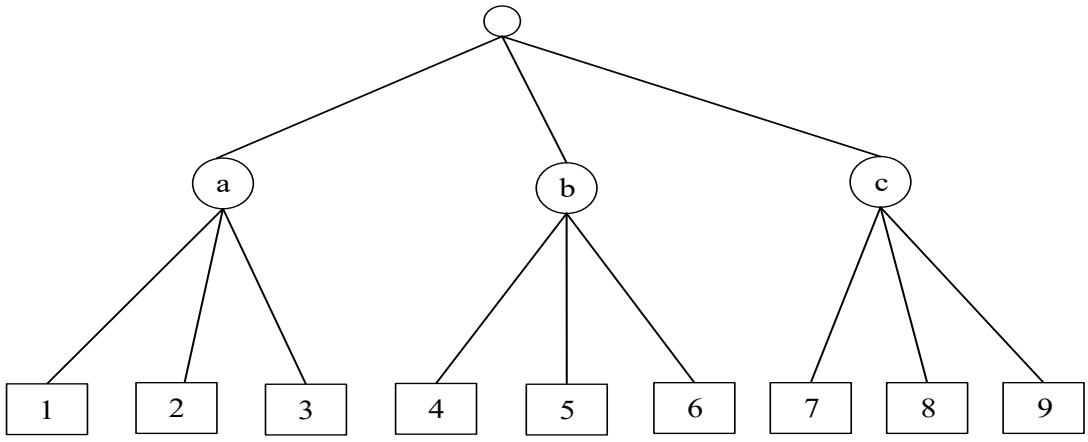


Figure 4.2 A Class Hierarchy Exhibiting Three Classes

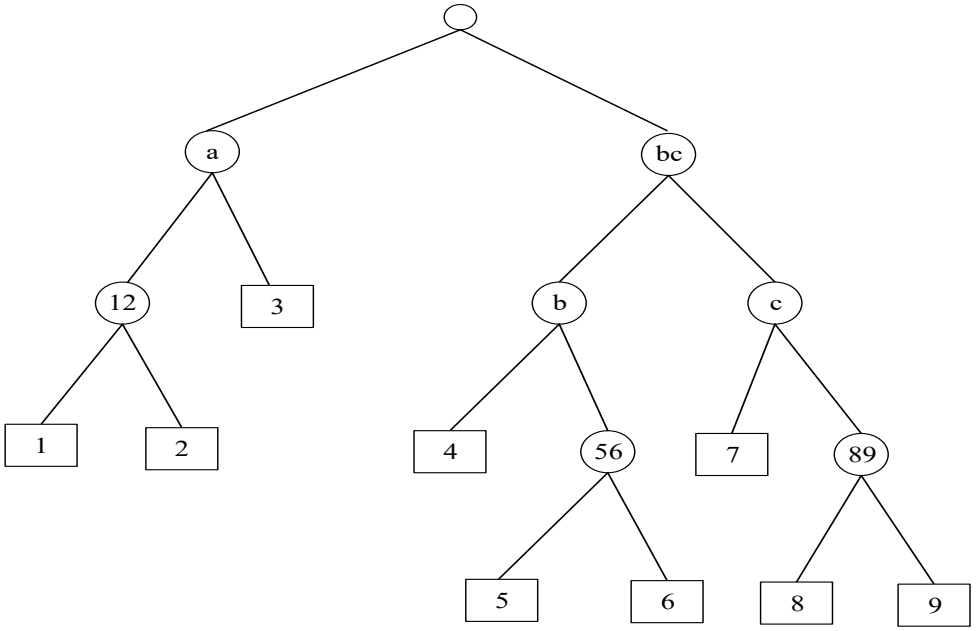


Figure 4.3 Binarization of the n Array Class Hierarchy

4.1.2.1 One Vs Rest

One-versus - Rest (or one-versus - All, OvA or OvR, one-versus - All, OAA) strategies identified with single classifier training per class, with instances of such classes as positive examples and every other sample as negative. This system requires the basic classifier to make a genuine confidence score for decision making, not only a class mark; the names of isolated classes alone can prompt ambiguities, which has various class forecasts for a solitary example. Settling on Decision implies applying all classifiers to the example that is inconspicuous x and k class prediction for which the important classifier reports the highest confidence score.

For example, [30] let D be the set of training examples. For any label I, take elements of D with label I as positive examples and all other elements of D as negative examples. Then, construct a binary classification problem for I. Since there are k possible labels, k binary classi_ers w_1, w_2,...,w_k are produced. Once the k classifiers are learned, the decision is made by winner takes all (WTA) strategy, such that $(x) = argmax_i W_i^T x$. The "score" $W_i^T x$ can be thought of as the probability that x has label i. Graphically, consider the data set with three color labels shown in the Figure 4.4. Using binary classifiers, the black from the rest, blue from the rest, and green from the rest are seperated. The sample decomposition into binary problem are illustrated in Figure 4.5.

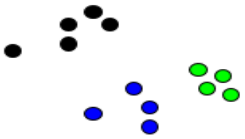
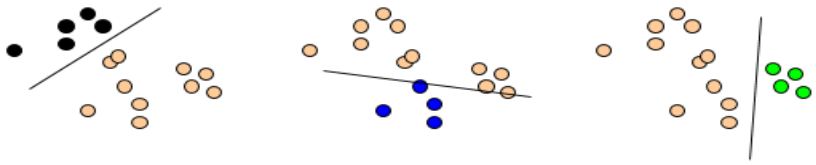


Figure 4.4 Data Sets with 3 Labels



.

Figure 4.5 Decomposed into Binary Problems

The only caveat is that when some points with a certain label are not linearly separable from the other, like shown in the Figure 4.6, this scheme cannot be used. It is not always possible to learn, because it is not always separable. Even though it works well and is the commonly used method, there is no theoretical justification for it.

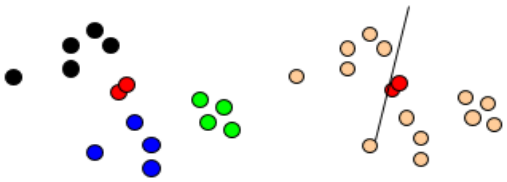


Figure 4.6 Red Points are not Linearly Separable from Other Points

In spite of the fact that this procedure is well known, it is an answer for issues that are experiencing different issues. In the first place, the extent of the certainty esteem may differ between the binary classifiers. Besides, regardless of whether the class appropriation is adjusted in the preparation set, binary classification learners will see an unequal circulation on the grounds that by and large the negative sets they see are bigger than the positive sets.

4.1.2.2 One Vs One

In one-versus - One (OvO) decrease, one trains (K − 1)/2 parallel classifiers for a K-way multi-class issue; every individual will get the example of a couple of classes from the first preparing set, and should figure out how to recognize these two classes. At the forecast time, the casting a ballot arrangement will be utilized: all K (1 - K)/2 classifiers will be connected to concealed example and the class that have the most

noteworthy number of "+1" expectations by the joined classifier. Like OvR OvO experience from equivocalness in certain regions of the information space may get equivalent votes. Assume that there is a separation between every pair of classes using a binary classifier in the hypothesis space. For example, all pairs of labels $\binom{k}{2}$, and for each pair, de_ne a binary learning problem. In each case, for pair (i, j), the positive examples are all examples with label i, and negative examples are those with label j. Now instead of k classifiers as in OvA, $\binom{k}{2}$ classifiers are applied. In this case each label gets k_1 votes, and the decision is more involved, because output of binary classifiers may not cohere. Figure 4.7 presents sample tournament and majority vote to this approach. To make a decision, an option is to classify example x to take label i if i wins on x more often than any j = 1,….., k. Alternatively, a tournament can be done. Starting with n=2 pairs, continue with the winners and go down iteratively.

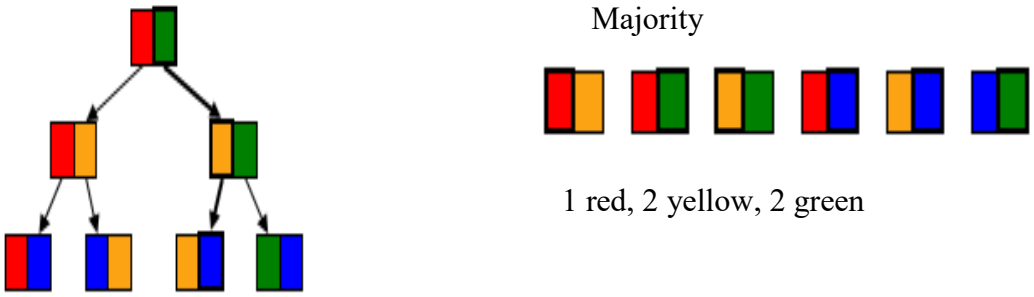


Figure 4.7 Tournament and Majority Vote

4.2 Single-tier SA System on Big Data Analytics Platform (SSABDP)

Sentiment analysis based on multiclass classification scheme is oriented towards classification of text into more detailed sentiment labels. Therefore single-tier Sentiment Analysis system on Big Data Analytics Platform (SSABDP) is proposed to develop multiclass SA. In SSABDP, Big Data Analytics Platform is developed to scale up the traditional analytics platform for analyzing large scale social data by using Apache Flume [13], HDFS, MapReduce [45] and Mahout machine learning library [3]. It consists of four processes: data collection, data preprocessing, class labeling and machine learning based sentiment classification. The process flow of SSABDP is illustrated in Figure 4.8 and the detail procedure of data collection and data preprocessing are already mention in the previous chapter. Therefore the detail procedure of class labeling and machine learning based sentiment classification with single-tier architecture are presented in the following subsections.

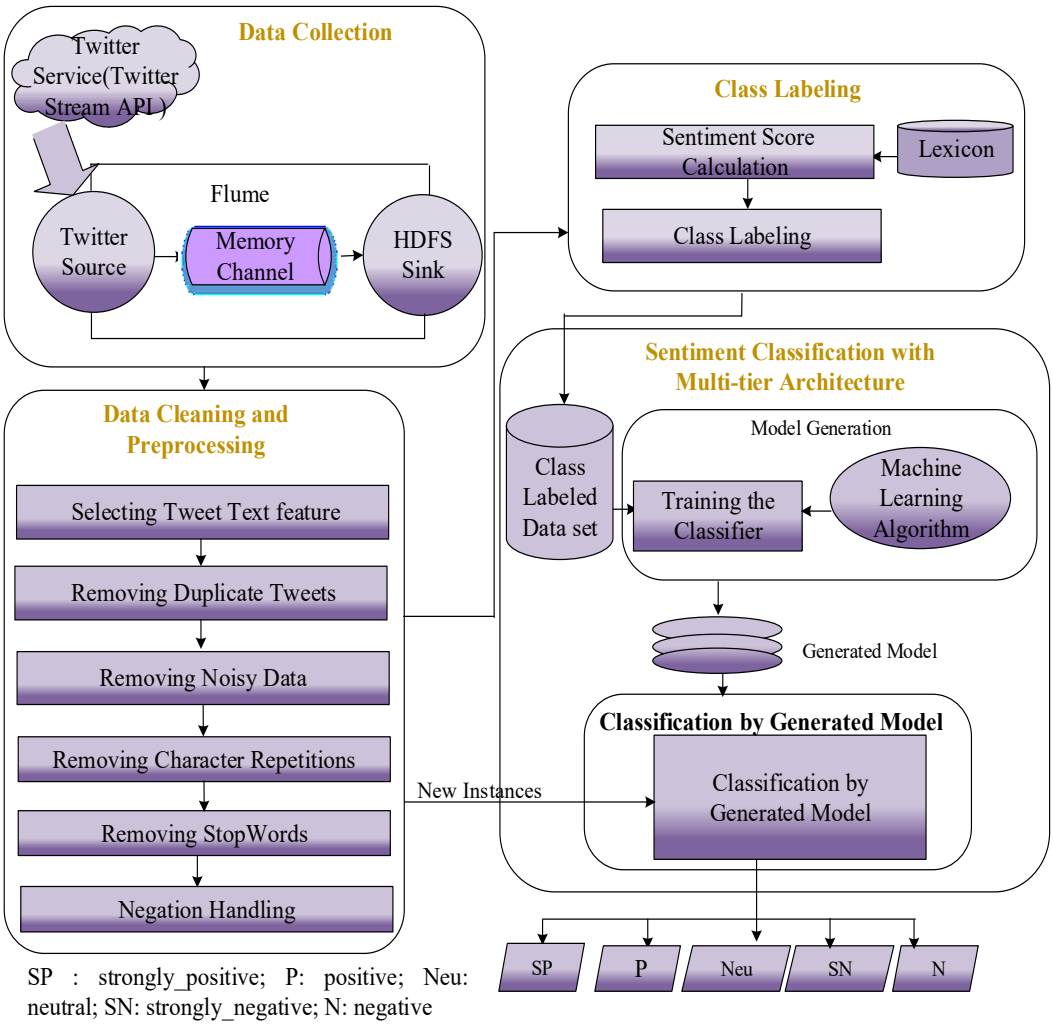


Figure 4.8 Process Flow Diagram of SSABDP

4.2.1 Class Labeling in SSABDP

Instead of manually labeling the class, SentiStrength lexicon based classifier is used for the task of annotating the training data for the learning-based classifier. SentiStrength [103], a lexicon-based classifier, uses additional (non-lexical) linguistic information and rules to detect the sentiment strength in short informal English text.

Procedure: Class_Labeling_Job

Input: k: tweets_idno, v: tweet_text_feature //k:key, v:value
Output: k:tweets_indo, v: class labeled data

```
1.  Function MAPPER(key, values)
2.  while(value € values):
3.      Perform Data Cleaning
4.      Calculate the polarity strength for each sentence
5.      Calculate the total score by combining the strength of sentiment
6.      if (score >0 && score <=2) then print   "positive"
7.     else if (score >2) then Print" strongly_positive"
8.     else if (score < 0 && score >= (-2)) then Print" negative"
9.     else if (score < (-2)) then Print"strongly_negative"
10.    else  print   "neutral"
11.    endif
12.    Emit(tweet & tweet_id, class label)
13. endwhile
```

Figure 4.9 Procedure of Class Labeling (for Multi-class)

Figure 4.9 shows the procedure for calculating the sentiment score by applying SentiStrength. The procedure is performed in a distributed manner using Map Reduce function. At the Mapper stage, the collected raw data is parsed with the JSONParser in order to select the tweets and tweets_id. After cleaning the data, the total polarity strength is calculated for each sentence by using SentiStrength_Data.

For each text, the output score is two integers: 1 to 4 for positive sentiment strength and a separate score of - 1 to - 4 for negative sentiment strength. If the sentiment score is greater than zero and, equal and less than 1, the output label is "positive". If the sentiment score is greater than 1, the output label is "strongly_positive". If the sentiment score is equal with 0, the output label is "neutral". If the sentiment score is less than zero and, equal and greater than (-1), the output label is "negative". If the score is less than -1, the output is "strongly_negative". The Reduce stage outputs the results obtained by the Mappers. The sample class labeled training data is shown in Table 4.1.

Table 4.1 Sample Class Labeled Data (Multi-class)

Tweet Text	Classes
Woohoo my father is going to gift me an iPhone for the success of my research GreatNewsIsHere	positive
I got the iPhone 7 but I can not connect it because I do not have wifi at home and my stupid carrier stopped letting me use personal hotspot	negative
When your boyfriend buys you an iPhone 7 on NationalBoyfriendDay he is a keeper	neutral
Bought an iPhone 7 on eBay and got an iPhone 3 and iPhone 4 not trusting extremely in eBay sellers again	Strongly_negative
When the only reason you bought the new iphone was for the impeccable new camera to take very beautiful picture	Strongly_positive

4.2.2 Machine Learning based Sentiment Classification in SSABDP

Mahout Naive Bayes classifier [3], scalable machine learning algorithm, is conducted to develop the classification model. Naïve Bayes is a learning algorithm that is frequently employed to tackle text classification problems.

Procedure : Classification_Model_Development (Single-tier Architecture)

Input: class labeled data
Output: classification model

1. Begin
2. Perform data preprocessing
3. Split the input data into training and testing datasets
4. Transform training and testing datasets into sequence file
5. Convert sequence file to TFIDF feature vector
6. Train the classification model using the vector
7. Develop the classification models with different size of training data set
8. Calculate classification accuracy for each developed model
9. Select model based on classification accuracy
10. end

Figure 4.10 Classification Model Development Procedure (Single-tier Architecture)

Figure 4.10 illustrates the procedure of classification model development. The class labeled data is used as the input data and the input data is preprocessed by applying preprocessing steps. The preprocessed class labeled data set is split into training and testing datasets in order to build the classification model. And then the training and testing data are transformed into the sequence file. As this sequence file consists of key values pairs, class category and tweet_id are set to key and tweets text are set to value. Then Feature generation is performed by using the sparse vector

function. In feature generation, TFIDF feature vectors are generated for improving the performance of classification model. The TFIDF vectors are used to train the classification model. Different classification models are developed by applying different sizes of the training data set. For each developed model, classification accuracy is calculated to select suitable model. The suitable model is selected by comparing the accuracy of classification models with different sizes of training datasets.

Procedure : Classification_Job (Single-tier Architecture)

Input: k1: tweet_id, v1: Newly Collected Tweet Stream Data //k, k1, k2 : key, v, v1, v2 : values

Output: k: tweets, tweet_id, v: class category //class category : positive || negative || neutral

```
1.  Function Mapper(k1, v1)
2.  while(value € values)
3.      Performed Data cleaning
4.      Applied the Preprocessing steps for classification
5.      Create vector by using word-id, tfidf value
6.      Calculate vector_score by applying vector and developed model
7.      Assign bestscore to "-Double. MAX_VALUE", "bestcategory-Id" to
        "-1" and "category-Id" to index of vector_score
8.      if (vector_score > bestscore)
9.          Replace  bestcategory-Id to category-Id
10.     if (bestcategory-Id == 0)
11.         Print class category as " positive"
12.     else if (bestcategory-Id == 1)
13.         Print class category as " neutral"
14.     else if (bestcategory-Id == 2)
15.         Print class category as " negative"
16.      else if (bestcategory-Id == 3)
17.         Print class category as " strongly_positive"
18.      else if (bestcategory-Id == 4)
19.          Print class category as "strongly_negative"
20.      end if
21.     end if
22.         Emit(tweets & tweet_id, class category)
23. End while
24. End function
```

Figure 4.11 Sentiment Classification Procedure (Single-tier Architecture)

The newly incoming tweets are classified by using selected model. Figure 4.11 shows the procedure for sentiment classification of new instances with distributed manner using MapReduce function. At the Mapper stage, the newly incoming tweets need to be performed data cleaning and preprocessing to be effective classification. Word id and tfidf weights are used to create vectors of the new tweets. With the

classifier, the vector score is calculated by applying vector and developed model. To calculate the output results (class category), the "bestscore" is set to "-Double. MAX_VALUE", the "bestcategory_id" is set to "-1" and "category_id" is set to index of vector score. If the vector score is greater than bestscore, bestcategory_id is replaced with category_id. If the "bestcategory_id" is equal with "0", the classifier classify as "positive". If the "bestcategory_id" is equal with "1", the classifier classify as "neutral". If the "bestcategory_id" is equal with "2", the classifier classify as "negative". If the "bestcategory_id" is equal with "3", the classifier classify as "strongly_positive". If the "bestcategory-Id" is equal with "4", the classifier classify as "strongly_negative". The naïve bayes algorithm is considered naive because it assumes that the value of a particular feature is independent of the value of any other feature, given the class variable.

4.2.3 Experiments and Results of SSABDP

To evaluate the performance of SSABDP, the accuracy of single-tier sentiment classification is compared with only lexicon based classification. For evaluating the performance, the required system specification and the dataset are presented in this section. In addition, explanations about evaluation results are also described.

4.2.3.1 Experimental Environment of SSABDP

In the experiment, the cluster is composed of 4 computing nodes (VMs). Each cluster node run on each VM. The specifications of devices and necessary software component of the proposed system are described in table 4.2.

Table 4.2 Testing System Specification of SSABDP

Parameters	Specification
Server/Client OS	Ubuntu 14.04 LTS
Host Specification	Intel ® Core i7-3770 CPU @ 3.40GHz, 8GB Memory, 1TB Hard Disk
VMs Specification	4GB RAM, 100 GB Hard Disk
Software Component	- Hadoop 2.7.1 - Flume 1.6 -SentiStrength2 - Mahout 0.10.0

4.2.3.2 Data Set of SSABDP

In order to test the functionality of the proposed system and prove the achieved results with promising accuracy, tweets stream data related with iphone product is examined. The data are collected for two months from January to February in 2017. 200,000 tweet data are utilized as the training datasets and 50000 new batch of tweets are applied as the test set for evaluation of the performance of sentiment classification.

4.2.3.3 Evaluation Results of SSABDP

To evaluate the performance of SSABDP, the accuracy of single-tier sentiment classification is compared with only lexicon based classification. Table 4.3 illustrates the comparative results of single-tier sentiment classification and only lexicon based classification (SentiStrength).

Table 4.3 Comparative Results of SSABDP and Only Lexicon based Classification

	SSABDP		Only Lexicon based Classification	
	Accuracy(%)	**Overall Accuracy (%)**	**Accuracy(%)**	**Overall Accuracy (%)**
P	78.82	75.43	78.57	73.20
SP	69.54		68.48	
N	72.93		70.63	
SN	68.57		67.61	
Neu	82.79		80.72	

The results show that the classification accuracy of SSABDP is higher than only lexicon based approach. As the misclassification error of lexicon based classifier can be covered by proposed SA in which the combination of machine learning and lexicon based classifier.

4.3 Multi-tier SA System on Big Data Analytics Platform (MSABDP)

Sentiment analysis based on multiclass classification scheme is oriented towards classification of text into more detailed sentiment labels. However, multiclass classification with single-tier architecture where single model is developed and entire labeled data is trained may decrease the classification accuracy. Therefore multi-tier Sentiment Analysis system on Big Data Analytics Platform (MSABDP) is proposed to achieve high level performance of multiclass classification.

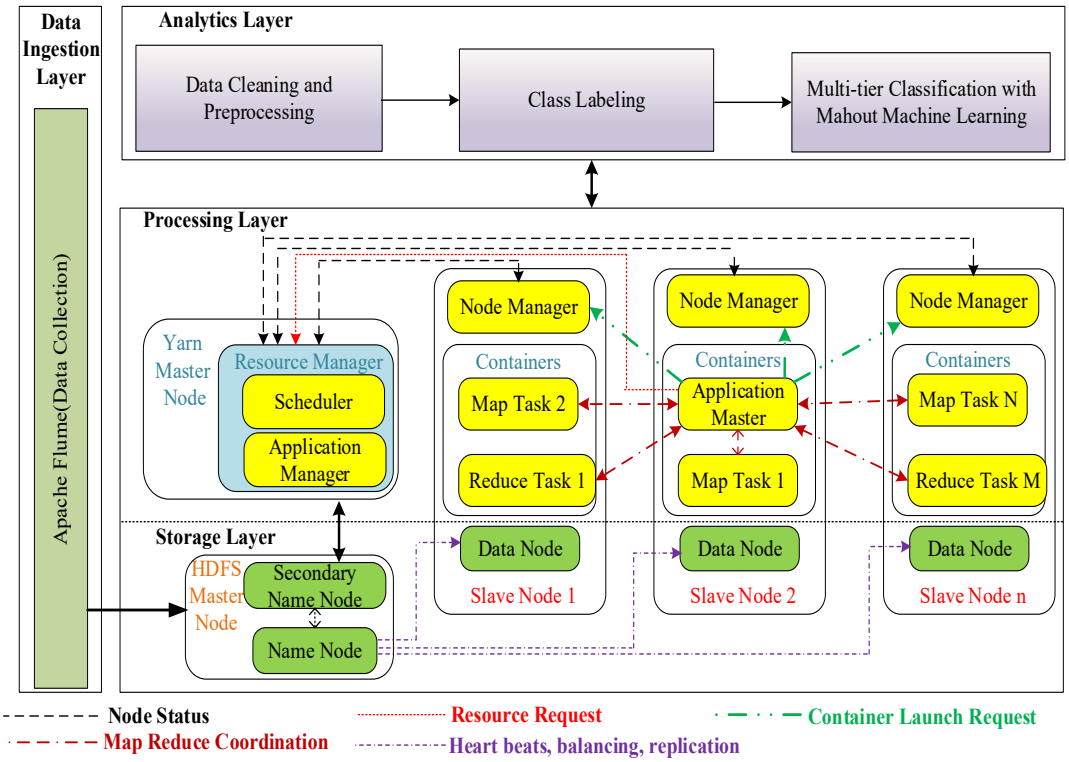


Figure 4.12 High Level Architecture of MSABDP (Hadoop Map Reduce)

In MSABDP, Big Data Analytics Platform is implemented to scale up the traditional analytics platform for analyzing large scale social data by using Apache Flume, HDFS, MapReduce [45] and Mahout Machine learning library [3]. MSABDP is implemented at four layers: Data Ingestion Layer, Storage Layer, Processing Layer, and Analytics Layer. In Data Ingestion Layer, Apache Flume is used to collect Twitter stream data and the collected data is ingested to HDFS through the memory channel. HDFS, scalable and reliable data storage, is located in Storage Layer. Yarn and MapReduce-2 are located in the Processing Layer to process vast amounts of data in-parallel on clusters of commodity hardware in a reliable, fault-tolerant manner. Data cleaning and preprocessing, class labeling and sentiment classification with multi-tier architecture are implemented in Analytics Layer. All of the processes from

Analytics Layer are executed in distributed manner by using HDFS and MapReduce. The sentiment classification with multi-tier architecture is implemented by using Mahout Machine learning library. High level architecture of the MSABDP is illustrated in Figure 4.12.

MSABDP consists of four modules: data collection, data cleaning and preprocessing, class labeling and sentiment classification with multi-tier architecture. The process flow of MSABDP is presented in Figure 4.13.

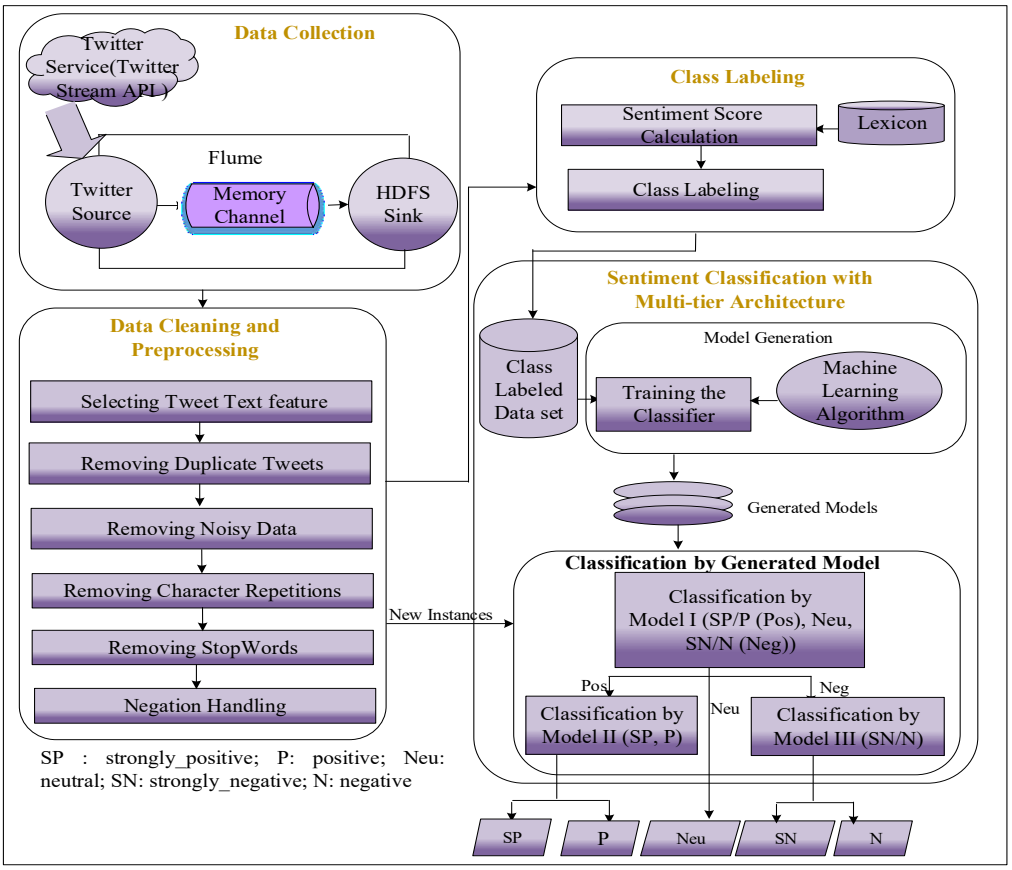


Figure 4.13 Process Flow Diagram of MSABDP

4.3.1 Class Labeling in MSABDP

Instead of manual labeling the class, SentiStrength lexicon based approach is used for annotating the training data of the learning-based classifier. SentiStrength [103], a lexicon-based classifier, uses additional (non-lexical) linguistic information and rules to detect sentiment strength in short informal English text. The SentiStrength consists of eight dictionaries: BoosterWordList, EmoticonLookupTable, EmotionLookupTable, EnglishWordList, IdionLookupTable, NegationWordList, QuestionWords and slangLookupTable. EmotionLookupTable consists of 2546

sentiment words with their strength. Some words include Kleene star stemming (e.g., ador*). A booster word list consists of 28 words and their strength of sentiment. List of idioms used to identify the sentiment of a few common phrases. This overrides individual sentiment word strengths. The emoticon list consists of 116 common emoticon words and their polarity strength (-1 or 1). Negative sentiment is ignored in questions. For each message, SentiStrength will give a positive feeling from 1 to 5 and a negative score from -1 to -5. The detail procedure of class labeling is already mention in section 4.2.1.

4.3.2 Sentiment Classification with Multi-tier Architecture

In order to implement the multi-tier classification, there are two main parts: classification model development and classification by developed model. The classification model is developed with multi-tier architecture and the new data (unknown data) are classified by the developed models.

4.3.2.1 Classification Model Development in MSABDP

Developing the classification model is a vital part of the MSABDP. Mahout naive Bayes classifier, scalable machine learning algorithm, is conducted to develop the classification model. Naïve Bayes is a learning algorithm that is frequently employed to tackle text classification problems. To develop the model using Mahout Naïve Bayes, the input data is transformed into the sequence file. As this sequence file consists of key value pairs, class category and tweets_id are set to key and tweets are set to value. Feature generation is performed by creating sparse vector. In feature generation, TFIDF feature vectors are used for improving the performance of classification models. For multi-tier architecture, three classification models are developed and each model inherits the same configuration as the first model.

Procedure : Classification_Model_Development (Model I)

Input: Class Labeled data (Pos, Neu, Neg)
Output: Classification model I

1. Begin
2. Transform input data to Sequence File
3. Convert Sequence File to TFIDF feature vector
4. Develop classification model I by using the vector
5. End

Figure 4.14 Procedure of Classification Model Development (Model I)

Figure 4.14 illustrates the procedure for developing classification model I. To develop the first model (Model I), class labeled datasets which class category is "P" and "SP" is identified as "Pos" and the class category which class category is "N" and "SN" is identified as "Neg". In model I, all of the labeled datasets (class category is Pos, Neu, Neg) are used to train the classifier and new test instance (unlabeled data) is classified into three classes (Pos, Neu, Neg). Neutral sentiment data is still labeled as "Neu" in the Model I and the neutral class are directly appended to the final classification result. The new instances which are classified as "Pos" are going to Model II and other instances are going to Model III.

Procedure : Classification_Model_Development (Model II)

Input: Class Labeled data (P,SP)
Output: Classification model II

1. Begin
2. Transform input data to Sequence File
3. Convert Sequence File to TFIDF feature vector
4. Develop classification model II by using the vector
5. End

Figure 4.15 Procedure of Classification Model Development (Model II)

In model II, the labeled datasets which class category is "P" and "SP" are used to train the classifier and the test instances which class category is "Pos" are divided into two classes: "positive" and "strongly_positive". The procedure for developing Model II is illustrated in Figure 4.15.

Procedure : Classification_Model_Development (Model III)

Input: Class Labeled data (N,SN)
Output: Classification model II

1. Begin
2. Transform input data to Sequence File
3. Convert Sequence File to TFIDF feature vector
4. Develop classification model III by using the vector
5. End

Figure 4.16 Procedure of Classification Model Development (Model III)

Figure 4.16 shows the procedure for developing classification model III. In model III, the class labeled datasets which class category is "N" and "SN" are used to train the classifier and new test instances which class category is "Neg" are divided into two classes: "negative" and "strongly negative".

4.3.2.2 Classification by Developed Model in MSABDP

The newly incoming tweets are classified by using developed model. The procedure of sentiment classification for new instances is presented in Figure 4.17. For classification, naïve bayes classifier uses probabilities to decide which class best matches for a given input text. Word id and tfidf weight are used to create vector for the new tweet. The naïve bayes classiier is classified by using the vector.

Procedure : Classification_Job (Multi-tier Architecture)

Input: New Collected Tweet Stream Data
Output: k : tweet, tweet_id, v: class category

```
1.  Function Mapper(k1, v1)
2.  while(value € values):
3.      Perform Data cleaning and preprocessing
4.      Create feature vector by using wordid, tfidf value
5.      Classify feature vector of the new instances by applying Model I
6.      Print classified results("Pos", "Neu", "Neg")
7.      If (results= = "Pos")
8.          Classify feature vector of tweet(class category is "Pos") by applying
            Model II
9.          Print classified results("positive", "strongly_positive")
10.     End if
11.     Else If (results= = "Neg")
12.         Classify feature vector of tweet(class category is "Neg") by
            applying Model III
13.         Print classified results("negative", "strongly_negative")
14.     End if
15.     Else Print classified results("neutral")
16.         Emit(tweet & tweet_id, class category)
17.     End if
18. End while
19. End function
```

Figure 4.17 Procedure of Sentiment Classification with Multi-tier Architecture

For model I, the score of three class label is calculated. The "bestscore" is set to "-Double.MAX_VALUE". The "bestcategoryId" is set to "-1" and "categoryId" is set to index of classification result. If the indexed score is greater than "bestscore", bestcategoryId is replaced with categoryId. If the "bestcategoryId" is equal with "1", the classifier classify as "Pos". If the "bestcategoryId" is equal with "0", the classifier classify as "Neu". Otherwise, it classify as "Neg". For model II, the score of two class label is calculated. If the "bestcategoryId" is equal with "1", the classifier classify as "strongly_positive". Otherwise, the classifier classify as "positive". For model III, the score of two class label is calculated. If the "bestcategoryId" is equal

with “1”, the classifier classify as “strongly_negative”. Otherwise, the classifier classify as “negative”. As the result combination is not needed, the Reduce stage outputs the results obtained by the Mapper function. The algorithm is considered innocent because it assumes that the value of that particular feature does not depend on the value of any other feature assigned to the class variable. Laplace smoothing is performed with value of α set to 1.

4.3.3 Experiments and Results of MSABDP

To evaluate the performance of MSABDP, multi-tier multi-class classification is compared with single-tier multi-class classification. In order to test the scalability of this system, the processing time of MSABDP are measured on different Hadoop cluster nodes. For evaluating, the required system specification and the dataset are presented in this section. In addition, explanations about evaluation results are also described.

4.3.3.1 Experiment Environment of MSABDP

In the experiment, the cluster is composed of four computing nodes (VMs). Each cluster node run on each VM. The specifications of devices and necessary software component of MSABDP are presented in Table 4.4.

Table 4.4 Testing System Specification of MSASDP

Parameters	Specification
Server/Client OS	Ubuntu 14.04 LTS
Host Specification	Intel ® Core i7-3770 CPU @ 3.40GHz, 8GB Memory, 1TB Hard Disk
VMs Specification	4GB RAM, 100 GB Hard Disk
Software Component	- Hadoop 2.7.1 - Flume 1.6 - SentiStrength 2 - Mahout 0.10.0

4.3.3.2 Data Set of MSABDP

In order to test the functionality of the proposed system, Twitter stream data related with IPHONE mobile product are examined. 200,000 tweets (from June to July) are collected as the training datasets and 50000 new batch of tweets are collected as the test set for evaluation of the performance of multi-tier classification.

5,000 tweets, randomly selected from collected data, evaluate the performance of SentiStrength lexicon based classification.

4.3.3.3 System Evaluations and Results Discussion of MSABDP

To establish the ground truth, SentiStrength lexicon based classifier is compared with manual classification. Figure 4.18 illustrates the comparative results of lexicon based classification and manual classification. The results show the tweets percentage of classification with SentiStrength based approach for positive, strongly_positive, negative, strongly_negative, neutral class labels are 20, 8, 24, 12 and 36. And manual classified Tweets percentages for positive, strongly_positive, negative, strongly_negative, neutral are 25, 11, 20, 16 and 28. Therefore, error rate for lexicon based classification is 6% in positive, 5% in strongly_positive and 4% in negative, 4% in strongly_negative, and 8 % in neutral. Therefore, the overall accuracy is 73%and error rate is 27%.

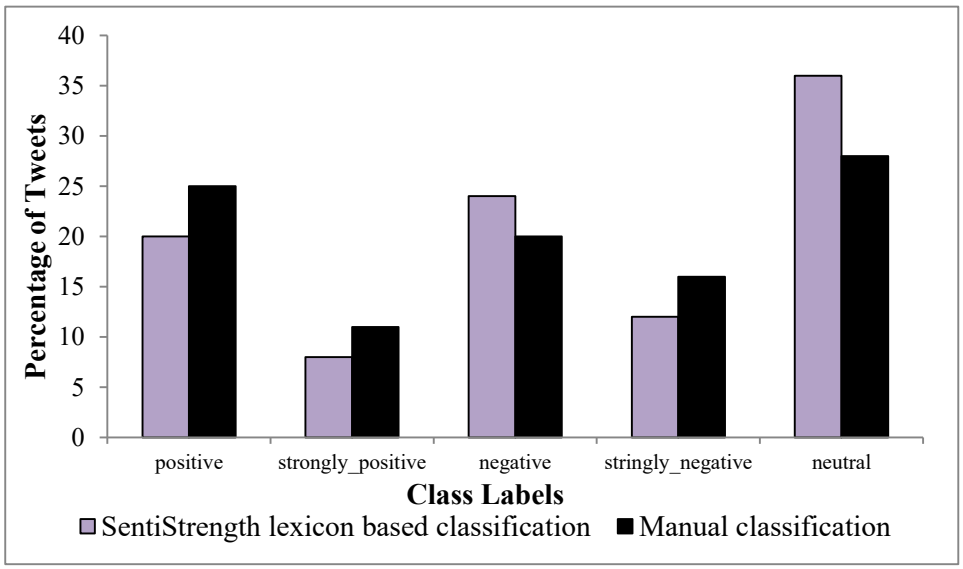


Figure 4.18 Percentage of Tweets on Lexicon based Classification and Manual Classification

Two experiments are conducted to evaluate the performance of multi-tier classification. At the first experiment, reference data which is correctly classified instances by using the preceding models are used. In this experiment, only local mistakes are identified because the misclassified instances from the preceding model have not been considered. At the second experiment, the accuracy is computed by using global data which is based on the subsequent results of preceding models. The

global data may contain the misclassified instances from all of the hierarchical classification models.

Table 4.5 Classification Accuracy of Single-tier and Multi-tier Classification

Architecture		Classes	Accuracy (%)	Overall Accuracy (%)
Multi-tier classification (with reference data)	Model I	Pos Neg Neu	- - 86.77	82.37
	Model II	positive strongly_positive	85.92 80.69	
	Model III	negative strongly_negative	79.52 78.95	
Multi-tier classification (with global data)	Model I	Pos Neg Neu	- - 84.93	80.03
	Model II	positive strongly_positive	83.41 76.32	
	Model III	negative strongly_negative	78.86 77.63	
Single-tier classification		neutral positive strongly_positive negative strongly_negative	80.41 78.35 70.58 77.35 69.51	75.24

Table 4.5 shows the comparative results of single-tier and multi-tier classification. The results show the multi-tier classification (with reference data) is higher than single-tier with 7% and the multi-tier classification (with global data) is higher than Single-tier with 5%.

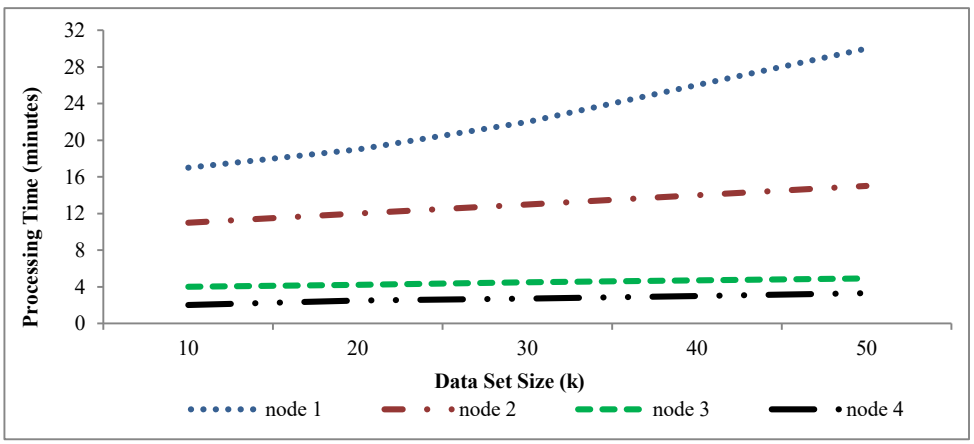


Figure 4.19 Processing Time of MSABDP

In order to test the scalability of the system, the MSABDP run the job with different number of tweets and with different number of nodes. Each node is developed on each machine. Figures 4.19 show the scalability of MSABDP and single node cluster to four node cluster. In particular, the processing time of data analysis are significantly decrease when adding one node to three nodes and the processing time of data analysis are hardly decrease when adding three nodes to four nodes. According to the results, the processing time is not proportional to the number of nodes due to the latency of IO performance of Hadoop cluster with default configurations.

4.4 Multi-tier SA System with Sarcasm Detection on Hadoop MapReduce (MSASDH)

The presence of sarcasm, an interfering factor that can flip the sentiment of the given text, is one of the challenges of Sentiment Analysis. Therefore, Multi-tier Sentiment Analysis system with sarcasm detection on Hadoop (MSASDH) is proposed to achieve high-level performance of sentiment classification. MSASDH identifies sarcasm and sentiment-emotion by conducting rule based sarcasm-sentiment detection scheme and sentiment classification with Multi-tier architecture.

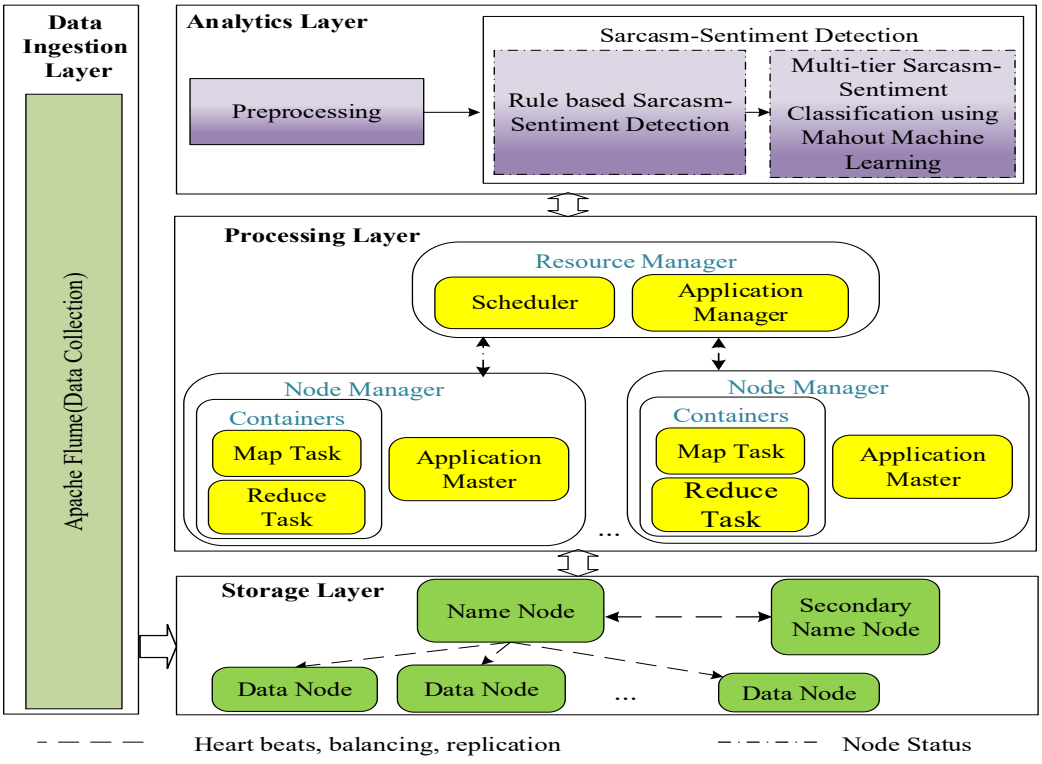


Figure 4.20 High Level Architecture of MSASDH

In MSASDH, Big Data Analytics Platform is implemented to scale up the traditional analytics platform for analyzing large-scale tweets by using Apache Flume [13], HDFS [45], MapReduce [45] and Mahout Machine learning library [3]. The Big Data Analytics Platform is composed of four layers and MSASDH is implemented at four layers. High level architecture of MSASDH is illustrated in Figure 4.20. It consists of four layers and the function of MSASDH on each layer is described as follow:

1. **Data Ingestion Layer** - In this layer, tweet stream data is collected and the collected data ingested to HDFS through the memory channel by using Apache Flume.
2. **Storage Layer** – HDFS, scalable and reliable data storage, is located in Storage Layer. HDFS serves master/slave architecture and single NameNode serve as a master server. NameNode executes file system namespace operations (i.e. opening, closing, and renaming files and directories). It also manages the mapping of blocks to DataNodes. DataNodes store the actual data in HDFS.
3. **Processing Layer** - Yarn and MapReduce-2 are located in the Processing Layer to process vast amounts of data in parallel on clusters of commodity hardware in a reliable, fault-tolerant manner.
4. **Analytics Layer** - preprocessing, class labeling and sentiment classification with multi-tier architecture are implemented in Analytics Layer. All of the processes from Analytics Layer executed in distributed manner by using HDFS and MapReduce. The sentiment classification with multi-tier architecture is implemented by using Mahout Machine learning library.

MSASDH consists of three major processes: data collection, preprocessing and sarcasm-sentiment detection and the processes are executed at the above four layers. Figure 4.21 illustrates the process flow of MSASDH.

4.4.1 Data Collection in MSASDH

In this work, Apache Flume collects Twitter stream data by Twitter Agent and the data is filtered by keywords “iphone”. Twitter Agent has three main components – a TwitterSource, a Memory Channel and a HDFS Sink. The Twitter source processes events and moves them along by sending the stream data into a Memory channel. The Memory channel acts as a pathway between the TwitterSource and HDFS Sink.

HDFS Sink, which writes events to the location defined in HDFS in the HDFS Sink configuration, determines the size of the file with the Count roll parameter.

4.4.2 Data Preprocessing in MSASDH

The aim of the data preprocessing is removing duplicate Tweets & noisy data, removing character repetitions & stopwords. Detail procedures of preprocessing steps are already mention in previous chapter. The preprocessing process not only simplifies the classification task, but also serves to decrease greatly the processing cost in the training phase.

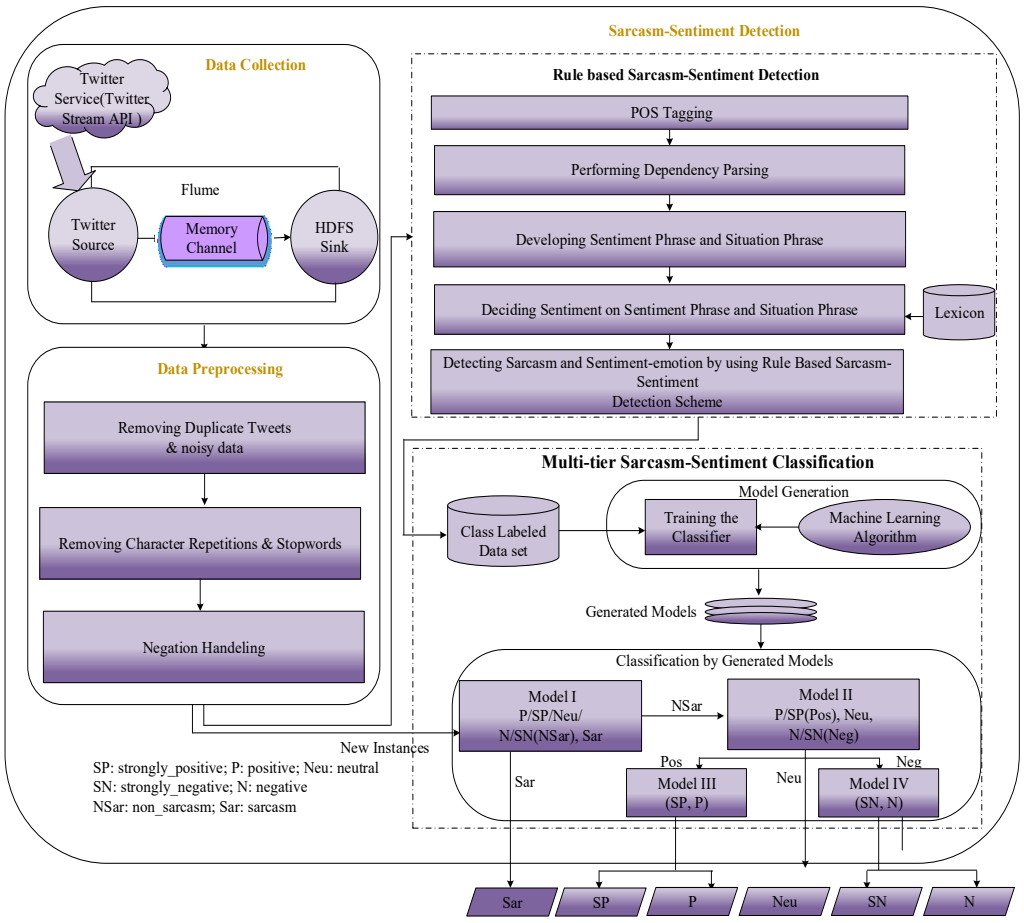


Figure 4.21 Process Flow Diagram of MSASDH

4.4.3 Sarcasm and Sentiment Detection

To detect sarcasm and sentiment, there are two major processes: rule based sarcasm-sentiment detection (sarcasm-sentiment class labelling) and multi-tier

sarcasm-sentiment classification. Detail procedures are presented in the following subsection.

4.4.3.1 Rule Based Sarcasm-Sentiment Detection

Instead of manual labelling the class, rule based sarcasm-sentiment detection approach is used for annotating the training data of the learning-based classifier. As the prestage of rule based sarcasm-sentiment detection, POS tagging, dependency parsing, developing sentiment and situation phrases, and deciding sentiment-emotion on the developed phrases are examined.

(a) POS Tagging

GATE Twitter POS tagger [59] is deployed to evaluate accurate POS tag information for the Twitter dataset. It is used to classify words into their part-of-speech and label them according to the tagset. The tagger is an adapted and augmented version of a leading CRF-based tagger, customized for English tweets. The tagger uses the Penn Treebank-tag set [70]. In this work, 19 tags for noun, adjective, adverbs and verbs are used for sentiment score calculation with linguistic feature selection.

(b) Dependency Parsing

The dependency parsing is a technique to analyze the syntactic relationship, such as relationships between words to isolate the whole sentence to morpheme. The syntactic structure is the parse tree which can be generated using parsing algorithms. These parse trees are useful and plays a critical role in the semantic analysis stage. In this work, Opennlp [109] is used for parsing. An example of parsing for text is "I love amazing new iphone because it runs awfully." The parse tree of an example text is shown in Figure 4.22.

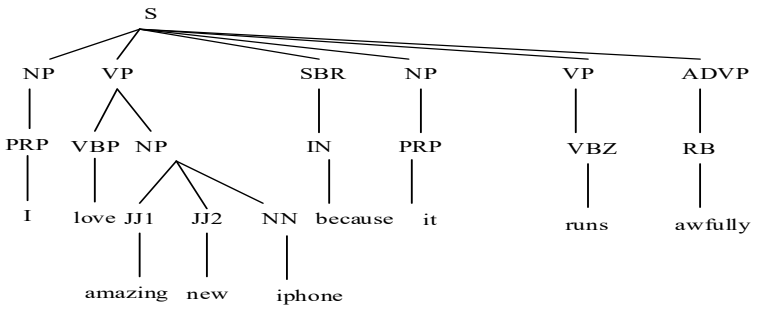


Figure 4.22 Sample Parse Tree

(c) Developing Sentiment and Situation Phrase

Using the parse tree, whether the part of speech of the parsed ones can be included as the part of speech of the developed phrases is determined. The combination of parts of speech represents as the developed phrases. The developed phrases are composed of sentiment phrases (SEP) and situation phrases (SIP). SIP means to the "action" in the sentence, and SEP means to the "emotion" in the sentence. To develop SIP and SEP, the input tweets is parsed into the form of phrases such as noun phrase (NP), verb phrase (VP), adjective phrase (ADJP), adverb phrase (ADVP), Negation pharse (NGP), etc. These phrases are subsequently classified into SIP and SEP as shown in the Mapper function of Figure 4.23. Table 4.6 describes the sample result phrases of developing SEP & SIP for the example text.

Table 4.6 Sample Result Phrases of Developing SEP & SIP Procedure

SEP	SIP
I love, amazing	run awfully

(d) Deciding Sentiment-Emotion on Sentiment Phrases (SEP) and Situation Phrases (SIP)

SentiStrength [103], a lexicon-based classifier, is used for deciding the sentiment-emotion. It uses additional (non-lexical) linguistic information and rules to detect sentiment score of developed phrases. For each message SentiStrength will give a positive feeling from 1 to 5 and a negative score from -1 to -5. For each sentiment phrase, if the sentiment score is equal with or greater than 2, the phrase is added to strongly_positive sentiment phrases ("SPSEP"). If the sentiment score is greater than 0 and less than 2, the phrase is added to positive sentiment phrases ("PSEP"). If the score is equal with or less than -2, the phrase is added to strongly_negative sentiment phrases ("SNSEP"). If the score is less than 0 and greater than -2, the phrase is added to negative sentiment phrases ("NSEP"). Otherwise, the phrase is added to neutral sentiment phrases ("NeuSEP"). As the same way, the sentiment-emotion of situation phrases ("SIP") is decided. For each situation phrase, if the sentiment score is equal with or greater than 2, the phrase is added to strongly_positive situation phrases ("SPSIP"). If the sentiment score is greater than 0 and less than 2, the phrase is added to positive situation phrases ("PSIP"). If the score is equal with or less than -2, the phrase is added to strongly_negative situation phrases ("SNSIP"). If the score is less

than 0 and greater than -2, the phrase is added to negative situation phrases ("NSIP"). Otherwise, the phrase is added to neutral situation phrases ("NeuSIP"). The procedure of deciding sentiment-emotion for developed phrases is illustrated in Reducer function of Figure 4.23. The output of this procedure is emotional SEP & SIP (PSEP, SPSEP, etc.).

```
Procedure : Deciding_Sentiment_Emotion_Job
Input : preprocessed tweets
Output :  Emotional SEP & SIP
Function Mapper (k1, v1)
while(value € values)
1.       POS Tagging
2.       Dependency Passing
3.       foundNG=false // NG: Negation
4.       if(value € NGW) // NGW : negation words
5.          foundNG = true
6.          if ( foundNG == true)
7.                Append NGP to SEP // NGP € (NGW+VP || NGW + ADJ)
8.          if (POStagger == ADJP || NP || NP + VP)
9.                Append  POStagger to SEP
10.          if ( POStagger == VP ||  VP + ADVP || VP + ADVP + ADJP ||
11.  VP + ADJP + NP || VP + NP || ADVP + VP || ADVP + ADJP + NP ||
     ADJP + VP)
12.               Append  POStagger to SIP
13. (totalscore>=2) //totalscore: the total score of each phrase
14.                    Append value to SPSEP
15.            else if (totalscore > 0 && totalscore <2)
16.                    Append value to PSEP
17.             else if(totalscore < 0 && totalscore >2)
18.                    Append value to NSEP
19.             else if(totalscore <= -2)
20.                    Append  value to SNSEP
21.             else if(totalscore ==  0)
22.                    Append value to NeuSEP
23.    while(value € SIP)
24.             if(totalscore>=2)
25.                    Append value to SPSIP
26.            else if(totalscore > 0 && totalscore <2)
27.                    Append value to PSIP
28.            else if(totalscore < 0 && totalscore >-2)
29.                    Append value to NSIP
30.            else if(totalscore <= -2)
31.                    Append  value to SNSIP
32.            else if(totalscore ==  0)
33.                    Append value to NeuSIP
34.    emit(tweets, values)
35. End function
```

Figure 4.23 Procedure of Deciding_Sentiment_Emotion for SEP & SIP

Sample result phrases of deceding positive sentiment phrases (“PSEP”) & negative situation phrases (“NSIP”) of the example text are shown in Table 4.7.

Table 4.7 Sample Result Phrases of Deciding Emotional SEP & SIP Procedure

PSEP	NSIP
I love, amazing	run awfully

(e) Detecting sarcasm and sentiment –emotion by using Rule based Sarcasm-Sentiment Detection Scheme

In this work, it takes testing tweets and emotional SIP & SEP from previous Deciding_Sentiment_values_Job procedure for detecting sarcasm and sentiment-emotion.

Table 4.8 Rule based Sarcasm-Sentiment Detection Scheme

		Emotional SIP				
Emotional SEP		PSIP	SPSIP	NSIP	SNSIP	NeuSIP
	PSEP	P	SP	Sar	Sar	P
	SPSEP	SP	SP	Sar	Sar	SP
	NSEP	Sar	Sar	N	SN	N
	SNSEP	Sar	Sar	SN	SN	SN
	NeuSEP	P	SP	N	SN	Neu

Table 4.8 illustrates the sarcasm and sentiment-emotion detection scheme. Based on this scheme, it determines sarcasm if the emotions of the sentiment and situation phrase are different. For example, if the testing tweet matches with any strongly_positive sentiment from SPSEP then it subsequently checks for any matches with checks match, then the testing tweet is sarcastic and similarly, it checks for sarcasm with a strongly_negative sentiment in a positive situation. Otherwise, the given tweet is not sarcasm. By using the sarcasm and sentiment-emotion detection scheme, the testing tweet is classified whether sarcasm or sentiment (P, SP, N, SN, Neu).

4.4.3.2 Multi-tier Sarcasm-Sentiment Classification

In order to implement the multi-tier sarcasm-sentiment classification, there are two main parts: model generation and classification by generated model. The classification model is generated by applying Mahout Naïve Bayes Algorithm [60] and the generated models are used to classify the new data (unlabeled data).

(a) Model Generation in MSASDH

To generate the models, the input data transformed into the sequence file. As this sequence file consists of key value pairs, class category and tweets_id are set to key and tweets are set to value. Lexical feature (ngram) and TFIDF feature vectors are used for improving the performance of classification models.

For multi-tier architecture, four classification models are generated and each model inherits the same configuration as the first model. To generate the first model (Model I), class labeled datasets that has all class categories except "Sar" is identified as "Nsar" and class category "Sar" is identified as "Sar". Model I is generated by training the classifier with all of the labeled datasets that has two class categories: "Sar" and "NSar". To generate the second model (Model II), class labeled datasets that class categories ("P" and "SP") identified as "Pos" and the class category ("N" and "SN") identified as "Neg". In model II, all of the labeled datasets (class category is Pos, Neu, Neg) are used to train the classifier. Model III is generated by training the classifier with the labeled datasets that class category "P" and "SP". Model IV is generated by training the classifier with the class labeled datasets which class category is "N" and "SN".

(b) Classification by Generated Models in MSASDH

The newly incoming tweets are classified by using generated models. Firstly, new test instance (unlabeled data) is classified into "Sar" or "NSar" by Model I. If the class category of new test instance is "NSar", the instance is moved to Model II. Otherwise, the instance is identified as "Sar". In Model II, the new instance is classified into "Pos" or "Neg" or "Neu". If the class category of new test instance is "Pos", the instance is moved to Model III. If the class category of new test instance is "Neg", the instance is move to Model IV. Otherwise, the instance is identified as "Neu". In Model III, the test instance is classified into "P" or "SP". In Model IV, the test instance is classified into "N" or "SN".

For classification, naïve bayes classifier uses probabilities to decide which class best matches for a given input text. Word id and tfidf weight are used to create vector for the new tweet. The naïve bayes classifier is classified by using the vector". For model I, the score of two class label is calculated. If the "bestcategoryId" is equal with "1", the classifier classify as "Sar". Otherwise, the classifier classify as "NSar". For model II, the score of three class labels is calculated. The "bestscore" is set to "-Double.MAX_VALUE". The "bestcategoryId" is set to "-1" and "categoryId" is set to index of classification result. If the indexed score is greater than "bestscore", bestcategoryId is replaced with categoryId. If the "bestcategoryId" is equal with "1", the classifier classify as "Pos". If the "bestcategoryId" is equal with "0", the classifier classify as "Neu". Otherwise, it classify as "Neg". For model III, the score of two class label is calculated. If the "bestcategoryId" is equal with "1", the classifier classify as "strongly_positive". Otherwise, the classifier classify as "positive". For model IV, the score of two class labels are calculated. If the "bestcategoryId" is equal with "1", the classifier classify as "strongly_negative". Otherwise, the classifier classify as "negative". As the result combination is not needed, the Reduce stage outputs the results obtained by the Mapper function.

4.4.4 Experiments and Results of MSASDH

To evaluate the performance of MSASDH, the usefulness of RSSD are measured and the accuracy of multi-tier classification with sarcasm detection is compared with multi-tier classification without sarcasm detection. In order to test the scalability of this system, the processing time of MSASDH are measured on different Hadoop cluster nodes. For evaluation, the required system specification and the dataset are presented in this section. In addition, explanations about evaluation results are also described.

4.4.4.1 Experiment Environment of MSASDH

In this experiment, the cluster is composed of four computing nodes (VMs) and each node is developed on each machine. The specifications of devices and necessary software component of MSASDH are presented in Table 4.9.

Table 4.9 Testing System Specification of MSASDH

Server/Client OS	Ubuntu 14.04 LTS
Host Specification	Intel ® Core i7-3770 CPU @ 3.40GHz, 8GB Memory, 1TB Hard Disk
VMs Specification	4GB RAM, 100 GB Hard Disk
Software Component	- Hadoop 2.7.1 - Flume 1.6 - SentiStrength 2 - Mahout 0.10.0

4.4.4.2 Data Set of MSASDH

In order to test the functionality of the MSASDH, tweets related with IPHONE mobile product are examined. As the role of RSSD is important for determining the performance of MSASDH, 10,000 tweets are randomly collected to measure the usefulness of RSSD. 200,000 tweets (June-July, 2017) are collected as the training datasets and 50000 new batches of tweets are collected as the test set for evaluating performance of MSASDH. To cover more sarcastic words, 20000 tweets having the hashtag "# sarcasm" are added to the training data set.

4.4.4.3 Evaluation Results of MSASDH

To measure the usefulness of RSSD, it compared with the existing sarcasm detection approaches. The existing sarcasm detection approach of S. Suzuki and PBLGA are taken as the baseline ones. Key Performance Indicator (i.e., accuracy, precision and recall) are used to evaluate the performance of RSSD.

Table 4.10. Performance of RSSD Compared To the Baseline Ones

	Accuracy(%)	Precision(%)	Recall(%)
PBLGA[96]	72	76	69
S.Suzuki et al. [102]	76	81	70
RSSD	**85**	**90**	**78**

Table 4.10 shows the comparative results of RSSD and the baseline ones. The result of applying RSSD to the dataset, RSSD achieved 85%, 90% and 78% accuracy, precision, recall respectively. According to the results, RSSD clearly outperforms the baseline ones, for the used two dataset: it has accuracy, precision and recall is noticeably higher than the baseline ones. The reason of this result would be that the existing sarcasm detection approach of S. Suzuki and PBLGA do not handle negation. It is clear that the advantage of RSSD is due to consider negation handling.

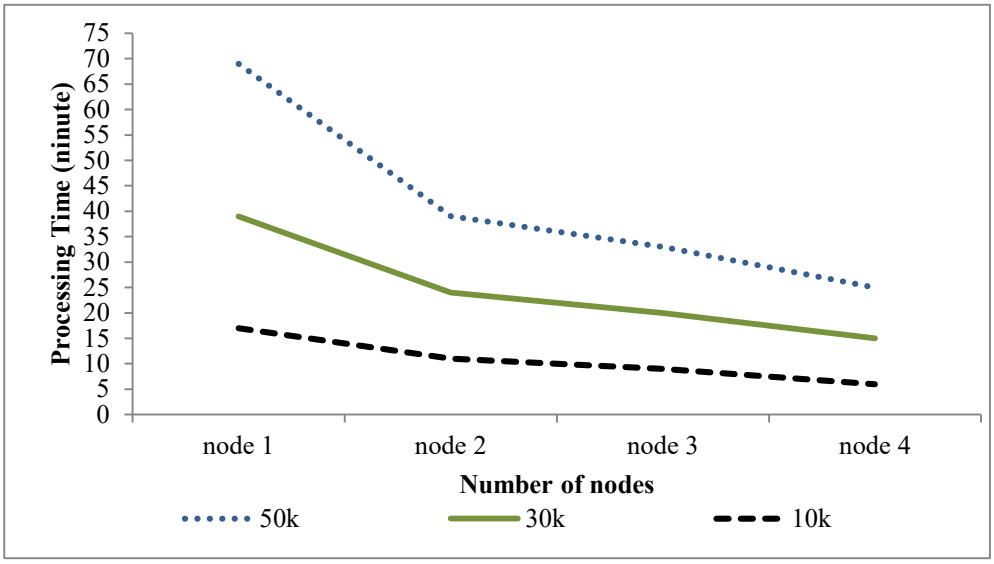
Key Performance Indicator (i.e., accuracy, precision and recall) are used to evaluate the performance of MSASDH.

Table 4.11 Comparative Results of MSASDH and MSABDP

	Classes	Accuracy (%)	Overall Accuracy (%)
MSABDP (without sarcasm detection)	positive	82	82
	strongly_positive	76	
	negative	88	
	strongly_negative	80	
	neutral	84	
MSASDH (with sarcasm detection)	positive	90	87
	strongly_positive	82	
	negative	95	
	strongly_negative	83	
	neutral	85	

Table 4.11 presents the classification accuracy of MSASDH and MSABDP. The overall accuracy of multi-tier SA with sarcasm detection (MSASDH) is higher than without sarcasm detection (MSABDP) by 5%.

Different number of tweets and different number of nodes are used to measure the processing time of MSASDH. This system run launched a MapReduce job. Figure 4.24 shows the processing time of MSASDH and this time is measured from data preprocessing to sarcasm-sentiment detection. The results show that the processing time of sarcasm-sentiment class labeling decreases when the number of nodes is increased. In particular, for 10k tweets, the processing time is decreased by 35% when increasing from single cluster node to 2 cluster nodes, and 47% for single cluster node to 3 cluster nodes, 65% for single cluster node to 4 cluster nodes.



4.24 Processing Time of MSASDH

For 30k tweets, the processing time is decreased by 41% when increasing from single cluster node to 2 cluster nodes, and 50% for single cluster node to 3 cluster nodes, 60% for single cluster node to 4 cluster nodes. For 50k tweets, the processing time is decreased by 50% when increasing from single cluster node to 2 cluster nodes, and 57% for single cluster node to 3 cluster nodes, 67% for single cluster node to 4 cluster nodes. According to the results, the processing time is not proportional to the number of nodes due to the latency of IO performance of hadoop cluster with default configurations. For 30k tweets, the processing time is decreased by 41% when increasing from single cluster node to 2 cluster nodes, and 50% for single cluster node to 3 cluster nodes, 60% for single cluster node to 4 cluster nodes. For 50k tweets, the processing time is decreased by 50% when increasing from single cluster node to 2 cluster nodes, and 57% for single cluster node to 3 cluster nodes, 67% for single cluster node to 4 cluster nodes. According to the results, the processing time is not proportional to the number of nodes due to the latency of IO performance of Hadoop cluster with default configurations.

4.5 Chapter Summary

In this chapter, multi-class SA system is implemented with different architectures on Big Data Analytics Platform. Firstly, SSABDP is developed for conducting multi-class SA and it is implemented by combining lexicon and learning based classification scheme with single-tier architecture. The results show that the classification accuracy

of SSABDP is higher than only lexicon based approach. The next one, MSABDP is implemented by combining lexicon and learning based classification scheme with multi-tier architecture. The evaluation results show that the proposed MSABDP is able to significantly improve the classification accuracy over SSABDP (multi-class classification with Single-tier architecture) by 7%. Moreover, the processing time results show that the MSABDP is efficient and scalable by decreasing the processing time when adding more nodes in the cluster. And MSASDH is implemented for achieving high-level performance of sentiment classification. MSASDH identifies sarcasm and sentiment-emotion by conducting rule based sarcasm-sentiment detection scheme and sentiment classification with Multi-tier architecture. The evaluation results show that the overall accuracy of multi-tier SA with sarcasm detection (MSASDH) is higher than without sarcasm detection (MSABDP) by 5%.

CHAPTER 5

REAL-TIME MULTI-TIER SA

Nowadays, Big Data, both structured and unstructured data, are generated from Social Media. Social Media are powerful marketing tools and SBD require real-time tracking and analytics because the speed may indeed be the most important competitive business profits. Compared to batch processing of SA on Big Data Analytics platform, Real-time analytic is data intensive in nature and require to efficiently collect and process large volume and high velocity of data. Real-time multiclass SA is oriented towards classification of text into more detailed sentiment labels in real-time manner. But Multiclass SA with Single-tier architecture where single classification model is developed and entire labeled data is trained may decrease the classification accuracy. Therefore, Real-time Multi-tier SA system (RMSA) is developed and detail procedures are presented in this chapter.

5.1 Real-time Multi-tier SA (RMSA)

RMSA is proposed for real-time analysis. In RMSA, real-time analytics platform is implemented for analyzing large volumes and high velocity of tweets by using Apache Flume [13], HDFS, Spark streaming [105] and Spark MLlib [104]. High level architecture of the RMSA is illustrated in Figure 5.1. It consists of four layers and the function of each layer is described in this subsection.

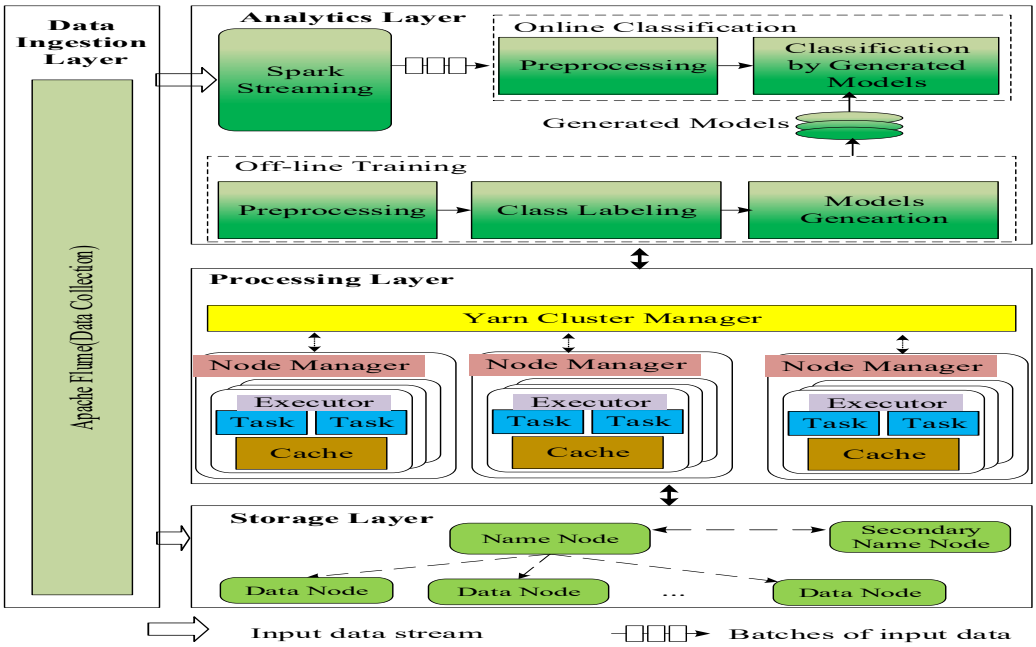


Figure 5.1 High Level Architecture of RMSA

Data Ingestion Layer - In this layer, tweet stream data is collected by Apache Flume. For offline processing, the collected data is pushed into HDFS sink. For on line processing, Spark Streaming sets up a receiver that acts an Avro agent and the data are pushed into the avro sink.

Storage Layer - In the Storage Layer, HDFS is used to store scalable and reliable data. HDFS provides a master / slave architecture and a single NameNode acts as the primary server. Name Node performs the operation of the file system namespace, such as opening, closing and renaming. Files and directories in addition, it also defines the block mapping to DataNodes DataNodes used to store real data in HDFS.

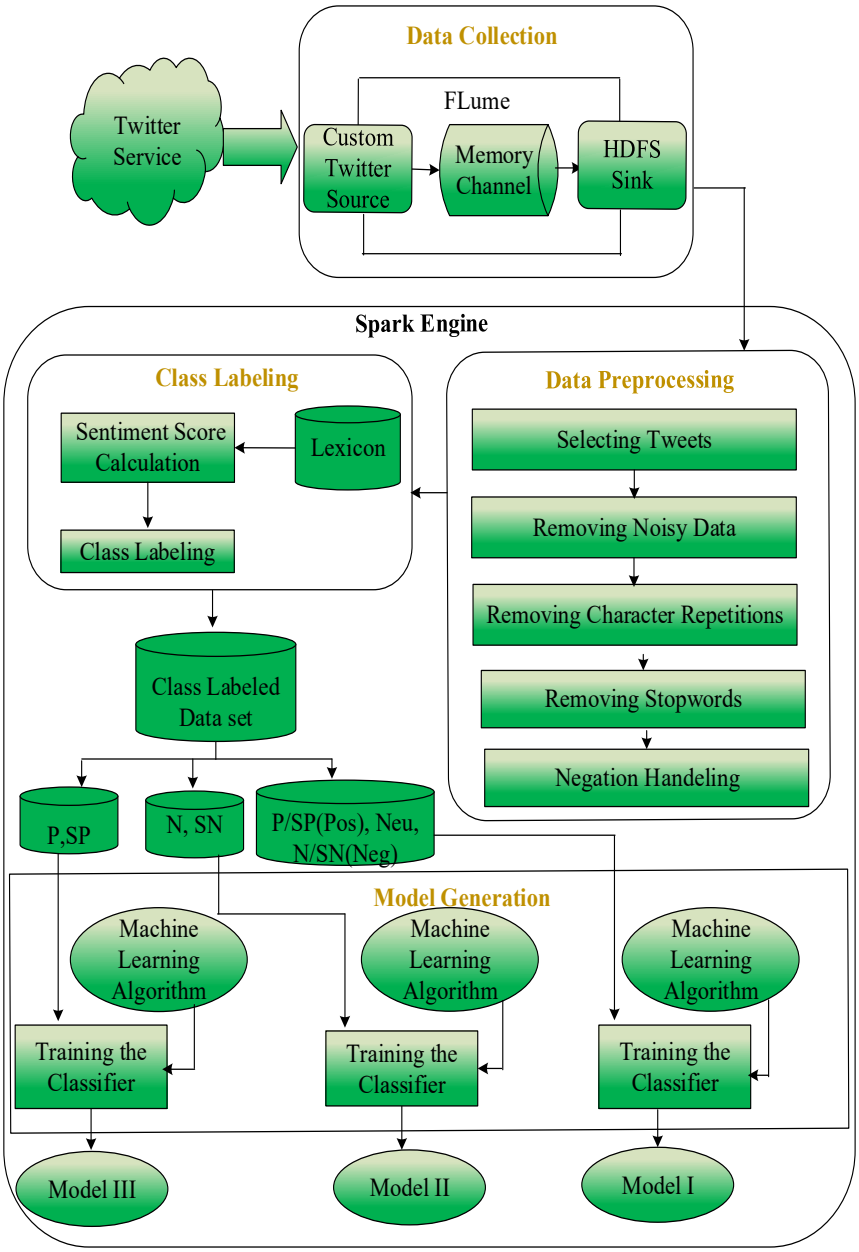
Processing Layer – Yarn Cluster Manager and Spark executives are in the processing layer to process large amounts of data in parallel on a product hardware cluster in a reliable and fault-tolerant manner. YARN Resource Manager keeps track of the resources of each Node Manager YARN. Node Manager manages resources on the Slave node. Each slave node may have one / many resource availability at the same time and each executor can have a work process. And according to schedule Each task is performed in a separate JVM under the operator. The data is loaded in terms of the RDD partition into multiple handlers and the conversion is applied to these RDD partitions. The code operates on a topic called work. The manager is designed as a multi-threaded process and hence can run multiple tasks simultaneously.

Analytics Layer – The two components: Off-line training and On-line classification are implemented in this layer. To generate the classification models, Data preprocessing, class labeling and models generation are performed at Off-line training. The new incoming preprocessed Real-time tweets are classified by On-line classification. All of the processes from Analytics Layer are executed in distributed manner by using Spark engine. The sentiment classification with Multi-tier architecture is implemented by using Spark MLlib.

RMSA is implemented with two components: Off-line training and On-line classification. At the Off-line training, the data is collected by Apache Flume and the collected data is stored in HDFS. The preprocessing is performed and the classification model is generated by Off-line training. At the On-line classification, the Real-time classification is performed by generated model. The new incoming tweets stream data is classified with Multi-tier architecture while once the model is generated.

5.1.1 Off-line Training

The Off-line training consists of four major processes: Data Collection, Data Preprocessing, Class Labeling and Classification model generation. Figure 5.2 illustrates the process of flow of Off-line classification.



P: positive, SP: strongly_positive, N: negative, SN: strongly_negative, Neu: neutral

Figure 5.2 Process flow Diagram of Off-line Training

5.1.1.1 Data Collection for Off-line Training

Data collection module requires a distributed, reliable, highly available, capable of HDFS sink. The custom Twitter source processes events and moves them along by d sending the stream data into a Memory channel. The Memory channel acts as a g pathway between the Twitter source and HDFS Sink. The events are written to a e configured location in HDFS. In the HDFS Sink configuration, defines the size of the l, files with the roll Count Parameter.

5.1.1.2 Data Preprocessing for Off-line Training

As flume ingests the data as the nested JSON format and it may contain irrelevant and duplicated data, this data has to be cleaned and preprocessed for effective analysis. Data preprocessing steps are briefly explained in the following subsection.

1. **Selecting Tweets**: For SA, tweet_text feature (tweets) is selected among other feature because it expresses twitter users' feeling and opinion. Tweet_id feature is also selected to assign as an id no of Spark data frame.
2. **Removing Noisy Data**: The term noisy data describes any piece of information within the tweet that will not be useful for the machine learning algorithm to assign a class to that tweet. There are included noisy data such as character repetitions, website links with URL, @username, punctuation additional white space Replace hash tags with the same word without the hash tags. For example, #fun is replaced with fun. Converted @username to "usermentionsymbol by replacing @username instances found in tweets with "usermentionsymbol" for the classifier to easily identify that a user is being referenced. Non-Alphabets are replaced with space.
3. **Removing Character Repetition**: To remove the character repetition, there is needed to be checked whether the repetitive characters contain or not in the input data. If the repetitive characters are found and the character count is more than 2, replace the character itself by deleting the repetitive characters with substring function.
4. **Removing Stopwords**: When working with text classification methods, removal of stopwords is a common approach to reduce noise in the data. In this work, not only common stopwords but also stopwords based on classification domain are

considered by manually examining the data. For example, domain stopwords contain iphone, apple, mobile, etc.

5. **Negation Handling**: Negation handling is one of the factors that significantly affect the accuracy of learning based classifier. For example: the word "good" in the phrase "not good" will be contributing to positive sentiment rather that negative sentiment as the presence of "not" before it is not taken into account.

5.1.1.3 Class Labeling for Off-line Training

The detail procedure of class labeling is already mentioned in the section 4.1.1.

5.1.1.4 Classification Model Generation for Off-line Training

The detail procedure of classification model generation is already mentioned in the section 4.2.2.1.

5.3.2 On-line Classification

The On-line classification consists of three major processes: data collection, Spark Streaming, Data Preprocessing and Multi-tier classification. Figure 5.3 illustrates the process of flow of Off-line classification.

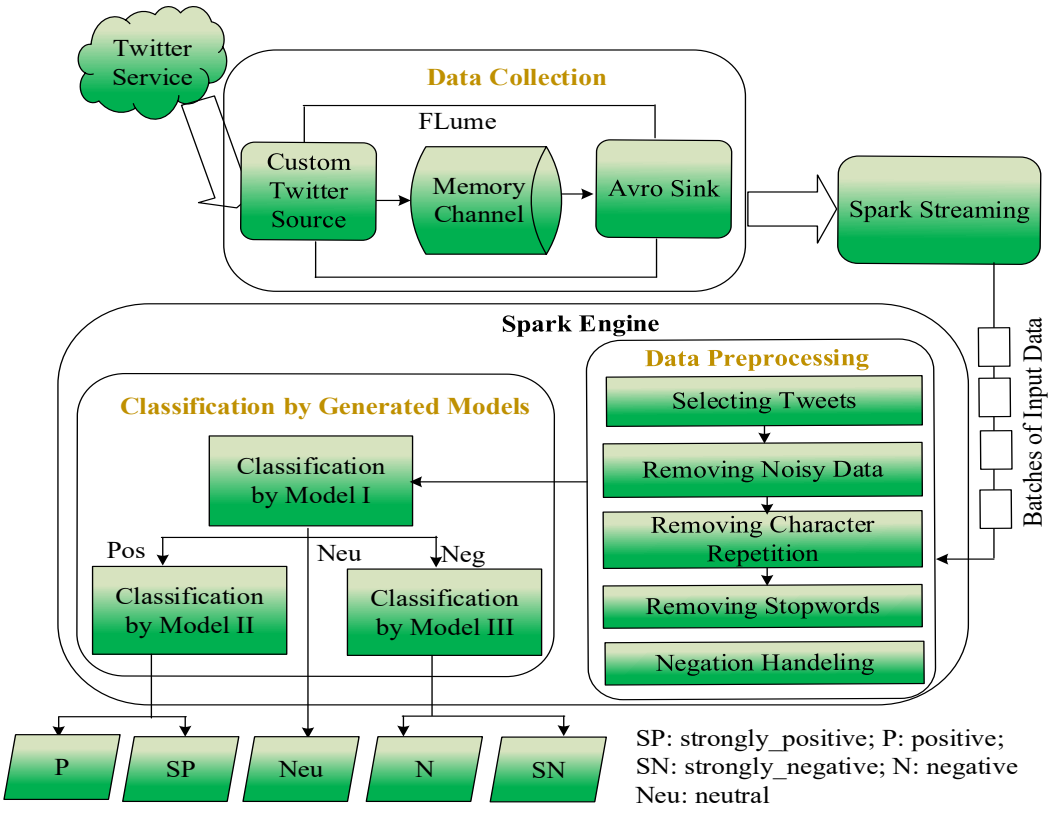


Figure 5.3 Process flow Diagram of On-line Classification

5.1.2.1 Data Collection for On-line Classification

In this work, Apache Flume pushes tweets to Spark streaming for each tweet without buffering any tweet into any batch by setting real time streaming configuration and the data are pushed into Avro sink. The configuration consists of three main components: Twitter source, Memory channel, Avro sink. The custom Twitter source moves events to Avro sink through Memory channel. Flume events sent to this sink are turned into Avro events and sent to the configured hostname / port pair and the coming tweets are stored in memory.

5.1.2.2 Spark Streaming

Once the new tweets arrives in Avro sink, Spark Streaming [105] divides the live stream of data into batches (called micro batches) of a pre-defined interval (n seconds). And each batch of data is handled as RDD. Once the data is available as RDD objects, which are then processed by the Spark engine to generate the final stream of results in batches. The time interval for Spark Streaming is defined by data processing requirements. Therefore, Spark Streaming can provide the capability to process data in near real time.

5.1.2.3 Data Preprocessing for On-line Classification

The aim of the data preprocessing is removing duplicate tweets & noisy data, removing character repetitions & stopwords. The preprocessing process simplifies the classification task. The detail procedure of preprocessing steps are already mentioned in section in 5.3.1.2.

5.1.2.4 On-line Classification by Generated Models

For classification, Naïve Bayes uses probabilities to decide best matches for the input tweets. Word id and TFIDF weight are used to create vector for the input tweets. The classification is performed by using the vector. The probability identification of Naïve Bayes is performed as follows:

$$Y_{nb} = arg_y \max P(Y = y) \prod_{i=1}^{n} P(X_i = u_i | Y = y) \quad (5.1)$$

Where $Y_{nb} \in \{y_1, y_2, \ldots, y_k\}$ for a given x and $X = (X_1 = u_1 \ldots, X_m = u_m)$ is the instance of the tweets to be classified.

For linear methods can be formulated as a convex optimization problem, i.e. the task of finding a minimizer of a convex function *f* that depends on a variable vector w, which has d entries.

$$f(w) \coloneqq \tau R(w) + \frac{1}{n}\sum_{i-1}^{n} L(w; x_i, y_i) \tag{5.2}$$

Vectors $x_i \in R^d$ are the training tweets, for $1 \leq i \leq n$, and $y_i \in R$ are their corresponding prediction labels, the method linear if $L(w; x, y)$can be expressed as a function of $W^T x \; and \; y$. The objective function f has two parts: the regularizer that controls the complexity of the model, and the loss that measures the error of the model on the training data. The loss function $L(w; x, y)$ is typically convex function in w. The fixed regularization parameter $\tau \geq 0$ defines the trade-off between the two goals of minimizing the loss (training error) and minimizing model complexity. For Linear SVC, the above equation 5.2 with the loss function in the formulation:

$$L(w; x, y) \coloneqq \max(0, 1 - yW^T x) \tag{5.3}$$

For Logistic Regression, the above equation 5.2 with the loss function in the formulation:

$$L(w; x, y) \coloneqq \log(1 + - \exp(W^T x)) \tag{5.4}$$

And the model makes the prediction by applying the logistic function:

$$f(z) = \frac{1}{1+e^{-z}} \tag{5.5}$$

where $z = W^T x$.

5.2 Experimental Evaluation of RMSA

The performance comparison of three different scalable machine learning techniques are evaluated to select the high performance of learning based technique for RMSA. For evaluating the performance, the required system specification and the dataset are presented in this section. In addition, explanations about evaluation results are also described.

5.2.1 Experiment Specification of RMSA

In this experiment, the Spark cluster is developed with three instances using three virtual machines. Each VM run on each machine. The specifications of devices and necessary software components of RMSA are presented in Table 5.1.

Table 5.1 Testing System Specification of RMSA

Server/Client OS	Ubuntu 14.04 LTS
Host Specification	Intel ® Core i7-3770 CPU @ 3.40GHz, 8GB Memory, 1TB Hard Disk
VMs Specification	4GB RAM, 100 GB Hard Disk
Software Component	- Hadoop 2.7.1 - Flume 1.6 - Spark 2.2.0 -Scala 2.11.

5.2.2 Data sets of RMSA

In order to test the functionality of the RMSA, tweets related with IPHONE mobile product are examined. For Off-line training, 57,474 tweets (February to the 1st week of March, 2018) are collected as the training dataset. For On-line prediction, about 6897 tweets (2nd week of March, 2018) are collected in a real time and it is used as the test set for evaluating performance of RMSA.

5.2.3 Results of RMSA

Different number of tweets and three different number of machine learning techniques (Naïve Bayes, Linear SVC and Logistic Regression) are used to measure the performance and processing time of RMSA. This system run launched a Spark engine.

Figure 5.4 illustrates the processing time of Off-line training phrase and this time is measured from data preprocessing to model generation. The processing time is measured on the three different number of machine learning techniques (Naïve Bayes, Linear SVC and Logistic Regression) with two different architectures (Single-tier and Multi-tier). Naïve Bayes takes very less time and Linear SVC takes longest time for Off-line training.

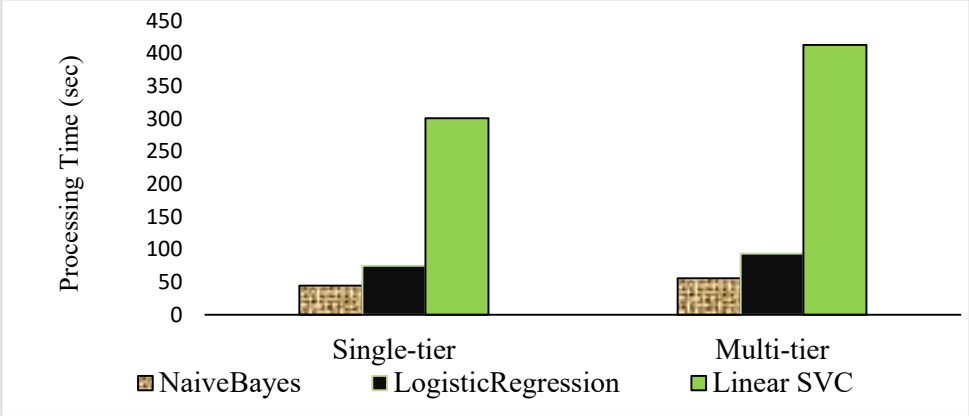


Figure 5.4 Processing Time of Different Classifiers with Different Architectures for Off-line Training

Table 5.2 shows the processing time of On-line prediction phrase and this time is measured from data collection to unseen data prediction. According to the results, Naïve bayes takes very less time for both Off-line training and On-line predicting compared to the Logistic Regression and Linear SVC. Logistic Regression is faster than Linear SVC for both training and prediction. Linear SVC takes longest time for both Off-line training and On-line predicting compared to the Logistic Regression and Linear SVC.

Table 5.2 Processing Time of Different Classifiers with Different Architecture for On-line Prediction

Architecture	Machine Learning (ML) Techniques	Rate of tweets
Single-tier	NaïveBayes	1,688 tweets/min
	Linear SVC	1,215 tweets/min
	Logistic Regression	1,639 tweets/min
Multi-tier	NaïveBayes	1,678 tweets/min
	Linear SVC	1,037 tweets/min
	Logistic Regression	1,615 tweets/min

The comparative performance of three different classifiers are shown in Table 5.3. Key Performance Indicator (i.e., fscore, accuracy) is used to evaluate the performance of different classifiers. Among these methods, Liner SVC with oneVsRest approach has provided the best results with 96% for Multi-tier architecture. Naïve Bayes classifier does not perform well for large scale data and Logistic Regression approach achieves higher accuracy than Naïve Bayes classifier.

Table 5.3 Comparative Results of Different Classifiers(Multi-tier Architecture)

MLTechni-ques	Models	Classes	F-score	Accuracy (%)	Overall-Accuracy (%)
Naïve Bayes	Model I	Neu Pos Neg	80 69 82	79	86
	Model II	SP P	80 92	89	
	Model III	SN N	88 90	89	
Logistic Regression	Model I	Neu Pos Neg	92 90 93	93	94
	Model II	SP P	92 96	97	
	Model III	SN N	97 96	97	
Linear SVC	Model I	Neu Pos Neg	92 91 93	92	96
	Model II	SP P	95 98	95	
	Model III	SN N	96 97	97	

The performance results (in terms of accuracy) of Single-tier and Multi-tier architectures are summarized and compared for three different classifiers, as shown in Table 5.4.

Table 5.4 Overall Accuracy of Single-tier Vs Multi-tier

Architecture	ML Techniques	Overall Accuracy (%)
Single-tier	NaiveBayes	69
	Logistic Regression	91
	Linear SVC	89
Multi-tier	NaiveBayes	86
	Logistic Regression	94
	Linear SVC	96

It is clear that, in each and every method, the Multi-tier architecture improves the accuracy significantly. The result based on RMSA, Multi-tier architecture is better dealing with the high correlation data. It is remarkable that Logistic Regression is the best classifier for single-tier architecture but it doesn't perform well for Multi-tier

architecture. Because Logistic Regression is not good when high correlation structures are observed in the data. Linear SVC with oneVsRest approach achieves the highest accuracy for Multi-tier architecture but it does not well performed for multiclass classification with Single-tier architecture. Because SVM have difficulty when the classes are not linearly separable.

5.3 Chapter Summary

In this chapter, Real-time Multi-tier SA system (RMSA) is developed for achieving high level performance of multi-class classification in Real-time manner. Lexicon and learning based classification scheme with Multi-tier architecture are combined to develop the proposed system. Real-time twitter stream data is collected by apache flume and, large volumes and high velocity of social data is efficiently analyzed by Spark. To improve the classification accuracy, the suitable classifier is selected by comparing the accuracy of three different learning based multiclass classification techniques: Naïve Bayes, Linear SVC and Logistic Regression. The evaluation results show that Real-time Multi-tier SA will achieve the promising accuracy and Linear SVC is better than other techniques for Real-time Multi-tier SA.

CHAPTER 6

CONCLUSION AND FURTHER RESEARCH DIRECTION

Today, the structured and unstructured data on the internet is increasing at a rapid pace. Different industries and companies are using this huge structured and unstructured data for extracting the people's views towards their industrial and business purpose for growing the company. Social Media is a very important source for extracting the information according to the purpose and process. This also results in rapidly growing popularity and interest in automated Opinion mining also known as SA. Social Media provides large volume and high velocity of data that can be used for training a classifier for SA. Processing such large amount of data, for SA, is very time consuming with single node. To develop SA system on social Big Data, it still have many challenges such as Big Data collection, storage, and processing in an efficient timely manner. This research focuses primarily on developing high level performance of SA system on different platforms and has five main contributions:

1. Big Data Analytics Platform is implemented for scaling up the traditional analytics platform to analyze large volume of SBD in an efficient and timely manner.
2. Instead of manual labeling, the proposed SA is implemented by combining lexicon-based and learning based approaches.
3. Multi-tier SA on Big Data Analytics Platform (MSABDP) is proposed to improve the performance of multi-class classification.
4. To improve the classification accuracy, the proposed SA system is implemented by conducting RSSD and distributed learning based classification with multi-tier architecture.
5. Real-time stream analytics platform based on Spark is implemented and the proposed SA system is developed on this platform for analyzing real-time manner.

6.1 Thesis Summary

This Thesis covered SA on different platforms and the proposed SA system classifies the polarity on the Twitter stream data. This thesis consists of three parts: SA on traditional analytics platform, SA on Big Data Analytics Platform (Hadoop MapReduce) and Real-time SA on Big Data Analytics Platform (Spark). The work performed SA on traditional analytics platform has been published in the publication

[p1]. The works related to SA on Big Data Analytics platform (Hadoop) have been published in [p2, p3 and p4]. The work related to Real-time SA on Big Data Analytics platform (Spark) can be found in [p5] of Author's publication section.

In SA on traditional analytics platform, the proposed SA run on serial computing. The sentiment classification is implemented by combining lexicon and supervised machine learning-based approach. Lexicon-based approach is adopted to reduce time and labor consuming for manual labeling while SentiStrength lexicon-based classifier is applied to label the class. The system enables high-level performance of learning based classification while taking advantage of the lexicon-based classifier's effortless setup process. The evaluation result shows the reliable performance of lexicon-based classifier by comparing manual classification results and achieved the accuracy rate is 88%. The class labeled data is used to develop learning-based classification model. To achieve high-level performance of classification model, the suitable model is selected by comparing the accuracy of three different classifier. The evaluation result show that Naïve Bayes classification model is better and achieved accuracy rate is 93.4%. In addition, the selected classification model is used to classify the newly collected data. The evaluation Result shows the accuracy of selected model is 94% and the result is compared with the manual classification one. The above evaluation results show that the proposed system is achieved the promising accuracy.

With the rapid increase in the amount of social data produced and that is available, the increasing demand of the processing power for solving computational problems has been compelling for innovative ways coping with the need, beyond the level of conventional computing. Big Data Analytics platform is developed to scale up the traditional analytics platform for analyzing large scale social big data. SA is implemented on Big Data Analytics platform for extracting useful information from large volumes of social big data. Hadoop is built for big data analytics and it is a good platform for being able to manage large data at scale and which can improve scalability and efficiency by adopting distributed processing environment since they have been implemented using a MapReduce framework and a Hadoop distributed storage (HDFS). The proposed SA system consists of four modules: data collection, data cleaning, class labeling and preprocessing, sentiment classification. The sentiment classification is implemented by combining lexicon and supervised machine learning-based approach. Lexicon-based approach is adopted to reduce time and labor consuming for manual labeling while SentiStrength lexicon-based classifier

is applied to label the class. The system classifies the polarity (positive, negative and neutral) on the real Twitter stream data. The system enables high-level performance of learning based classification while taking advantage of the lexicon-based classifier's effortless setup process. Evaluation results show that the reliability of the performance of lexicon-based classifier by comparing manual classification results and achieved the accuracy rate is 75%. The class labeled dataset is preprocessed and the evaluation result show that preprocesses can be able to improve the accuracy of classifier. To achieve high-level performance of the classification model, the suitable model is selected by comparing the accuracy of Mahout naïve bayes classifier with different size of datasets. The evaluation result show that 80% training dataset size is better and achieved the accuracy rate is 82.56%. The overall accuracy and f-measure of the proposed SA system are 84.2 and 83.0. For scalability, the evaluation results show that the running time of the system with different volumes of data decreases when adding more nodes into the cluster.

SA based on multiclass classification with Single-tier architecture, where single model is developed and entire labeled data is trained, may decrease the classification accuracy. Therefore, Multi-tier SA system is implemented on Big Data Analytics Platform to extract the valuable information from large amount of social data. Apache Flume is used to collect a huge amount of Twitter stream data and MSABDP classifies the Twitter data into five classes: strongly_positive, positive, neutral, negative, and strongly_negative. The sentiment classification is implemented by combining lexicon and supervised machine learning-based approach. The system enables high-level performance of learning based classification while taking advantage of the lexicon-based classifier's effortless setup process. To increase the multiclass classification accuracy, learning based approach with Multi-tier architecture is applied to classify the multiclass. the Multi-tier classification is higher than Single-tier with 7%. The processing time of MSABDP with different volumes of data decreases when adding more nodes into the cluster.

The presence of sarcasm, an interfering factor that can flip the sentiment of the given text, is one of the challenges of SA in social Media text, especially tweets. Therefore Multi-tier SA with Sarcasm Detection on Hadoop platform (MSASDH) is proposed for the classification accuracy improvement of emotional judgment in the large-scale social data. This system is implemented by conducting Rule based Sarcasm Sentiment Detection (RSSD) and scalable learning based classification

scheme with multi-tier architecture. According to the comparative result of the RSSD with existing state-of-art approach, the usefulness of the RSSD could be confirmed. The results show the accuracy of Multi-tier SA with sarcasm detection scheme is higher than without sarcasm detection. Therefore, the results show that MSASDH can enhance the accuracy of SA and opinion mining by detecting sarcastic statements. Moreover, this results show that the MSASDH is efficient and scalable by decreasing the processing time while processing on the different Hadoop cluster node.

In microblogging environment the real-time interaction is a key feature and thus the ability to automatically analyze information and predict user sentiments as discussions develop is a challenging issue. For that reason, the third stage is developing Real-time Multi-tier SA system (RMSA) on Big Data Analytics Platform (Spark). This system, consists of two components: Off-line training and On-line classification, implemented on Spark Real-time analytics platform. Combination of lexicon based and learning based approach with Multi-tier architecture are implemented. To achieve the suitable learning based technique for RMSA, the performance comparison of three learning based techniques (Naïve Bayes, Logistic Regression and Linear SVC) are evaluated. The evaluation results show the Linear SVC achieved the highest accuracy for RMSA but it takes longest time for training and predicting unseen data compared to the Naïve Bayes and Logistic Regression. So the Linear SVC is the best classifier for RMSA when the training model is not needed to build several times.

6.2 Scope and Limitations

In this research, the proposed SA is developed with four modules: data collection, preprocessing, class labeling and sentiment classification. The proposed system is implemented on different analytics platforms by using different architectures. However the proposed SA collects and analyzes the data from only social media (twitter). Nowadays, other social media are growing exponentially and provides a distinctive advantage for this research, the proposed SA should be collected and analyzed the data from other social media.

Twitter including vast amounts of information about almost all industries from entertainment to sports, health to business etc. Twitter provides unprecedented access to lawmakers and to celebrities, as well as to news as it's happening. Twitter represents an important data source for the business models of huge companies as

well. However only product (iphone) tweets are collected and analyze them in this research. For developing complete SA system, it is required to implement on other domains.

Hug amount of twitter data is collected even in a specified period. Manual Labeling for those huge amount data is time and labor consuming. Instead of manual labeling, the existing lexicon based classifier (SentiStrength) is used to classify them and the classifier provide just the sentence level classification. For better accuracy of training data, it is necessary to develop the aspect level sentiment classification.

When Real-time analysis is performed, the proposed system implemented on Spark. Actually, Spark doesn't support real-time data stream processing fully because Spark streaming partitioned the live data stream into batches (specified seconds) and it known as Spark RDDs (Resilient Distributed Database). The operations are applied on these RDDs to process them. After processing, the result is again converted into batches. This way, Spark streaming is just a micro-batch processing and does not support full real-time processing. Therefore the system is needed to perform on another real-time analytics platform for fully support of real-time processing.

6.3 Further Research Direction

There are several promising directions pursue in the future. Specifically, the visualization technology can be used for displaying the real-time dashboard based on a real-time processed data, which helps in both decision making and visualization purposed. Because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports. Data visualization is a quick, easy way to convey concepts in a universal manner.

Recent work considers the sentence level sentiment classification on the single domain (iphone) of twitter. For high level performance, the aspect level sentiment classification should be developed in the future research. And the system should be implemented on other domains such as political and education domains. Other social media such as facebook requires brands to effectively reach a target audience, listen to an exponentially growing social community and influence fans' and followers' purchasing decisions. Therefore the proposed SA should be implemented on other

social media. For high performance processing, the proposed SA should be implemented on other analytic platforms.

For real time processing, the proposed SA can be implemented on Storm. Storm run “topologies” while Hadoop run “jobs”. The basic primitives Storm provides for doing stream transformations are “spouts” and “bolts”. Spouts and bolts have interfaces that implement for running the application-specific logic. A spout is a source of streams and a bolt does single-step stream transformations. Bolt creates new streams based on its input streams. Complex stream transformations, like computing a stream of trending topics from a stream of tweets, require multiple steps and thus multiple bolts. Everything in Storm runs in parallel in a distributed way. Spouts and bolts execute as many threads across the cluster, and they pass messages to each other in a distributed way. Messages never pass through any sort of central router, and there are no intermediate queues. A tuple is passed directly from the thread who created it to the threads that need to consume it. Storm supports true streaming processing model through core Storm layer. Storm supports 3 message processing guarantees: at-least once, at-most once and exactly once. Storm’s reliability mechanisms are purely distributed, scalable, and fault-tolerant.

AUTHOR'S PUBLICATIONS

[p1] W.N.Chan and T.Thein, "Sentiment Analysis for Twitter Stream Data by Combining Lexicon and Machine Learning Approaches", In Proceedings of the 15th International Conference on Computer Applications (ICCA, 2017), Yangon, Myanmar, 16-17 February, 2017, pp. 145-151.

[p2] W.N.Chan and T.Thein, "Multi-tier Sentiment Analysis System in Big Data Environment", International Journal of Computer Science and Information Security (IJCSIS), Vol. 15, No. 9, September 2017, pp. 204-221.

[p3] W.N.Chan and T.Thein, "Multi-tier Sentiment Analysis System with Sarcasm Detection: A Big Data Approach", In Proceedings of the 16th International Conference on Computer Applications (ICCA, 2018), Yangon, Myanmar, 22-23 February, 2018, pp. 40-49.

[p4] W.N.Chan and T.Thein, "Sentiment Analysis System in Big Data Environment", International Journal of Computer Science and Information Security (IJCSSE), Vol. 33, No. 4, May 2018.

[p5] W.N.Chan and T.Thein, "A Comparative Study of Machine Learning Techniques for Real-time Multi-tier Sentiment Analysis", 1st IEEE International Conference on Knowledge Innovation and Invention, Jeju, Korea, 23-27 July, 2018, pp. 90-93.

Bibliography

[1] A. Assiri, A.Emam, H.Al-dossari, "Real-Time SA of Saudi Dialect Tweets Using SPARK," In Proceedings of the "IEEE International Conference on Big Data (Big Data)", 5-8 Dec, 2016.

[2] A. Baltas, A.Kanavos, A. K. Tsakalidis, "An Apache Spark Implementation for SA on Twitter Data," ," In Proceedings of the "Algorithmic Aspects of Cloud Computing", April, 2017, pp.15-25.

[3] A. C. Oliver, "Machine-learning-with-mahout". [Online]. Available: http://www.infoworld.com/article/2608418/application-development/enjoy-machine-learning-with-mahout-on-hadoop.html, [Accessed: Dec. 3, 2016].

[4] A. Genkin, D. D. Lewis, D. Madigan, "Sparse Logistic Regression for Text Categorization," 2005.

[5] A. Giachanou and F. Crestani, "Like it or not: A Survey of Twitter Sentiment Analysis Methods," ACM Comput. Surv., vol. 49, no. 2, 2016, pp. 28.

[6] A. Hadian and S. Shahrivari, "High Performance Parallel k-means Clustering for Disk-Resident Datasets on Multi-core CPUs," J. Supercomput. 2014, 69, pp. 845–863.

[7] A. Haripriya, S. Kumari, "Real Time Analysis of Top Trending Event on Twitter: Lexicon Based Approach," In Proceedings of the "8th ICCCNT 2017", India, July 3 - 5, 2017.

[8] A. Ichinose, A. Takefusay, H. Nakadaz, M. Oguchi, "A Study of a Video Analysis Framework Using Kafka and Spark Streaming," In Proceedings of the "IEEE International Conference on Big Data (BIGDATA)", 2017.

[9] A. M. G. Almeida, S. B. Jr, E. C. Paraiso, "Multi-class Emotions classification by Sentic Levels as features in Sentiment Analysis," In Proceedings of the "5th Brazilian Conference on Intelligent Systems", 2016, pp. 486-491.

[10] A. Pak and P. Paroubek: "Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives," In Proceedings of the

"SemEval '10' 5th International Work-shop on Semantic Evaluation", 2010, pp. 436-439.

[11] A. Zimek, F. Buchwald, E. Frank, and S. Kramer, "A Study of Hierarchical and Flat Classification of Proteins," IEEE/ACM Transactions on Computations on Computational biology and Bioinformatics, Vol. 7, No.3, July-September, 2010, pp. 563-571.

[12] Algorithmia, "Introduction to Microservices" [Online]. Available: https://blog.algorithmia.com/introduction-to-microservices/, [Accessed: Dec.11, 2018].

[13] Apache flume, "Flume 1.6.0 user guide" [Online]. Available: https://flume.apache.org/releases/content/1.6.0/FlumeUserGuide.html, [Accessed: Aug. 12, 2016]

[14] Authentication, "Oauth with the Twitter APIs" [Online]. Available https://developer.twitter.com/en/docs/basics/authentication/overview/oauth.html, [Accessed: Jun. 15, 2016].

[15] B. Liu, X. Li, W.S. Lee, and P.S. Yu, "Text Classification by Labeling Words," In Proceedings of the "National Conference on Artificial Intelligence", Menlo Park, CA; Cambridge, MA; London.. MIT Press, 2004, pp. 425-430.

[16] B. Liu, E.Blasch, Y.Chen, D.Shen, G.Chen, "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier," In Proceedings of the " IEEE International Conference on Big Data", 2013, pp. 99-104.

[17] B. Milsom, "Twitter API vs Firehose", [Online]. Available: https://www.echosec.net/twitter-api-vs-firehose/, June 2015; [Accessed: Sep.13, 2016].

[18] B. Shu, H. Chen, M. Sun, "Dynamic Load Balancing and Channel Strategy for Apache Flume Collecting Real-time Data Stream," In Proceedings of the "IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications", 2017, pp. 542-549.

[19] B. Yadranjiaghdam, “Developing A Real-time Data Analytics Framework For Twitter Streaming Data,” Dec, 2016.

[20] C. Chambers, A. Raniwala, F. Perry, S. Adams et al. “Flume Java: Easy, Efficient Data Parallel Pipelines,” ACM Sigplan Not. 2010, 45, pp. 363–375.

[21] C. Dobre, and F. Xhafa, “Parallel Programming Paradigms and Frameworks in Big Data Era,” International Journal of Parallel Programming 42 (5), pp. 710–738.

[22] C. Engle, A. Lupher, R. Xin, M. Zaharia, and M. Franklin, “Fast Data Analysis Using Coarse-grained Distributed Memory,” In Proceedings of the “2012 ACM SIGMOD International Conference on Management of Data”, Scottsdale, AZ, USA, 20–24 May 2012; pp. 689–692.

[23] C. Harvey, “Big Data Pros and Cons,” [Online]. Available: https://www.datamation.com/big-data/big-data-pros-and-cons.html, [Accessed: 25-Oct. 25, 2018].

[24] C. Yang, H. Wu, Q. Huang, Z. Li, and J. Li, “Using Spatial Principles to Optimize Distributed Computing for Enabling the Physical Science Discoveries,” In Proceedings of the “National Academy of Sciences 108 (14)”, pp. 5498–5503.

[25] C. Yang, Y. Xu, and D. Nebert, “Redefining the Possibility of Digital Earth and Geosciences with Spatial Cloud Computing.” International Journal of Digital Earth 6 (4), pp. 297–312.

[26] D. Henschen, “Big Data Platform Comparisons: 3 Key Points,” [Online]. Available: “https://www.informationweek.com/big-data/big-data-analytics/big-data-platform-comparisons-3-key-points/d/d-id/1113860”, [Accessed: Mar. 15, 2017]

[27] D.I. George Amalarethinam, V. Jude Nirmal, “Real-Time Sentiment Prediction on Streaming Social Network Data Using In-Memory Processing”, World Congress on Computing and Communication Technologies (WCCCT), 2016, pp.69-72.

[28] D. Maynard, and M. Greenwood, “Who Cares About Sarcastic Tweets? Investigating the Impact of Sarcasm on Sentiment Analysis,” in Proc. 9th Int.

Conf. Language Resources Evaluation, May 2014, pp. 4238–4243.

[29] D. Michailidis, N. Stylianou, and I. Vlahavas, “Real Time Location Based Sentiment Analysis on Twitter - The AirSent System,“ Rio Patras, Greece, July 9–15, 2018.

[30] D. Roth, “Multiclass Classification,” Oct 27, 2016.

[31] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, “New Avenues in Opinion Mining and Sentiment Analysis,” IEEE Intell. Syst.,” 2013, vol. 28, no. 2, pp.15–21.

[32] E. Falk, V. K. Gurbani, R. State, “Queryable Kafka: An Agile Data Analytics Pipeline for Mobile Wireless Networks,” Proceedings of the VLDB Endowment, Vol. 10, No. 12, 2017, pp. 1646-1657.

[33] E. Haddi, X. Liua, Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis,” Procedia Computer Science 17, 2013, pp. 26 – 32.

[34] E. R. Sparks, A. Talwalkar, V. Smith, J. Kottalam, X. Pan, et al., MLI: an API for Distributed Machine Learning, in Proceedings of the “IEEE 13th International Conference on Data Mining”, Dallas, TX, USA, December7-10, 2013, pp.1187–1192.

[35] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert ve, R. Huang, “Sarcasm as Contrast between A Positive Sentiment and Negative Situation,” In Proceedings of the “International Conference on Empirical Methods in Natural Language Processing”, Seattle, 18-21 October, 2013, pp. 704-714.

[36] F. N. Afrati, V. Borkar, M. Carey, N. Polyzotis, J. D. Ullman, “Map-Reduce Extensions and Recursive Queries”, In Proceedings of the “14th International Conference on Extending Database Technology”, Uppsala, Sweden, 22–24 March 2011; pp. 1–8.

[37] G. B. Orgaz, J. J. Jung,D.Camacho, “Socialbigdata: Recent Achievements and New Challenges,” Information Fusion 28, 2016, pp. 45–59.

[38] G. Goswami, “Effective Image Analysis on Twitter Streaming sing Hadoop Eco System on Amazon Web Service EC2,” International Journal of Advanced Research in Computer Science and Software Engineering,

Volume 5, Issue 9, September 2015.

[39] G. H. John and P. Langley, “Estimating Continuous Distributions in Bayesian Classifiers,” in Proceedings of the “11 th conference on Uncertainty in artificial intelligence,"1995, pp. 338–345.

[40] G. Qiu, B. Liu, J. Bu, C. Chen, “ Expanding Domain Sentiment Lexicon Through Double Propagation,” In Proceedings of the “21st International Jont Conference on Artifical Intelligence (IJCAI’09)”, Pasadena, CA,USA, 11–17 July 2009; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2009; pp. 1199–1204.

[41] H. Nazeer, W. Iqbal, F. Bokhari, F. Bukhari et al. “Real-time Text Analytics Pipeline Using Open-source Big Data Tools”, 12 Dec 2017.

[42] H. Saif, M. Fernandez, Y. He, & H. Alani, "On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter," In Proceedings of the “9th Language Resources and Evaluation Conference (LREC)”, Reykjavik, Iceland, 2014, pp.80-81.

[43] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, “Big Data and its Technical Challenges,” Communications of the ACM 57 (7), pp. 86–94.

[44] H. Yu, V. Hatzivassiloglou, “Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences,” In Proceedings of the “2003 Conference on Empirical Methods in Natural Language Processing (EMNLP’03)”, Sapporo, Japan, 11–12 July 2003, pp. 129–136.

[45] “Hadoop Yarn”, [Online]. Available: https://hadoop.apache.org/docs/r2.7.1/hadoop-yarn/hadoop-yarn-site/YARN.html, [Accessed: Aug. 20, 2016].

[46] I. Ha, B. Back , B. Ahn, “MapReduce Functions to Analyze Sentiment Information from Social Big Data,” International Journal of Distributed Sensor Networks, 2015.

[47] I. Triguero, D. Peralta, J. Bacardit, S. García, and F. Herrera, “MRPR: A MapReduce Solution for Prototype Reduction in Big Data Classification,”

Neurocomputing, 20 February, 2015, pp. 331–345.

[48] IBM, “The Power of One: IBM + Hortonworks Drives Advanced Analytics, [Online]. Available: https://www.ibm.com/analytics/hadoop/big-data-analytics, [Accessed: Oct.22, 2018].

[49] J. Alcaide, R. Justo ve . M. I. Torres, “Combining Statistical and Semantic Knowledge for Sarcasm Detection in Online Dialogues,” in Pattern Recognition and Image Analysis, Yuri I. Zhuravlev, 2015, pp. 662

[50] J. Dean, S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, In Proceedings of the “6th Symposium on Opearting Systems Design & Implementation”, San Francisco, CA, USA, 6–8 December, 2004.

[51] J. Ekanayake, H. Li, B. Zhang, T. Gunarathne, “Twister: A Runtime for Iterative MapReduce”, In Proceedings of the “19th ACM International Symposium on High Performance Distributed Computing”, Chicago, IL, USA, 20–25 June 2010; pp. 810–818.

[52] J. Fan, and H. Liu, “Statistical Analysis of Big Data on Pharmacogenomics,” Advanced Drug Delivery Reviews. 65 (7), pp. 987–1000.

[53] J. M. Hellerstein, “Datalog Redux: Experience and Conjecture,” In Proceedings of the “29 th ACM Symposium on Principles of Database Systems”, 6-11 June, 2010, pp. 1-2.

[54] J. Natkins, “Analyzing Twitter data with Hadoop”, [Online]. Available: http://blog.cloudera.com/blog/2012/10/analyzing-twitter-data-with-hadoop-part-2-gathering-data-with-flume/, [Accessed: Aug.28, 2016].

[55] J. Read, J. Carroll, “Weakly Supervised Techniques for Domain-independent Sentiment Classification,” In Proceedings of the “1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion”, Hong Kong, China, NY, USA, 6 November, 2009, pp. 45–52.

[56] J. Singh, G. Singh and R. Singh, “Optimization of Sentiment Analysis using Machine Learning Classifiers,” Comput. Inf. Sci. (2017) 7: 32, 11 December, 2017.

[57] K. Metzler, D. A. Kim, N. Allum, A. Denman, “Who Is Doing

Computational Social Science? Trends in Big Data Research," 2016.

[58] K. PRASHANTH, K. M. George, N. Park and E.C.Tin, "A Case Study on Vercity In Twitter Data using Oil Company Related Tweets," 2015.

[59] L. Derczynski, A. Ritter, S. Clarke, and K. Bontcheva. "Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data," In Proceedings of the "International Conference on "Recent Advances in Natural Language Processing", 2013.

[60] L. Giura, "Sentiment Analysis using Mahout Naïve Bayes," [Online]. Available: http://technobium.com/sentiment-analysis-using-mahout-Naive - Bayes/, [Accessed: Oct. 20, 2016].

[61] L. Lugnegård, "Building A High Throughput Microscope Simulator Using the Apache Kafka Streaming Framework," 2018.

[62] L. Neumeyer, B. Robbins, A. Nair, A. Kesari, "S4: Distributed Stream Computing Platform," In Proceedings of "IEEE International Conference on Data Mining Workshops", New South Wales, Sydney, 13 December, 2010, pp. 170–177.

[63] L.Velikovich, S.Blair-Goldensohn K.Hannan, R.McDonald, "The Viability of Web-Derived Polarity Lexicons Human Language Technologies," In Proceedings of the "2010 Annual Conference of the North American", June 2010, pages 777–785.

[64] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, B. Liu, "Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis," 21 January, 2011.

[65] M. Bouazizi, T. Ohtsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," IEEE Access, 2016.

[66] M. Bouazizi, T. Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classfication," In Proceedings of the "IEEE ICC 2016 SAC Social Networking", 22-27 May, 2016.

[67] M. Hu, B. Liu, "Mining and Summarizing Customer Reviews," In Proceedings of the "10 th ACM SIGKDDInternational Conference on

Knowledge Discovery and Data Mining", Seattle,WA, USA, 22–25 August 2004;ACM: New York, NY, USA, 2004, pp. 168–177.

[68] M. Isard, M. Budiu, Y. Yu, A. Birrell, D. Fetterly, "Dryad: Distributed Data-parallel Programs from Sequential Building Blocks," SIGOPS Oper. Syst. Rev. 2007, 41, pp. 59–72.

[69] M. Karanasou, A. Ampla, C. Doulkeridis and M. Halkidi, "Scalable and Real-time Sentiment Analysis of Twitter Data," In Proceedings of the "IEEE 16th International Conference on Data Mining Workshops (ICDMW)", 2016, pp. 944-951.

[70] M. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," October, 1993.

[71] M. Moh, A. Gajjala, S.C.R.Gangireddy, T.S.Moh, "On Multi-Tier Sentiment Analysis Using Supervised Machine Learning," In Proceedings of the "IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology", 6-9 December, 2015, pp. 341-344.

[72] M. Skuza, A. Romanowski, "Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction," In Proceedings of the "Federated Conference on Computer Science and Information Systems (FedCSIS)", Poland, 2015, pp.1349-1354.

[73] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based Methods for Sentiment Analysis," Comput. Linguist., vol. 37, no. 2, 2011, pp. 267–307.

[74] M. Thelwall, K. Buckley, G. Paltoglou, "Sentiment Strength Detection for the Social Web," Journal of the American Society for Information Science and Technology, January 2012, pp. 163-173.

[75] M. Whitehead, L. Yaeger, "Sentiment Mining using Ensemble Classification Models," In Innovations and Advances in Computer Sciences and Engineering; Springer Netherlands: Dordrecht, The Netherlands, 2010, pp. 509–514.

[76] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker et al., “Spark: Cluster computing with working sets,” In Proceedings of the “2nd USENIX Conference on Hot Topics in Cloud Computing”, Boston, MA, USA, 22–25 June, 2010, pp. 10.

[77] M. Zaharia, M. Chowdhury, T. Das, A. Dave et al., “Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing”, In Proceedings of the “9th USENIX Conference on Networked Systems Design and Implementation”, San Jose, CA, USA, 25–27 April, 2012.

[78] M. Zaharia, B. Robbins, A. Nair, A, A. Kesari, “S4: Distributed Stream Computing Platform”, In Proceedings of the “IEEE International Conference on Data Mining Workshops (ICDMW)”, New South Wales, Sydney, 13 December, 2010, pp. 170–177.

[79] N. Ammn, and M. Irfanuddin, “Big Data Challenges.” International Journal of Advanced Trends in Computer Science and Engineering 2 (1), pp. 613–615.

[80] N. Garg, Apache Kafka, Birmingham, UK: Packt Publishing, 2013, pp. 88.

[81] N. Khan, I. Yaqoob, I. A. T. Hashem, Z. Inayat, et al., “Big Data: Survey, Technologies, Opportunities, and Challenges,” The Scientific World Journal 2014, pp. 1–18.

[82] N. M. Sharef, H. M. Zin and S. Nadali, “Overview and Future Opportunities of Sentiment Analysis,” Journal of Computer Sciences 2016, 12 (3):pp. 153-168, DOI: 10.3844/jcssp.2016, pp. 153-168.

[83] N. Nasir, Kashif Zafar, Zareen Alamgir, “Sentiment Analysis of Social Media Using MapReduce,” 2017.

[84] N. Nodarakis, S. Sioutas, A. Tsakalidis, G. Tzimas, “MR-SAT: A MapReduce Algorithm for Big Data SA on Twitter,” In Proceedings of the “12th International Conference on Web Information Systems and Technologies (Webist 2016)”, Rome, Italy, 23-25 April, 2016.

[85] O. Kolchyna, T. P. Souza, P. Treleaven, T. Aste, “Twitter Sentiment Analysis: Lexicon Method, Machine Learning Method and Their Combination,” [Online] Available:

https://www.researchgate.net/publication/279864933_Twitter_Sentiment_Analysis_Lexicon_Method_Machine_Learning_Method_and_Their_Combination, [Accessed: Jan.19, 2019].

[86] P. Zikopoulos, C. Eaton, D, DeRoos, T. Deutch and G. Lapis, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data," McGralw-Hill Osborne Media, 2011, pp. 176.

[87] Q. Huang, and C. Xu, "A Data-Driven Framework for Archiving and Exploring Social Media Data," Annals of GIS 20, pp. 265–277.

[88] R. D. Schneider, "Hadoop for Dummies, Special Edition", Published by John Wiley & Sons Canada Ltd, 26 September, 2012.

[89] R. Gonzalez-Ibanez, S. Muresan ve N. Wacholder, "Identifying Sarcasm in Twitter: A Closer Look," In Proceedings of the "49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies", Oregon, 2011.

[90] R. Gulla*, U. Shoaiba, S. Rasheedb, W. Abidb, et al., "Twitter's Data for Opinion Mining in Political," In Proceedings of the "20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems", KES2016, United Kingdom, 5-7 September 2016, pp. 1560-1570.

[91] R. M. Yoo, A. Romano, C. Kozyrakis, "Phoenix rebirth: Scalable MapReduce on a Large-scale Shared-memory System," In Proceedings of "IEEE International Symposium on Workload Characterization", Austin, TX, USA, 4–6 October 2009, pp. 198–207.

[92] R. Ramesh, G. Divya , D. Divya, M. Kurian and V. Vishnuprabha, "Big Data Sentiment Analysis using Hadoop," IJIRST –International Journal for Innovative Research in Science & Technology, vol. 1, no. 1, pp. 92-96, 2015.

[93] S. Chaturvedi, V. Mishra, N. Mishra, "Sentiment Analysis using Machine Learning for Business Intelligence," In Proceedings of the "IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI-2017)", 2017, pp. 2162-2166.

[94] S. Ghemawat, H. Gobioff, and S. T. Leung, " Google File System," In Proceedings of the "19th ACM Symposium on Operating Systems Principles 2003", Bolton Landing, NY, USA, pp. 29-43.

[95] S. Jung and Y. Shin, "Study of the Big Data Collection Scheme Based Apache Flume for Log Collection" International Journal of Computer Theory and Engineering, Vol. 10, No. 3, June 2018.

[96] S. K. Bharti, B. Vachha, R.K. Pradhan, K.S. Babu and S.K. Jena, "Sarcastic Sentiment Detection in Tweets Streamed in Real time: A Big Data Approach," Digital Communications and Networks, 2016.

[97] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins et al. "Big data, Analytics and the Path from Insights to Value. MIT Sloan Management Review," [Online] Available: http://sloanreview.mit.edu/article/big-data-analytics-and-the-path-from-insights-to-value/. 21, [Accessed Dec. 2, 2015].

[98] S. Poria , E. Cambria , R. Bajpai , A. Hussain , "A Review of Affective Computing: from Unimodal Analysis to Multimodal Fusion," Inf. Fus. 37 (2017), 2017, pp. 98–125.

[99] S. R. El-Beltagy and A. Ali, "Open Issues in the Sentiment Analysis of Arabic Social Media: a case study," In Proceedings of the "9th International Conference on Innovations in Information Technology", 2013, pp. 5–220.

[100] S. Seo, E. J. Yoon, J. Kim, et al., "Hama: An Efficient Matrix Computation with the MapReduce Framework," In Proceedings of the "IEEE Second International Conference on Cloud Computing Technology and Science", Indianapolis, USA, 30 December, 2010, pp. 721-726.

[101] S. Shaikh, "Flume Installation and Streaming Twitter Data Using Flume", [Online]. Available: https://www.eduonix.com/blog/bigdata-and-hadoop/flume-installation-and-streaming-twitter-data-using-flume/, [Accessed: Oct. 20, 2016].

[102] S. Suzuki, R. Orihara, Y. Sei, Y. Tahara, A. Ohsuga, "Sarcasm Detection Method to Improve Review Analysis," In Proceedings of the "9th International Conference on Agents and Artificial Intelligence", 2017, pp. 519-526.

[103] “SentiStrength”, [Online]. Available: http://sentistrength.wlv.ac.uk/, [Accessed: Aug. 12, 2016].

[104] “Spark MLlib”, [Online]. Available: https://spark.apache.org/mllib/, [Accessed: Mar. 22, 2018].

[105] “Spark Streaming”, [Online]. Available: https://spark.apache.org/streaming/, [Accessed: Jan.2, 2018].

[106] Storm Homepage. [Online]. Available: http://storm-project.net/, [Accessed: Dec. 14, 2013].

[107] T. Kraska, A. Talwalkar, J. C. Duchi, R. Griffith, et al. “Mlbase: A Distributed Machine Learning System,” In Proceedings of the “6th Biennial Conference on Innovative Data Systems Research”, Asilomar CIDR, CA,USA, January6-9, 2013.

[108] T. White, “Hadoop: The Definitive Guide; O’Reilly Media: Sebastopol”, CA, USA, 2012.

[109] “The Apache OpenNLP Library is A Machine Learning Based Toolkit for the Processing of Natural Language Text,” [Online]. Available: https://opennlp.apache.org/, [Accessed: June-26, 2017]

[110] “Twitter Statistics,” [Online]. Available : http://www.statisticbrain.con/twitter-statistics/ [Accessed: Jan.2, 2013]

[111] V. Atzivassiloglou, K. R. McKeown, “Predicting the Semantic Orientation of Adjectives,” In Proceedings of the “35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics”, Madrid, Spain, 7–12 July 1997; pp. 174–181.

[112] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal et al. “Apache Hadoop Yarn: Yet Another Resource Negotiator,” In Proceedings of the “4th Annual Symposium on Cloud Computing; Santa Clara”, CA, USA, 1–3 October 2013; pp. 5.

[113] V. N .Khuc, C.Shivade, R.Ramnath, J.Ramanathan, “Towards Building

Large-Scale Distributed Systems for Twitter SA," In Proceedings of the "27th Annual ACM Symposium on Applied Computing", Trento, Italy, March 26-30, 2012, pp. 459-464.

[114] V. Tablan, I. Roberts, H. Cunningham, and K. Bontcheva, "GATECloud.net: A Platform for Large-Scale, Open-Source Text Processing on the Cloud," Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 371, 1983, pp. 1–13.

[115] W. Fang, B. He, Q. Luo, N. K. Govindaraju, "Mars: Accelerating MapReduce with Graphics Processors," IEEE Trans. Parallel Distrib. Syst. 2010, 22, pp. 608–620.

[116] W. Jun, W. Wenhao, C. Renfei, "Distributed Data Streams Processing Based on Flume/Kafka/Spark," In Proceedings of the "3rd International Conference on Mechatronics and Industrial Informatics (ICMII 2015)", 2015, pp. 948-952.

[117] "Welcome to Apache Hadoop!," [Online]. Available: http://hadoop.apache.org/, [Accessed: Dec. 28, 2017].

[118] X. Ding, B. Liu, P.S. Yu, "A Holistic Lexicon-based Approach to Opinion Mining," In Proceedings of the "2008 International Conference on Web Search and Data Mining (WSDM'08)", Palo Alto, CA, USA, 11–12 February 2008; ACM: New York, NY, USA, 2008, pp. 231–240.

[119] X. Zhang, X. Zheng, "Comparison of Text Sentiment Analysis Based on Machine Learning," In Proceedings of the "15th International Symposium on Parallel and Distributed Computing", 2016, pp. 23-233.

[120] Y. Arslan, A. Birturk, B. Djumabaev, D. K¨uc,"Real-Time Lexicon-Based Sentiment Analysis Experiments On Twitter With A Mild (More Information, Less Data) Approach," In Proceedings of the "IEEE International Conference on Big Data (BIGDATA)", 2017, pp. 1892-1897.

[121] Y. Bu, B. Howe, M. Balazinska, M. D. Ernst, "HaLoop: Efficient Iterative Data Processing on Large Clusters," In Proceedings of the VLDB Endow, 2010, 3, pp. 285–296.

[122] Y. Lu, M. Castellanos, U. Dayal, C. Zhai, “Automatic Construction of a Context-aware Sentiment Lexicon: An Optimization Approach,” In Proceedings of the “20th International Conference on World Wide Web (WWW’11)”, Hyderabad, India, 30 March 2011; ACM: New York, NY, USA, 2011; pp. 347–356.

[123] Y. Madani, E. Mohammed and B. Jamaa, "A Parallel Semantic Sentiment Analysis," In Proceedings of the “3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)”, Rabat, 2017.

[124] Y. Qi, “Random Forest for bioinformatics,” 2011.

[125] Y. Zhai, Y. S. Ong, and I. W. Tsang. 2014. “The Emerging “Big Dimensionality,” IEEE Computational Intelligence Magazine 9 (3), pp. 14–26.

[126] Z. Jianqiang, "Pre-processing Boosting Twitter Sentiment Analysis?," In Proceedings of the “2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)”, pp. 748-753, 2015.

[127] Z. K. Chen, S. Q. Yang, S. Tan, H. Zhao, L. He, G. Zhang, and H. Y. Yang. 2014b. “The Data Allocation Strategy Based on Load in NoSQL Database,” Applied Mechanics and Materials, pp. 1464–1469.

[128] Z. Li, “Naive Bayes Algorithm For Twitter Sentiment Analysis And Its Implementation In MapReduce,” Diss. University of Missouri Columbia, Dec. 2014.

ACRONYMS

API	Application Program Interface
CPU	Central Processing Unit
CSV	Comma Separated Value
EDW	Enterprise Data Warehouse
ELM	Extended Linear Machine
ETL	Extract, Transform, Load
GPS	Global Positioning System
HDFS	Hadoop Distributed File System
HDP	Hortonworks Data Platform
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IBM	International Business Machines Corporation
IDC	International Data Corporation
ILC	Intelligent Lighting Control
IT	Information Technology
JSON	Javascript Object Notation
KNN	K-Nearest Neighbor
LIWC	Linguistic Inquiry and Word Count
MCCP	Medicare Coordinated Care Plan
ML	Machine Learning
MlLib	Apache Spark's scalable machine learning library
MPP	Massively parallel processing
MR	MapReduce

MSABDP	Multi-tier Sentiment Analysis System on Big Data Analytics Platform
MSASDH	Multi-tier Sentiment Analysis System with Sarcasm Detection on Hadoop
NBC	Naive Bayes classifier
NeuSEP	Neutral Sentiment Phrases
NeuSIP	Neutral Situation Phrases
NLP	Natural Language Processing
NoSQL	Not Only SQL
NSEP	Negative Sentiment Phrases
NSIP	Negative Situation Phrases
OAuth	Open Authorization
OneR	One Rule
OVA	One against All
OVO	One against One
PE	Processing Environment
PMC	Polling Multinomial Classifier
POS	Part-Of-Speech
PSEP	Positive Sentiment Phrases
PSIP	Positive Situation Phrases
RDD	Resilient Distributed Datasets
RFID	Radio-Frequency Identification
RMSA	Real-time Multi-tier Sentiment Analysis System
RSSD	Rule based Sarcasm Sentiment Detection
RT	Retweet
SA	Sentiment Analysis

SBD	Social Big Data
SM	Social Media
SMO	sequential minimal optimization
SNSEP	Strongly_negative Sentiment Phrases
SNSIP	Strongly_negative Situation Phrases
SPSEP	Strongly_positive Sentiment Phrases
SPSIP	Strongly_positive Situation Phrases
SQL	Structured Query Language
SSABDP	Single-tier Sentiment Analysis System on Big Data Analytics Platform
SVM	Support Vector Machines
TF-IDF	Term Frequency-Inverse Document Frequency
UDF	User Define Function
URL	Uniform Resource Locator
VM	Virtual Machine
VSM	Vector Space Model
WAN	Wireless Access Network
WFC	work flow controller
YARN	Yet Another Resource Negotiator

APPENDIX A

CONFIGURATION AND SOFTWARE SETUP

The performance evaluation of SA on Big Data Analytics platform is implemented with Hadoop Multi Node Cluster**.** The cluster is composed of four computing nodes (VMs). The VMs are interconnected via a 1-gigabit Ethernet. The host machine runs Windows 10 and has Intel Core i7-2.90GHz processor, 4GB physical memory, and 950-GB disk. As software components: Hadoop 2.7.1, Mahout 0.10.0, Flume 1.6, Spark 2.2.0 are used. The installation and configuration of software components are presented the following subsection.

Hadoop 2.7 Multi Node Cluster Setup

To install Hadoop cluster on the machines, JAVA, SSH and other software utilities are applied on VMs. Hadoop is a framework written in Java for running applications on large clusters of commodity hardware so Hadoop requires a working Java. Hadoop requires SSH access to manage its different nodes, i.e. remote machines plus the local machine.

JAVA Installation

For java installation, the java package can be downloaded via https://www.oracle.com/technetwork/java/javase/downloads/index.html. The installation steps are described as follows:

*Create a directory and place the java package on the directory. (The location is user's choice)

```
# mkdir –p /usr/lib/jvm
# sudo mv jdk-8u121- linux-x64.tar.gz /usr/lib/jvm
#sudo mv jre-8u121- linux-x64.tar.gz /usr/lib/jvm
```

*Navigate to the directory of java package and unpack the tarball archives to java package.

```
# cd /usr/lib/jvm
# sudo tar zxvf jdk-8u121-linux-x64.tar.gz
# sudo tar zxvf jre-8u121- linux-x64.tar.gz
# sudo rm jdk-8u121- linux-x64.tar.gz
# sudo rm jre-8u121- linux-x64.tar.gz
```

*After unpack the package and then inform Operating System (Ubuntu) where Java installation is located.

```
# sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0_121/bin/javac" 1
# sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jre1.8.0_121/bin/java" 1
# sudo update-alternatives --set "javac" "/usr/lib/jvm/jdk1.8.0_121/bin/javac"
# sudo update-alternatives --set "java" "/usr/lib/jvm/jre1.8.0_121/bin/java"
```

*And then update the system-wide path, reload the path and test the installation.

```
# sudo echo "JAVA_HOME=/usr/lib/jvm/jdk1.8.0_121" >> /etc/profile
# sudo echo "PATH=$PATH:$JAVA_HOME/bin" >> /etc/profile
# sudo echo "export JAVA_HOME" >> /etc/profile
# sudo echo "export PATH" >> /etc/profile
# . /etc/profile
# java –version
# javac –version
```

SSH Installion and Configuration

Hadoop control scripts rely on SSH to perform cluster-wide operations. It requires SSH access to manage its nodes. The installation steps and Configurations steps are described as follows:

*Install ssh and generate an RSA key pair Copy the public key (~/.ssh/id_rsa.pub) content and append to the file Try ssh on localhost.

```
apt-get install ssh
#which ssh
#which sshd
#which ssh-keygen
# ssh-keygen –t rsa –P “”
~/.ssh/authorized_keys
cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
# ssh localhost
```

*In hadoop, the master’s public key should be added to all the slaves’ ~/.ssh/authorized_keys file, so that master can easily communicate to all the slaves. Master public key file is id_rsa.pub and open it with text editor and copy the contents. Paste the contents of id_rsa.pub into it.

```
# nano ~/.ssh/authorized_keys
# ssh slave
```

Hadoop Installation

To install hadoop, download the hadoop distribution tar gz file via "http://www.eu.apache.org/dist/hadoop/common/". The installation steps are described as follows:

*Extract Hadoop tar file and change the owner and mode of folder.

```
# tar –xvzf hadoop-2.7.1.tar.gz
# chown –R 777 hadoop-2.7.1
# chmod –R 777 hadoop-2.7.1
```

* Edit configuration file "hadoop-env.sh" (located in HADOOP_HOME/etc/hadoop) and set JAVA_HOME.

```
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_121
```

*Set JAVA_HOME and HADOOP_HOME as environment variable in bashrc file.

```
# nano ~/.bashrc
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_121
export HADOOP_HOME=/home/hadoop/hadoop-2.7.1
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
#source ~/.bashrc
#hadoop version
```

*Create a base directory (/var/opt/hadoop/cluster) for hadoop to store dfs and mapreduce data.

```
# cd /var/opt
# mkdir hadoop
# cd hadoop
# mkdir cluster
```

*Change the name of virtual machine

```
#nano /etc/host/hostname
delete localhost
change master (or) slave1 (or) slave2 (or) slave3
```

*In MapReduce+HDFScluster, server1 serves as master and slave, and the other servers serve as slaves. Connect all slaves from the master through ssh.

```
# ssh localhost
# ssh slave1
# ssh slave2
# ssh slave3
```

*Edit hadoop configuration file of masters (only in master e.g server1 that run NameNode). This file is used for Secondary NameNode.

```
root@ubuntu:/# gedit /home/hadoop/hadoop-2.7.1/conf/masters
server1
```

*Edit config file /home/hadoop/hadoop-2.7.1/conf/slaves (only in master e.g server1 that run NameNode and JobTracker)

```
root@ubuntu:/# gedit /home/hadoop/hadoop-1.1.2/conf/slaves
server1
slave1
slave2
slave3
```

*Configure core-site.xml in the hadoop configuration directory. (all machines)

```
<configuration>
<property>
<name>fs.defaultFS</name>
<value>hdfs://server1:9000</value>
</property>
</configuration>
```

*Configure mapred-site.xml in the hadoop configuration directory. (all machines)

```
<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
<property>
<name>mapreduce.job.tracker</name>
<value>HadoopServer1:9001</value>
</property>
<property>
<name>mapred.child.java.opts</name>
<value>-Xmx4096m</value>

</property>
```

*Configure yarn-site.xml in the hadoop configuration directory. (all machines)

```
<property>
<name>
yarn.resourcemanager.hostname</name>
<value>server1</value>
</property>
<property>
```

```
<name>
yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>
yarn.nodemanager.vmem-check-enabled</name>
<value>false</value>
</property>
<property>
<name>
yarn.nodemanager.vmem-pmem-ratio</name>
<value>4</value>
</property>
```

*Configure hdfs-site.xml in hadoop configuration file. (all machines)

```
<property><name>dfs.replication</name>
<value>4</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>file:///home/hadoop/hadoopdata/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.name.dir</name>
<value>file:///home/hadoop/hadoopdata/hdfs/datanode</value>
</property>
<property>
<name>dfs.namenode.checkpoint.dir</name>
<value>file:///home/hadoop/hadoopdat/hdfs/secondarynamenode</value>
</property>
<property>
<name>dfs.permissions.enabled</name>
<value>false</value>
</property>

</configuration>
```

*Format the HDFS namenode and r*un this command on master machine.*

```
#  hadoop namenode –format
```

*Start and stop the HDFS nodes and MapReduce daemons.

```
#./start-all.sh
#./stop-all.sh
```

*Check whether the expected Hadoop processes are running. (Master Machine)

root@master:/# jps

Response:

2287 TaskTracker
2149 JobTracker
1938 DataNode
2085 Secondary NameNode
2349 jps
1788 NameNode

*Check whether the expected Hadoop processes are running. (Slave1 Machine)

root@slave1:/# jps

Response:

2287 TaskTracker
1938 DataNode
2349 jps

*Check whether the expected Hadoop processes are running. (Slave2 Machine)

root@slave2:/# jps

Response:

2287 TaskTracker
1938 DataNode
2349 jps

*Check whether the expected Hadoop processes are running. (Slave3 Machine)

root@slave3:/# jps

Response:

2287 TaskTracker
1938 DataNode
2349 jps

*Browse the Hadoop MapReduce application using Web-UI to view the status screen.

http://server1_IP:50070 (namenode's HTTP server address and port)
http://server1_IP:50090(secondary namenode's HTTP server address and port)
http:server1_IP:19888(MapReduce job history server's address and port)
htpp://server1_IP:8080(resource manager's http server address and port

*Run a sample MapReduce application via the command line.

```
# hadoop fs –put /home/hadoop/test.txt test.txt
# hadoop jar hadoop-2.7.1-examples.jar wordcount test.txt output
```

Mahout Installation

To install Mahout, mahout distribution tar gz file can be downloaded via "http://ftp.wayne.edu/apache/mahout/". The installation steps are presented as follows:

*Create directory for mahout distribution. Extract and Move unzip folder into the directory

```
# sudo tar -zxvf mahout-distribution-0.10.0.tar.gz.
# sudo mv mahout-distribution-0.10.0 /usr/lib/mahout
```

*Set environment variables into the bashrc file by adding the following lines into it.

```
# sudo gedit ~/.bashrc
export MAHOUT_HOME=/usr/lib/mahout
# source ~/.bashrc
```

Apache Flume Installation

To install Apache Flume, Download release of apache flume binary distribution via "http://flume.apahce.org/download.html". The installation steps are describes as follows:

*Before installing the package, and necessary to run it to install the latest updates. And then flume directory is created and change the ownership and permissions of the directory.

```
# apt-get update
# sudo mkdir /usr/local/flume
# sudo chown -R hduser /usr/local/flume
# sudo chmod -R 755 /usr/local/flume
```

*Untar the apache-flume-1.6.0-bin.tar.gz file. Move the contents of apache-flume-1.6.0-bin folder to flume directory.

```
# tar xzf apache-flume-1.6.0-bin.tar.gz
# mv apache-flume-1.6.0-bin/* /usr/local/flume
```

*Set environment variable in bashrc file by adding the flume path.

```
# sudo gedit $HOME/.bashrc
export FLUME_HOME=/usr/local/flume
PATH=$PATH:$FLUME_HOME/bin
export CLASSPATH=$CLASSPATH:$FLUME_HOME/lib/*:.
source $HOME/.bashrc
```

*Change the directory to the location of flume configuration file. Copy the default "flume-env.sh.template" to "flume-env.sh". Add the java location and twitter classpath into the flume-env.sh file. Copy the default "flume-conf.properties.template" to "flume-conf.properties".

```
# cd $FLUME_HOME/conf
# cp flume-env.sh.template flume-env.sh
# gedit flume-env.sh
export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_121
FLUME_CLASSPATH="/usr/local/flume/lib/flume-sources-twitter-
json-0.1.jar"
cp flume-conf.properties.template flume-conf.properties
```

*Verify the flume installation with the following command on terminal.

```
# ./flume-ng version
```

*Configure the twitter source, HDFS sink and memory channel in flume configuration file. Sample flume configuration file for HDFS sink is described as follows.

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = HDFS
TwitterAgent.sources.Twitter.type =
org.flume.source.twitter.json.TwitterSource
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sources.Twitter.consumerKey=XXXXXXXXX
TwitterAgent.sources.Twitter.consumerSecret = XXXXXXXXX
TwitterAgent.sources.Twitter.accessToken = XXXXXXXXX
TwitterAgent.sources.Twitter.accessTokenSecret = XXXXXXXXX
TwitterAgent.sources.Twitter.keywords = iphone
TwitterAgent.sources.Twitter.languages = en
TwitterAgent.sinks.HDFS.channel = MemChannel
TwitterAgent.sinks.HDFS.type = hdfs
```

```
TwitterAgent.sinks.HDFS.hdfs.path=hdfs://localhost:8020/user/flume
_samsung/tweet/%Y/%m/%d/%H/
TwitterAgent.sinks.HDFS.hdfs.useLocalTimeStamp = true
TwitterAgent.sinks.HDFS.hdfs.fileType = DataStream
TwitterAgent.sinks.HDFS.hdfs.writeFormat = Text
TwitterAgent.sinks.HDFS.hdfs.batchSize = 10000
TwitterAgent.sinks.HDFS.hdfs.rollSize = 0
TwitterAgent.sinks.HDFS.hdfs.rollCount = 10000
TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 10000
```

*Configure the twitter source, Aviro sink and memory channel in flume configuration file. Sample flume configuration file for HDFS sink is described as follows.

```
TwitterAgent.sources = Twitter
TwitterAgent.channels = MemChannel
TwitterAgent.sinks = avroSink
TwitterAgent.sources.Twitter.type =
org.flume.source.twitter.json.TwitterSource
TwitterAgent.sources.Twitter.consumerKey=XXXXXXXX
TwitterAgent.sources.Twitter.consumerSecret=XXXXXXXXX
TwitterAgent.sources.Twitter.accessToken=XXXXXXXXX
TwitterAgent.sources.Twitter.accessTokenSecret=XXXXXXXXX
TwitterAgent.sources.Twitter.keywords=iphone
TwitterAgent.sources.Twitter.languages = en
TwitterAgent.sinks.avroSink.type = avro
TwitterAgent.sinks.avroSink.batch-size = 1
TwitterAgent.sinks.avroSink.hostname = server1
TwitterAgent.sinks.avroSink.port = 4141
TwitterAgent.channels.MemChannel.type = memory
TwitterAgent.channels.MemChannel.capacity = 10000
TwitterAgent.channels.MemChannel.transactionCapacity = 100
TwitterAgent.sources.Twitter.channels = MemChannel
TwitterAgent.sinks.avroSink.channel = MemChannel
```

*Run flume configuration file for Twitter Agent.

```
Flume-ng agent –n TwitterAgent –f /usr/local/flume/conf/flumespark.conf
```

Spark Installation Steps

To install spark on hadoop cluster, download spark distribution via "https://spark.apache.org/downloads.html". The installation steps are described as follows:

*Create Spark directory and set permissions.

```
sudo mkdir /usr/local/spark
sudo chown -R sparkuser /usr/local/spark
sudo chmod -R 755 /usr/local/spark
```

*Extract and Move Files to the directory.

```
sudo tar xzf spark-2.2.0-bin-hadoop2.7.tgz
sudo tar xzf scala-2.11.tgz
sudo mv spark-2.2.0-bin-hadoop2.7/* /usr/local/spark
sudo mv scala-2.11.7/* /usr/local/scala
```

*Set Environment properties in bashrc and reload the environment.

```
# sudo gedit $HOME/.bashrc
export SCALA_HOME=/usr/local/scala
export SPARK_HOME=/usr/local/spark
export
PATH=$SPARK_HOME/bin:$JAVA_HOME/bin:$SCALA_HOME/b
in:$PATH
# source $HOME/.bashrc
```

*Switch to the spark configuration directory. In this directory, copy "spark-env.sh.template" file to "spark-env.sh" and add Properties to the "spark-env.sh" file.

```
#cd /usr/local/spark/conf
#sudo cp spark-env.sh.template spark-env.sh
#sudo gedit spark-env.sh
export SCALA_HOME=/usr/local/scala
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export SPARK_WORKER_MEMORY=1g
export SPARK_WORKER_INSTANCES=2
export SPARK_MASTER_IP=127.0.0.1
export SPARK_MASTER_PORT=7077
export SPARK_WORKER_DIR=/app/spark/tmp
```

* In the spark configuration directory, copy "spark-defaults.conf.template" file to "spark-defaults.conf ". And then properties is added to "spark-defaults.conf " file.

```
#sudo cp spark-defaults.conf.template spark-defaults.conf
#sudo gedit spark-defaults.conf
spark.master                    spark://spark_installed_IP:7077
```

*In the spark configuration directory, copy “slaves.template” file to “slaves“ file. And then properties is added to “slave” file.

```
#sudo cp slaves.template slaves
#sudo gedit slaves
slave1//for salve1 machine
slave2// for salve2 machine
```

*Run Spark with start and stop command.

```
#./start-all.sh
#./stop-all.s
```

* Run a sample Spark application via the command line.

```
# ./bin/spark-submit --master <master-url> --deploy-mode cluster --class
<main-class> <application-jar>
```

*Browse the Spark application using Web-UI to view the status screen.

```
http://spark_installed_IP:8080
```

APPENDIX B

EXPERIMENT RESULTS OF PROPOSED SA IN BIG DATA ENVIRONMENT

The following tables show the experiment results of SA in Big Data Environment. The experiment is conducted with different techniques and different architectures on twitter data set (offline and online). There are preprocessing techniques, sarcasm detection approach, two different architectures (Single-tier and Multi-tier), and three different machine learning techniques (Naïve Bayes, Logistic Regression and Linear SVC).

Sample Tweet Stream Data

{"filter_level":"low","retweeted":false,"in_reply_to_screen_name":null,"truncated":false,"lang":"en","in_reply_to_status_id_str":null,"id":783222637170262018,"in_reply_to_user_id_str":null,"timestamp_ms":"1475569802931","in_reply_to_status_id":null,"created_at":"Tue Oct 04 08:30:02 +0000 2018","favorite_count":0,"place":null,"coordinates":null,"text":"RT @hankypanty: Updated software on my old iPhone.\nIt's now slow/hanging.\nYes, @Apple - I fell for your scam of forcing me to buy a new pho\u2026","contributors":null,"retweeted_status":{"filter_level":"low","contributors":null,"text":"Updated software on my old iPhone.\nIt's now slow/hanging.\nYes, @Apple - I fell for your scam of forcing me to buy a new phone.\n\nAn Android.","geo":null,"retweeted":false,"in_reply_to_screen_name":null,"truncated":false,"lang":"en”,"name":"Apple","indices":[63,69],"screen_name":"Apple","id_str":"380749300"}]},"retweet_count":2,"in_reply_to_user_id":null,"favorite_count":22,"id_str":"783220031928823808","place":null,"user":{"location":"Mumbai","default_profile":false,"profile_background_tile":false,"statuses_count":34921,"lang":"en","profile_link_color":"0084B4”,"following":null,"protected":false,"favourites_count":2467,"description":"Comedian.Author of 'Under Delhi'. Founder of East India Comedy. Yet another Feminist Indian Male :).\n\nOnly for bookings: pantonfirecomedy@gmail.com.(Direct.)","contributors_enabled":false,"profile_sidebar_border_color":"A8C7F7","name":"SorabhPant","profile_background_color":"022330","created_at":"Sat Dec 05 11:31:12+0000 2009","default_profile_image":false,"followers_count":427379,geo_enabled":true,"follow_request_sent":null,"url":http://www.facebook.com/SorabhPant,retweet_count":0,"id_str":"783222637170262018","user":{"location":"Mumbai,India","default_profile":true,"profile_background_tile":false,"statuses_count":8246,"lang":"en”,"favourites_count":0,"profile_text_color":"333333","verified":false,"description":"A devil's mind with an idle workshop. Coffee connoisseur. Tweets intended for humor, to be taken with a pinch of salt, offensive only to those looking for one.”,"follow_request_sent":null,"url":null,"utc_offset":19800,"time_zone":"NewDelhi","notifications":null,"profile_use_background_image":true,"friends_count":134,"profile_sidebar_fill_color":"DDEEF6","screen_name":"salilmp","id_str":"100968460","listed_count":8,"is_translator":false}}

Sample Text Element in Tweet Stream Data

No	Sample Texts (without preprocessing)
1	"iphone 7 has the worst battery life compared to rivals." https:,,t.co,pkk2dr7fyo
2	my ass would cry if my iphone was in the blender 💀😷 https:,,t.co,oqsyezztj4
3	iphone 6 is about to be smashed by a hammer. piece of junk! 7 plus is finally here!
4	America can not make an iphone that does not explode but can make a transformer car
5	I have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy
6	i looooooooove this iphone update because i can send my mom really cute gifs of luke bryan☺□😍
7	finally back w an iphone and getting my life back together lol
8	number of product is limited, do not miss your chance! iphone 7, plus $ 670 https:,,t.co,ouhxjtahwp
9	iphone 8 special touch-id details, best battery life stats & more - pocketnow daily: https:,,t.co,x5cbhivwiz #yt #pocketnowvideo
10	Thanks for amazing iphone with it’s worst battery-life performance!!!!”
11	I love amazing new iphone because it runs awfully
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple's hardware desi…
13	iphone 7+ has been pre-ordered since day 1 and it still isn't here. now my iphone 6 has died and won't turn back on.
14	so i got the iphone 7 today😇
15	rt @tldtoday: so, uh. i’m giving away 12 iphone 7s….. rt if you’re in! https:,,t.co,xah5dppi3y https:,,t.co,ltaz8tdgwf
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc https:,,t.co,sszkolvr73 https:,,t.co,tgoa0l5e7b

Comparative Results of SentiStrength And Manual Classification for Tenary Class

No	Sample Texts (with preprocessing)	SentiStrength Classification	Manual Classification
1	Iphone7 scores worst battery-life performance among flagship smartphones: the lg g5 with 1759 minutes an urlinksymbol	negative	negative
2	My ass would cry if my iphone was in the blender urlinksymbol	negative	negative
3	iphone 6 is about to be smashed by a hammer piece of junk 7 plus is finally here	negative	negative
4	I do_not like iphone 7 because it has the worst battery life compared to rivals urlinksymbol	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	negative	neutral
6	i loove this iphone update because i can send my mom really cute gifs of luke bryan	positive	positive
7	finally back w an iphone and getting my life back together laugh out loud	positive	positive
8	number of product is limited do not miss your chance iphone 7 plus 670 urlinksymbol	positive	neutral
9	iphone 8 special touch id details best battery life stats amp more pocketnow daily urlinksymbol yt pocketnowvideo	positive	positive
10	thank for amazing iphone with it s worst battery life performance	neutral	sarcasm
11	i love amazing new iphone because it runs awfully	neutral	sarcasm
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple s hardware desi	neutral	neutral
13	iphone 7 has been pre ordered since day 1 and it still is not here now my iphone 6 has died and will not turn back on	neutral	neutral
14	so i got the iphone 7 today	neutral	neutral
15	rt usermentionsymbol so uh i m giving away 12 iphone 7s rt if you are in urlinksymbol urlinksymbol	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc urlinksymbol urlinksymbol	neutral	neutral

Comparative Results of SentiStrength And Manual Classification for Multi-class

No	Sample Texts (with preprocessing)	SentiStrength Classification	Manual Classification
1	"iphone 7 has the worst battery life compared to rivals." https:,,t.co,pkk2dr7fyo	negative	negative
2	my ass would cry if my iphone was in the blender 💀😷 https:,,t.co,oqsyezztj4	Strongly_ negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer. piece of junk! 7 plus is finally here!	negative	negative
4	I do_not like iphone 7 because it has the worst battery life compared to rivals urlinksymbol	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	negative	neutral
6	i loove this iphone update because i can send my mom really cute gifs of luke bryan☺□😍	Strongly_ positive	Strongly_ positive
7	finally back w an iphone and getting my life back together lol	positive	positive
8	number of product is limited, do not miss your chance! iphone 7, plus $ 670 https:,,t.co,ouhxjtahwp	positive	neutral
9	iphone 8 special touch-id details, best battery life stats & more - pocketnow daily: https:,,t.co,x5cbhivwiz #yt #pocketnowvideo	Strongly_ positive	Strongly_ positive
10	Thanks for amazing iphone with it’s worst battery-life performance!!!!”	neutral	sarcasm
11	I love amazing new iphone because it runs awfully	neutral	sarcasm
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple's hardware desi…	neutral	neutral
13	iphone 7+ has been pre-ordered since day 1 and it still isn't here. now my iphone 6 has died and won't turn back on.	neutral	neutral
14	so i got the iphone 7 today😊	neutral	neutral
15	rt @tldtoday: so, uh. i’m giving away 12 iphone 7s….. rt if you’re in! https:,,t.co,xah5dppi3y https:,,t.co,ltaz8tdgwf	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc https:,,t.co,sszkolvr73 https:,,t.co,tgoa0l5e7b	neutral	neutral

Comparative Classification Results of Mahout Naïve Bayes Classifier (With and Without Preprocessing)

No	Sample Texts (with preprocessing)	Without Preprocessing	With Preprocessing
1	"iphone 7 has the worst battery life compared to rivals." https:,,t.co,pkk2dr7fyo	negative	negative
2	my ass would cry if my iphone was in the blender 💀😷 https:,,t.co,oqsyezztj4	Strongly_ negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer. piece of junk! 7 plus is finally here!	negative	negative
4	I do_not like iphone 7 because it has the worst battery life compared to rivals urlinksymbol	positive	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	neutral	neutral
6	i loove this iphone update because i can send my mom really cute gifs of luke bryan☺□😍	neutral	Strongly_ positive
7	finally back w an iphone and getting my life back together lol	positive	positive
8	number of product is limited, do not miss your chance! iphone 7, plus $ 670 https:,,t.co,ouhxjtahwp	neutral	neutral
9	iphone 8 special touch-id details, best battery life stats & more - pocketnow daily: https:,,t.co,x5cbhivwiz #yt #pocketnowvideo	Strongly_ positive	Strongly_ positive
10	Thanks for amazing iphone with it’s worst battery-life performance!!!!”	neutral	neutral
11	I love amazing new iphone because it runs awfully	neutral	neutral
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple's hardware desi…	neutral	neutral
13	iphone 7+ has been pre-ordered since day 1 and it still isn't here. now my iphone 6 has died and won't turn back on.	neutral	neutral
14	so i got the iphone 7 today😊	neutral	neutral
15	rt @tldtoday: so, uh. i’m giving away 12 iphone 7s….. rt if you’re in! https:,,t.co,xah5dppi3y https:,,t.co,ltaz8tdgwf	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc https:,,t.co,sszkolvr73 https:,,t.co,tgoa0l5e7b	neutral	neutral

Comparative Results of Sentiment Classification With Sarcasm and Without Sarcasm Detection for Multi-class

No	Sample Texts (with preprocessing)	Without Sarcasm	With Sarcasm
1	Iphone7 scores worst battery-life performance among flagship smartphones: the lg g5 with 1759 minutes an urlinksymbol	negative	negative
2	My ass would cry if my iphone was in the blender urlinksymbol	Strongly_ negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer piece of junk 7 plus is finally here	negative	negative
4	I do_not like iphone 7 because it has the worst battery life compared to rivals urlinksymbol	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	negative	negative
6	i loove this iphone update because i can send my mom really cute gifs of luke bryan	Strongly_ positive	Strongly_ positive
7	finally back w an iphone and getting my life back together laugh out loud	positive	positive
8	number of product is limited do not miss your chance iphone 7 plus 670 urlinksymbol	neutral	neutral
9	iphone 8 special touch id details best battery life stats amp more pocketnow daily urlinksymbol yt pocketnowvideo	Strongly_ positive	Strongly_ positive
10	thank for amazing iphone with it s worst battery life performance	neutral	sarcasm
11	i love amazing new iphone because it runs awfully	neutral	sarcasm
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple s hardware desi	neutral	neutral
13	iphone 7 has been pre ordered since day 1 and it still is not here now my iphone 6 has died and will not turn back on	neutral	neutral
14	so i got the iphone 7 today	neutral	neutral
15	rt usermentionsymbol so uh i m giving away 12 iphone 7s rt if you are in urlinksymbol urlinksymbol	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc urlinksymbol urlinksymbol	neutral	neutral

Comparative Classification Results of Different Architectures (Single-tier and Multi-tier) for Naïve Byes Classifier (for Real-time Multi-lass)

No	Sample Texts (with preprocessing)	Single-tier	Multi-tier
1	Iphone7 scores worst battery-life performance among flagship smartphones: the lg g5 with 1759 minutes an urlinksymbol	negative	negative
2	My ass would cry if my iphone was in the blender urlinksymbol	negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer piece of junk 7 plus is finally here	negative	negative
4	I do_not like iphone 7 because it has the worst battery life compared to rivals urlinksymbol	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	neutral	neutral
6	i loove this iphone update because i can send my mom really cute gifs of luke bryan	positive	Strongly_ positive
7	finally back w an iphone and getting my life back together laugh out loud	positive	positive
8	number of product is limited do not miss your chance iphone 7 plus 670 urlinksymbol	neutral	neutral
9	iphone 8 special touch id details best battery life stats amp more pocketnow daily urlinksymbol yt pocketnowvideo	Strongly_ positive	Strongly_ positive
10	thank for amazing iphone with it s worst battery life performance	neutral	neutral
11	i love amazing new iphone because it runs awfully	neutral	neutral
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple s hardware desi	neutral	neutral
13	iphone 7 has been pre ordered since day 1 and it still is not here now my iphone 6 has died and will not turn back on	neutral	neutral
14	so i got the iphone 7 today	neutral	neutral
15	rt usermentionsymbol so uh i m giving away 12 iphone 7s rt if you are in urlinksymbol urlinksymbol	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc urlinksymbol urlinksymbol	neutral	neutral

Comparative Classification Results of Different Architectures (Single-tier and Multi-tier) for Linear SVC Classifier (for Real-time Multi-lass)

No	Sample Texts (with preprocessing)	Single-tier	Multi-tier
1	"iphone 7 has the worst battery life compared to rivals." https:,,t.co,pkk2dr7fyo	negative	negative
2	my ass would cry if my iphone was in the blender 💀😂 https:,,t.co,oqsyezztj4	Strongly_ negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer. piece of junk! 7 plus is finally here!	negative	negative
4	america cant make an iphone that doesnt explode but can make a transformer car	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	neutral	neutral
6	i love this iphone update because i can send my mom really cute gifs of luke bryan☺□😍	Strongly_ positive	Strongly_ positive
7	finally back w an iphone and getting my life back together lol	positive	positive
8	number of product is limited, do not miss your chance! iphone 7, plus $ 670 https:,,t.co,ouhxjtahwp	neutral	neutral
9	iphone 8 special touch-id details, best battery life stats & more - pocketnow daily: https:,,t.co,x5cbhivwiz #yt #pocketnowvideo	Strongly_ positive	Strongly_ positive
10	Thanks for amazing iphone with it’s worst battery-life performance!!!!”	neutral	neutral
11	I love amazing new iphone because it runs awfully	neutral	neutral
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple's hardware desi…	neutral	neutral
13	iphone 7+ has been pre-ordered since day 1 and it still isn't here. now my iphone 6 has died and won't turn back on.	neutral	neutral
14	so i got the iphone 7 today😂	neutral	neutral
15	rt @tldtoday: so, uh. i’m giving away 12 iphone 7s….. rt if you’re in! https:,,t.co,xah5dppi3y https:,,t.co,ltaz8tdgwf	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc https:,,t.co,sszkolvr73 https:,,t.co,tgoa0l5e7b	neutral	neutral

Comparative Classification Results of Different Architectures (Single-tier and Multi-tier) for Logistic Regression Classifier (for Real-time Multi-lass)

No	Sample Texts (with preprocessing)	Single-tier	Multi-tier
1	"iphone 7 has the worst battery life compared to rivals." https:,,t.co,pkk2dr7fyo	negative	negative
2	my ass would cry if my iphone was in the blender 💀😷 https:,,t.co,oqsyezztj4	Strongly_ negative	Strongly_ negative
3	iphone 6 is about to be smashed by a hammer. piece of junk! 7 plus is finally here!	negative	negative
4	america cant make an iphone that doesnt explode but can make a transformer car	negative	negative
5	i have been typing the longest word document ever on my fucking iphone because my laptop is still in london i am going crazy	neutral	neutral
6	i love this iphone update because i can send my mom really cute gifs of luke bryan☺□😍	Strongly_ positive	Strongly_ positive
7	finally back w an iphone and getting my life back together lol	positive	positive
8	number of product is limited, do not miss your chance! iphone 7, plus $ 670 https:,,t.co,ouhxjtahwp	neutral	neutral
9	iphone 8 special touch-id details, best battery life stats & more - pocketnow daily: https:,,t.co,x5cbhivwiz #yt #pocketnowvideo	Strongly_ positive	Strongly_ positive
10	Thanks for amazing iphone with it’s worst battery-life performance!!!!”	neutral	neutral
11	I love amazing new iphone because it runs awfully	neutral	neutral
12	pixel presentation highlighting minor ways the phone differs from iphone demonstrates how influential apple's hardware desi…	neutral	neutral
13	iphone 7+ has been pre-ordered since day 1 and it still isn't here. now my iphone 6 has died and won't turn back on.	neutral	neutral
14	so i got the iphone 7 today😊	neutral	neutral
15	rt @tldtoday: so, uh. i’m giving away 12 iphone 7s….. rt if you’re in! https:,,t.co,xah5dppi3y https:,,t.co,ltaz8tdgwf	neutral	neutral
16	bluetooth a2dp audio music receiver adapter for iphone ipod 30 pin dock sp e0xc https:,,t.co,sszkolvr73 https:,,t.co,tgoa0l5e7b	neutral	neutral