Multi-tier Sentiment Analysis System with Sarcasm Detection: A Big Data Approach

Wint Nyein Chan[1] and Thandar Thein[2]
[1]University of Computer Studies, Yangon and [2]University of Computer Studies, Maubin
E-mail: [1]wintnyeinchan2012@gmail.com; [2]thandartheinn@gmail.com

Abstract

Social Media is one of the generating sources of big data and analyzing social big data can provide the valuable information. For analyzing the social big data in an efficient and timely manner, the traditional analytic platform is needed to be scaled up. The powerful technique is necessary to extract the valuable information from social big data. Sentiment Analysis can facilitate valuable information by extracting public opinions. The presence of sarcasm, an interfering factor that can flip the sentiment of the given text, is one of the challenges of Sentiment Analysis. In this paper, Multi-tier Sentiment Analysis 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 learning based sentiment classification with Multi-tier architecture. The large amount of tweets is collected by Apache Flume and it is used for system evaluation. The evaluation results show that detecting sarcasm can enhance the accuracy of Sentiment Analysis. Moreover, the results show that the MSASDH is efficient and scalable by decreasing the processing time when adding more nodes into the cluster.

Keywords: *Big Data, Hadoop, Machine Learning, Sarcasm, Sentiment Analysis, Tweets*

1. Introduction

Due to the rapid growth of the Internet and online activity, Social Media are popular as these services allow users to create and share information within open and closed communities. Every time a large volume of both structured and unstructured data are being generated from Social Media. Social Media are one of generating sources of Big Data and it can offer business insights by analyzing the public opinions. Twitter is one of the popular social media, which combines features of blogs and social network services. Twitter was established in 2006 and experienced rapid growth of users in the first years [13]. Twitter has 330 million monthly active users [9] and they posts 500 million tweets every day. Twitter is a good source of information in the sense of snapshots of moods and feelings as well as up-to-date events and current situation commenting.

With the large volume of data being generated from Twitter, this leads to some challenges like processing on large data sets, extraction of useful information from online generated data sets etc. Big Data Analytics has become popular for analyzing and managing large volume of the structure and unstructured data [6]. Hadoop is a good platform 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 is one of the main agenda in big data and the current Sentiment Analysis only focus on 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 Sentiment Analysis.

In this work, Sentiment Analysis system is proposed to extract valuable information from social data. The main contributions of this paper are as follows:

1. Big Data Analytics platform is implemented and the proposed Sentiment Analysis system is developed on Big Data Analytics Platform for analyzing large volumes of tweets in an efficient and timely manner.
2. Instead of manually labeling the class, Rule based Sarcasm-Sentiment Detection scheme (RSSD) is developed to label the class.
3. To improve the classification accuracy, the proposed Sentiment Analysis system is implemented by conducting RSSD and distributed learning based classification with multi-tier architecture.

The remainder of this paper is arranged as follows: In section 2, the related work presents large-scale Sentiment Analysis in a distributed environment and sarcasm detection approaches on tweets. Section 3 describes the implementation of Big Data Analytics platform. In Section 4, the process flow diagram of MSASDH is illustrated and detail functions of each process are presented. Section 5 presents the experimental setup and results discussions. Section 6 presents some concluding remarks and discusses the possible directions for future works.

2. Related Work

In this section the literature review is done on two folds. At first, Sentiment Analyses in distributed environment are reviewed and then literature on sarcasm detection follows.

The purpose of Hadoop is used for storing and analyzing the huge set of data in a distributed computing environment. V. N. Khuv et al.[15] proposed distributed system for Sentiment Analysis of large scale data by using a MapReduce framework and a distributed database model. The system involved two components: lexicon builder [5] and sentiment classifier. They used the sentiment lexicon together with machine learning technique in order to solve the misclassification error of lexicon based classifier. The system was implemented by using existing twitter datasets. The experiment results showed that their lexicon and learning based classifier could obtain higher accuracy than the lexicon-based classifier which only relies on searching for sentiment words/phrases. For scalability, the evaluation results showed that the running time of the proposed system with different volumes of data decreases when adding more machines in the cluster. The authors [2] performed sentiment mining using a Naive Bayes classifier (NBC) to evaluate the scalability of NBC in large data sets. To achieve fine-grain control of the analysis procedure, they implemented the NBC on top of Hadoop framework with additional four modules: the work flow controller (WFC), the data parser, the user terminal and the result collector. The two datasets which had already been labeled the class were used for their experiments. They have verified that NBC was able to scale up to analyze the millions of movie reviews with increasing throughput.

Researchers had made a few experiments on sarcasm detection. S. K. Bharti et al.[10] 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 [12], 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. 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.

In this work, Sentiment Analysis is implemented on Big Data Analytics Platform to 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). These systems can facilities sarcasm and sentiment classification by conducting Multi-tier Sentiment Analysis with sarcasm detection scheme. To acquire effective training data for scalable learning based classification, RSSD is used for class labeling. Negation handling is examined to improve the performance of RSSD. To improve the classification accuracy, learning based classification approach with Multi-tier architecture is conducted. Mahout Naïve Bayes classifier is applied to provide scalable

learning based sentiment classification. A large volume of Twitter stream data is used for execution and the data are collected by Apache Flume.

3. Implementing Big Data Analytics Platform

In MSASDH, Big Data Analytics Platform is implemented to scale up the traditional analytics platform for analyzing large-scale tweets by using Apache Flume [11], HDFS, MapReduce [3] and Mahout Machine learning library [1].

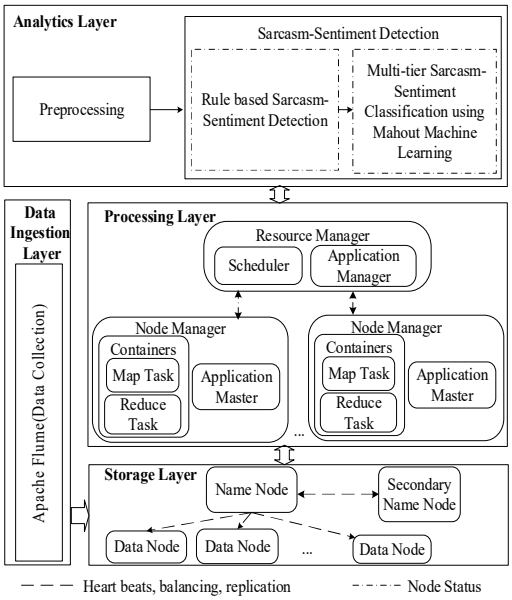

Figure 1. High Level Architecture of MSASDH

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 Figure1. It consists of four layers and the function of MSASDH on each layer is described as follow:

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.

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.

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.

Analytics Layer - preprocessing, Rule based Sarcasm-Sentiment Detection and Multi-tier sarcasm-sentiment classification are implemented in this layer. All of the processes from Analytics Layer executed in distributed manner by using HDFS and MapReduce. The Multi-tier sarcasm-sentiment classification is implemented by using Mahout Machine learning library.

4. Multi-tier Sentiment Analysis with Sarcasm Detection

MSASDH consists of three major processes: data collection, preprocessing and sarcasm-sentiment detection. Figure 2 illustrates the process flow of MSASDH.

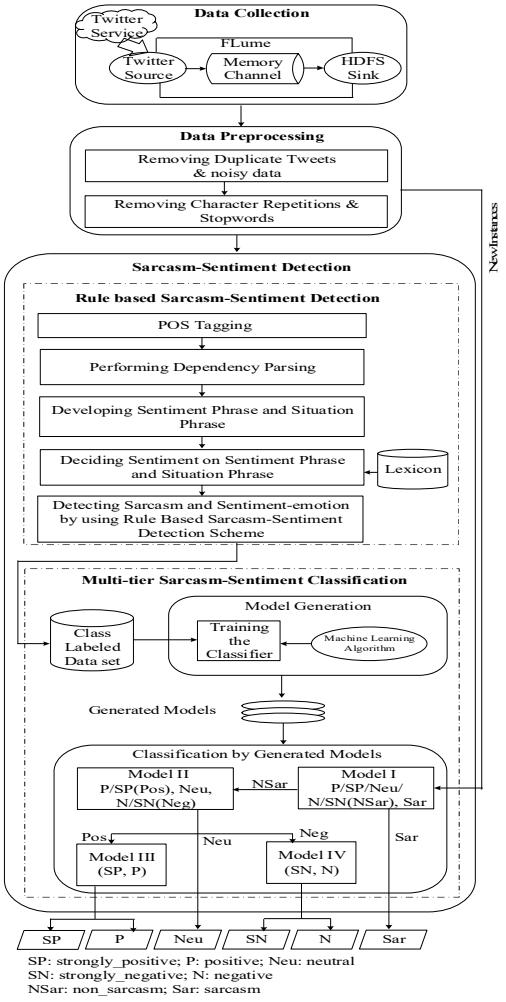

Figure 2. Process flow Diagram of MSASDH

4.1. Data Collection

In this work, Apache Flume collects Twitter stream data by Twitter Agent and the data is filtered by keywords. 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 a configured location in HDFS. In the HDFS Sink configuration, defines the size of the files with the roll Count Parameter.

4.2. Data Preprocessing

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

4.3. Sarcasm-Sentiment Detection

To detect sarcasm and sentiment, there are two major processes: Rule based Sarcasm-Sentiment detection and Multi-tier sarcasm-sentiment classification.

4.3.1. Rule based Sarcasm-Sentiment Detection

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

4.3.1.1. POS Tagging

GATE Twitter POS tagger [4] 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 [7] tag set.

4.3.1.2. 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. In this work, Opennlp [14] 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 3.

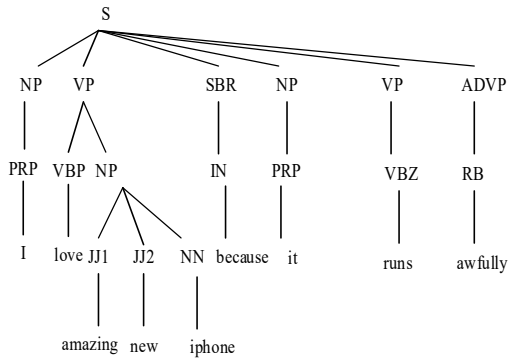

Figure 3. Sample Parse Tree

4.3.1.3. 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. Table 1 describes the sample result phrases of developing SEP & SIP for the example text in section (4.3.1.2).

Table 1. Sample Result Phrases of Developing SEP & SIP Procedure

SEP	SIP
I love, amazing	run awfully

4.3.1.4. Deciding Sentiment-Emotion on Sentiment Phrase and Situation Phrase

SentiStrength [8], 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 text, SentiStrength outputs a positive sentiment score 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 "SPSEP". If the sentiment score is greater than 0 and less than 2, the phrase is added to "PSEP". If the score is equal with or less than -2, the phrase is added to "SNSEP". If the score is less than 0 and greater than -2, the phrase is added to "NSEP". Otherwise, the phrase is added to NeuSEP. As the same way, the sentiment-emotion of SIP is decided. The procedure of deciding sentiment-emotion for developed phrases is illustrated in Reducer function of Figure 4. The output of this procedure is emotional SEP & SIP (PSEP, SPSEP, PSIP, etc.). Table 2 presents the notations for emotional SEP & SIP. Sample result phrases of deciding PSEP & NSIP of the example text from section (4.3.1.2) are shown in Table 3.

Table 2: Notation of Emotional SEP & SIP

Notation	Description
PSEP	positive sentiment phrases
SPSEP	Strongly_positive sentiment phrases
PSIP	positive situation phrases
SPSIP	strongly_positive situation phrases
NSEP	negative sentiment phrases
SNSEP	strongly_negative sentiment phrases
NSIP	negative situation phrases
SNSIP	strongly_negative situation phrases
NeuSEP	neutral sentiment phrases
NeuSIP	neutral situation phrases

```
Procedure : Deciding_Sentiment_emotion_Job
1.  Input : preprocessed tweets
2.  Output :  Emotional SEP & SIP
3.  Function Mapper (k1, v1)
4.  while(value € values)
5.      POS Tagging
6.      Dependency Passing
7.      foundNG=false // NG: Negation
8.      if(value € NGW) // NGW : negation words
9.          foundNG = true
10.         if ( foundNG == true)
11.             Append NGP to SEP // NGP € (NGW+VP || NGW + ADJ)
12.         if (POStagger == ADJP || NP || NP + VP)
13.             Append  POStagger to SEP
14.         if ( POStagger == VP ||  VP + ADVP || VP + ADVP + ADJP || VP + ADJP + NP || VP + NP || ADVP + VP || ADVP + ADJP + NP || ADJP + VP)
15.         Append  POStagger to SIP
16.         Emit SEP &  SIP
17. Function Reducer (k2, v2) // v2 € SEP & SIP
18. while (value € values)
19.     while (value € SEP)
20.         if  (totalscore>=2)  //totalscore:  the total score of each phrase
21.             Append value to SPSEP
22.         else if (totalscore > 0 && totalscore <2)
23.             Append value to PSEP
24.         else if(totalscore < 0 && totalscore >-2)
25.             Append value to NSEP
26.         else if(totalscore <= -2)
27.             Append  value to SNSEP
28.         else if(totalscore ==  0)
29.             Append value to NeuSEP
30.   while(value € SIP)
31.         if(totalscore>=2)
32.             Append value to SPSIP
33.         else if(totalscore > 0 && totalscore <2)
34.             Append value to PSIP
35.         else if(totalscore < 0 && totalscore >-2)
36.             Append value to NSIP
37.         else if(totalscore <= -2)
38.             Append  value to SNSIP
39.         else if(totalscore ==  0)
40.             Append value to NeuSIP
41.   emit(tweets, output)
```

Figure 4. Procedure of Deciding_Sentiment_emotion for SEP & SIP

Table 3. Sample Result Phrases of Deciding Emotional SEP & SIP Procedure

PSEP	NSIP
I love, amazing	run awfully

4.3.1.5. Detecting sarcasm-sentiment 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.

		Emotional SIP				
		P-SIP	SP-SIP	N-SIP	SN-SIP	Neu-SIP
Emo-tional SEP	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

Figure 5. Rule based Sarcasm-Sentiment Detection Scheme

Figure 5 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 this sarcasm-sentiment detection scheme, the testing tweets is classified whether sarcasm or sentiment (P, SP, N, SN, Neu).

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 and the generated models are used to classify the new data (unlabeled data).

4.3.2.1 Model Generation

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”.

4.3.2.2. Classification by Generated Models

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. The 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. Laplace smoothing is performed with value of α set to 1.

5. Experiments and Results

Experiment parameters for implementing the MSASDH, the dataset and explanations about evaluation results are presented in this section.

5.1. Experiment Environment

In this experiment, the cluster is composed of four computing nodes (VMs) with one name node and three data nodes. Each node is developed on each machine. The specifications of devices and necessary software component of MSASDH are presented in Table 4.

Table 4. Experiment Parameters

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

5.2. Datasets

In order to test the functionality of the MSASDH, tweets related with IPHONE and SAMSUNG 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 of Model I .

5.3. Evaluation Results

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 5. Performance of RSSD Compared To the Baseline Ones (IPHONE)

	Accuracy (%)	Precision (%)	Recall (%)
PBLGA [10]	72	76	69
S.Suzuki et al. [12]	76	81	70
RSSD	**85**	**90**	**78**

Table 5 shows the comparative results (IPHONE) of RSSD and the baseline ones. The result of applying RSSD to the dataset (IPHONE), RSSD achieved 85%, 90% and 78% accuracy, precision, recall respectively.

Table 6. Performance of RSSD Compared To the Baseline Ones (SAMSUNG)

	Accuracy (%)	Precision (%)	Recall (%)
PBLGA [10]	71	75	64
S.Suzuki et al. [12]	74	79	69
RSSD	**82**	**87**	**77**

Table 6 shows the comparative results (SAMSUNG) of RSSD and the baseline ones. The result of applying RSSD to the dataset (SAMSUNG), RSSD achieved 82%, 87% and 77% accuracy, precision, recall respectively. According to the results of Table 5 and Table 6, 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.

Table 7. Comparative Results of MSASDH and MSA (IPHONE)

	Classified as	**Accuracy (%)**	**Overall Accuracy (%)**
MSA	P	82	82
	SP	76	
	N	88	
	SN	80	
	Neu	84	
MSASDH	P	90	87
	SP	82	
	N	95	
	SN	83	
	Neu	85	

Table 7 presents the classification accuracy (IPHONE) of MSASDH and MSA. The overall accuracy of Multi-tier Sentiment Analysis with sarcasm detection (MSASDH) is higher than without sarcasm detection (MSA) by 5%.

Table 8. Comparative Results of MSASDH and MSA (SAMSUNG)

	Classified as	**Accuracy (%)**	**Overall Accuracy (%)**
MSA	P	**80**	80
	SP	78	
	N	**79**	
	SN	76	
	Neu	85	
MSASDH	P	**86**	83
	SP	80	
	N	**85**	
	SN	79	
	Neu	85	

Table 8 shows the classification accuracy (SAMSUNG) of MSASDH and MSA. The overall accuracy of Multi-tier Sentiment Analysis with sarcasm detection (MSASDH) is higher than without sarcasm detection (MSA) by 3%. According to the results of Table 7 and 8, there is an enhancement after considering the sarcasm. Because most of the sarcastic tweets are negative tweets that have been misclassified as positive. In other words, many of the tweets, previously misclassified as positive are now well classified as negative.

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.

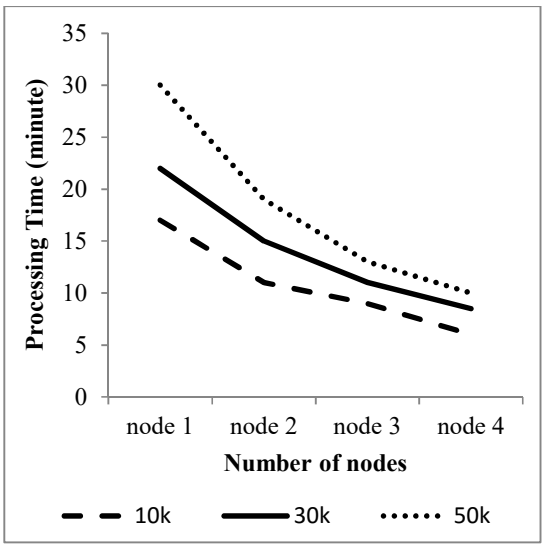

Figure 6. Processing Time of MSASDH

Figure 6 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 MSASDH 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. For 30k tweets, the processing time is decreased by 32% when increasing from single cluster node to 2 cluster nodes, and 50% for single cluster node to 3 cluster nodes, 65% for single cluster node to 4 cluster nodes. For 50k tweets, the processing time is decreased by 37% 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.

6. Conclusion and Future Work

In this paper, MSASDH is proposed for extracting valuable information from large-scale social big data. To achieve high-level performance of sentiment classification, this system is implemented by conducting 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 Sentiment Analysis with sarcasm detection scheme is higher than without sarcasm detection. Therefore, the results show that MSASDH can enhance sentiment analysis 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 this work, the proposed MSASDH detect only the types of sarcasm: contrast between positive sentiment and negative situation, and contrast between negative sentiment and positive situation. In future works, the MSASDH will need to be improved for solving other types of sarcasm and it will be implemented on Spark to facilities real time analysis of large-scale social data.

References

[1] A. Oliver, " Machine-learning-with-mahout," [online] Availabe: http://www.infoworld.com/article/2608418/application-development/enjoy-machine learning- ith-mahout-on hadoop.html [Accessed 3-December-2016]

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

[3] "Hadoop Yarn," [online] Available : https://hadoop.apache.org/docs/r2.7.1/hadoo-yarn/hadoop-yarn-site/YARN.html[Accessed 20-August-2016]

[4] 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.

[5] L. Velikovich, S. Blair-Goldensohn K. Hannan, R. McDonald, "The viability of web-derived polarity lexicons," The 2010 Annual Conference of the North American, pp. 777–785, June 2010.

[6] M. Skuza, A. Romanowski, " Sentiment Analysis of Twitter Data within Big Data Distributed Environment for Stock Prediction," Journal of Computer Science and Information Systems, Sept. 2015.

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

[8] M. Thelwall, K. Buckley, G. Paltoglou, "Sentiment Strength Detection for the Social Web," Journal of the American Society for Information Science and Technology Volume 63 Issue 1, pp. 163-173, January 2012.

[9] "Number of monthly active Twitter users worldwide from 1st quarter 2010 to 3rd quarter 2017 [online] Available: https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/[Accessed 4-December-2017]

[10] 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, http://dx.doi.org/10.1016/j.dcan.2016.06.002.

[11] 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/, June 30, 2015, [Accessed 20-Oct-2016]

[12] S. Suzuki, R. Orihara, Y. Sei, Y. Tahara, A. Ohsuga, “Sarcasm Detection Method to Improve Review Analysis,” 9th International Conference on Agents and Artificial Intelligence, pp. 519-526, 2017.

[13] T. P. Pilichowski, ,M. M. KrólC. M. Olszak, morf saedI weN :TCI ssenisuB ni secnavdA“ ,Ongoing Research”, 2016pp. 107.

[14] “The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text,” [online] Available: https://opennlp.apache.org/ [Accessed 26-June-2017]

[15] V. N. Khuc, C. Shivade, R. Ramnath, J. Ramanathan, “ Towards Building Large-Scale Distributed Systems for Twitter Sentiment Analysis,” 12 Proceedings of the 27th Annual ACM Symposium on Applied Computing, pp. 459-464, March, 2012.