International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 17 (2017) pp. 6564-6569

An Efficient Informal Data Processing Method by Removing Duplicated Data

Jaejeong Lee[1], Hyeongrak Park and Byoungchul Ahn*

Dept. of Computer Engineering, Yeungnam University, Gyeongsan, Korea.

**Corresponding author*

[1]*Orcid ID: 0000-0001-6637-3502*

Abstract

Useful information is extracted from many social networking services(SNS). The data uploaded by many users include the preferences or comments for the specific topics, which can be used for sentiment analysis. This sentiment information might be various social information, personal services and so on. However, SNS data are included a lot of duplicated data and spam data, which make slow down the processing time and the accuracy of the sentiment analysis. This paper presents an effective informal big data processing method by filtering out duplicated data or spam data. Hadoop Distributed File System(HDFS) and MapReduce method are used to extract sentiment information through machine learning. Experiment results increase not only the processing performance but also but also the accuracy of sentiment analysis. When duplication and spam are filtered out, the upload time is reduced about 4 seconds for 133,500 data. The analysis time after duplication and spam data are removed, is reduced by 36 percent and 41.26 percent respectively for 133,500 data. The incorrect spam detection is 20.53 percent for 35,320 data.

Keywords: Big data; SNS; Sentiment analysis; Spam filtering; Informal Data

INTRODUCTION

As the social networking service (SNS) becomes popular, it connects people with the same interest such as hobbies, sports, pets and so on. Typically short sentences are uploaded to social networks such as Twitters, Facebook and so on. The uploaded postings are highly valuable since there are many comparatively subjective opinions. After implementing the analysis system, it can be quickly estimated emotions of many users for uploaded sentences. It is easy to get new information and can be widely used to identify public opinions and preferences for specific topics. To investigate the preferences for specific topics, it is required to extract the polarity of opinions after collecting the data in SNS[1][2].

When a lot of data are generated, the duplicated data and spam data are also growing. Spammers create a large amount of spam data using automation tools and upload posting manually to avoid filtering. These generated data affect the speed of analysis or negative results and so on. Therefore, it is necessary a method to extract the information quickly and accurately by removing spam data on SNS. Data uploading to SNS follows the format of unconstrained informal data because there is no particular constraint for writing. Typically SNS data types might be formal data, semi-structured data or informal data. Informal data means the data such as video, image, document record file and so on. Many studies have been conducted to analyze informal data such as natural processing, morpheme analysis, Support Vector Machine (SVM) and so on [3][4][5][6][7][8].

This paper presents improved an informal data processing method by filtering out spam data. In Section 2, related works are discussed. In Section 3, the analysis model and the configuration for the HDFS and MapReduce are discussed. The performance of the proposed method is discussed in Section 4. Conclusion is described in Section 5.

RELATED WORK

Pak and Paroubek proposed a method for sentiment analysis which was processed by a sentiment classifier to distinguish positive, neutral and negative sentiments for messages[13]. They proposed an efficient analysis method, but did not consider for the collection and storage of data for analysis. Back et el. configured a parallel HDFS and MapReduce method to analyze informal sentiment data[14]. They analyzed sentimental data using their method, but did not consider unnecessary data removal for their analysis.

There are no research studies that remove unnecessary data efficiently in order to analyze subjective opinions efficiently. But most studies are concentrated to detect or analyze the spam generated from e-mail, SMS, web and so on. After spams are classified from its contents, e-mail address, phone number and user identification, they stored on blacklists to filter them out. Wang proposed a spam detection prototype system to identify suspicious in Twitter and compared several detection methods

based on its contents[16]. Chu et el. proposed a method to filter out spams from the contents of Tweet using Naive Bayesian classification algorithm[17]. Kim et al. proposed to categorize general account, tweet bot and cyborg by analyzing the features tweet interval time with entropy, spam content and so on[18]. They suggested how to distinguish malicious accounts but did not consider the handling of original texts. They also conducted to analyze the sentiment by referring to Twitter's hash tag.

It is necessary to filter out spams before polarity classification and to improve the accuracy of the sentiment analysis.

PROPOSED METHOD

A. Configuration of HDFS and MapReduce

In this paper, we configured the HDFS to collect and store Twitter data. We organized three Linux servers in parallel and configured the name node as redundancy server for the reliable server operation. The proposed HDFS consists of a primary server, a secondary server, and a data server to collect, store and process Twitter data as shown in Figure 1. The primary server is used as a main server for the distributed parallel processing and control of other servers. The secondary server is responsible for backing up the main name node. All servers have a data node and process the data. MapReduce is a programming model to process big data sets with a parallel and distributed algorithm on a cluster. In this paper, it performs sentiment analysis as shown in Figure 1. The analysis sequence starts from extracting words from Tweet data and pass those data to the Mapper. These extracted data are tokenized, created vector values and weighted by TF-IDF(Term Frequency-Inverse Document Frequency) value by Mahout. These values are used for sentiment analysis. These classified data are passed to the Reducer, added polarity values and stored in the MongoDB to calculate statistics.

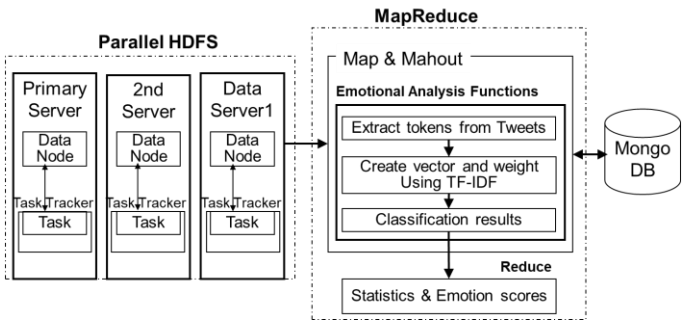

Figure 1: Proposed Configuration of HDFS

B. Informal Data Collection

For sentiment analysis, we collect the Twitter data for one week maximum using Twitter4j API. The token of Twitter is used to get authorization to collect data by the streaming API or the REST API. We used the REST API and the additional Bearer token to extend the number of query. As a result, the number of queries per 15 minutes was expanded from 180 to 450. The data collection process using Twitter4j API is as follows.

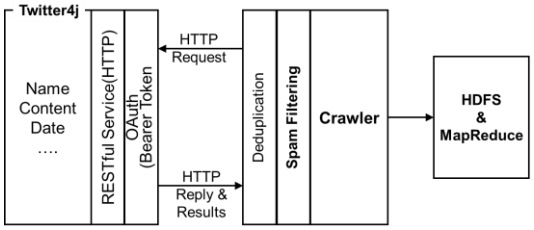

Figure 2: Data Collection Process using Twitter4j

C. Spams, Duplication data removal and sentimental classification

The data collected through Twitter4j API have a lot of data to process the sentiment analysis. By filtering out spams and duplicated data, we can improve the accuracy of sentiment analysis as well as the processing speed. Figure 3 shows pseudo code to index and remove spams. The spam training model is programmed to train the spam classifier by calculating the spam index. The index value is used to identify spam words or paragraphs for Twitter sentences. If the spam index value is three or higher, they are considered as spams. If not, they are considered as regular data. The spam filtering module stores common words trained with spams and common labels.

Figure 4 is pseudo code to remove duplicated words after checking duplication for each page. It removes blanks, non-important words or duplicated words. Therefore, non-duplicated words or sentences are used for sentiment analysis.

```
// Spam Removal
1.   Inputs:
2.     T - training model for Mahout
3.     K - keyword
4.     S = {S1, S2, …, Sk} – sentences // source data
5.     si - spam index, set to zero
6.   Outputs:
7.     R – result of common data
8.   Set up T
9.   foreach (Sn ∈ S)
10.    Calculate si(K, Sn)
11.    if si is more than criteria of spam index then skip
12.    else add Sn to R
13.  end
14.  return R
```

Figure 3: Pseudo-codes for spam removing module

```
// Duplication removal
1.    Inputs:
2.    S = {S1, S2, …, Sk} // sentences
3.    T - temporary space
4.    Outputs:
5.    U - unique data
6.    foreach (Sn ∈ S)
7.    T = Removal unnecessary characters(Sn)
8.    if U contains T then skip
9.      else add T to U
10.   end
11.   return U
```

Figure 4: Pseudo-codes for duplication removal module

D. Proposed sentiment analysis method

The polarity classification process using Mahout is divided into three phase, which are preparatory phase, training phase and classification phase as shown in Figure 5. Preparatory phase collects data randomly from Tweet to build a training set.

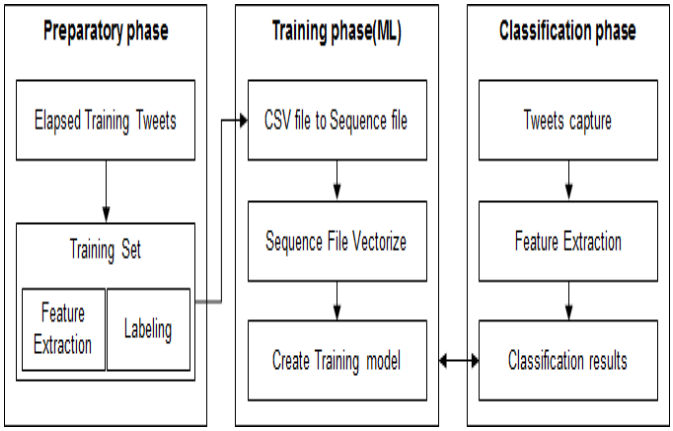

Figure 5: Polarity classification process using Mahout

Figure 6 is pseudo code to classify emotional polarity. It uses a sentiment training model to calculate sentiment polarity. If sentiment index is 0, it means neutral. If the index value is the upper 40% or higher, it means positive. If the index value is the lower 40% or lower, it means negative. Finally, the polarity is decided by calculating a sentence score through the training model.

```
// Sentimental classifier
1.    Inputs:
2.     T - training model for Mahout
3.     K - keyword
4.     S - source data
5.     S = {S1, S2, …, Sk} - sentences
6.     ps - positive minimum score, set to upper 40%
7.    ns - negative maximum score, set to lower 40%
8.     ss - sentence score, set to zero
9.    Outputs:
10.   R – result
11.   Set up T
12.   foreach (Sn ∈ S)
13.    Calculate ss(K, Sn)
14.   end
15.   if ss is zero then return neutral
16.   if ps contains ss then R = positive
17.   else if ns contains ss then R = negative
18.   else R = neutral
19.   return R
```

Figure 6: Pseudo code for polarity classification module

The collected Tweet data are transferred to the training phase after classifying polarity. These data are generated as a training model after they are converted to sequential files and vectorized. Finally, the polarity of classification result is derived from the classification phase.

EXPERIMENTAL EVALUATION

We have built a sentiment analysis system as shown in Table 1 to measure the performance of the proposed method. The operating system of all servers is Ubuntu 14.04.4 LTS 64x and consists of Hadoop parallel systems.

Table 1: System configuration for the sentiment analysis

Configuration	OS	CPU	Memory	HDD
Primary Server (main)	Ubuntu_14.04.4 LTS x64	Intel Core(TM)2 Duo E8400(2Core, 3.00GHz)	4GB	300G
Secondary Server (sub1)	Ubuntu_14.04.4 LTS x64	Intel Core(TM)2 Duo E8400(2Core, 3.00GHz)	4GB	300G
Data Server (sub2)	Ubuntu_14.04.4 LTS x64	Intel Core(TM)2 Duo E8300(2Core, 2.83GHz)	6GB	500G

A. Dataset collection

To measure the performance, we configured the dataset, which are consisted of a total of 21 according to three data collection methods. Configured dataset consists of a total of 989,148, as shown in Table 2. The training set consists of 15,364 and 11,591 respectively to divide spams and polarity. Since SNS changes in real time, the number of data varies slightly by the data collection time.

Table 2: Dataset for experiments

Dataset	Number of Data	Application method
Dataset 1	6,082	Default
	6,097	Duplication removal(DR)
	6,106	DR+Spam Filtering
Dataset 2	6,952	Default
	6,953	Duplication removal
	6,954	DR+Spam Filtering
Dataset 3	29,584	Default
	29,552	Duplication removal
	29,564	DR+Spam Filtering
Dataset 4	35,226	Default
	35,263	Duplication removal
	35,320	DR+Spam Filtering
Dataset 5	39,808	Default
	37,687	Duplication removal
	37,125	DR+Spam Filtering
Dataset 6	83,346	Default
	83,690	Duplication removal
	83,855	DR+Spam Filtering
Dataset 7	127,712	Default
	128,682	Duplication removal
	133,590	DR+Spam Filtering

B. Duplication and spam data removal

By the SNS characteristics posted freely, we have confirmed that the rates of duplication and spam are very high by advertising data, re-tweeting without additional personal opinion. Data duplication can be used differently depending on usage purposes. If the data are posted without additional comments, they are removed. The rates of common data, spam, and deduplication are shown in Table 3.

Table 3: Results of duplication and spam data removal

Dataset / Category	1	2	3	4	5	6	7
Default							
All data	6,082	6,952	29,584	35,226	39,808	83,346	127,712
Duplication removal							
All data	6,097	6,953	29,552	35,263	37,687	83,690	128,682
Duplicate data	2,937	5,400	16,055	14,743	26,389	33,036	63,214
Elapsed data	3,160	1,553	13,497	20,520	11,298	50,654	65,468
DR+Spam Filtering							
All data	6,106	6,954	29,564	35,320	37,125	83,855	133,590
Duplicate data	2,956	5,402	16,066	14,754	25,911	33,019	65,113
Elapsed data	3,150	1,552	13,498	20,566	11,214	50,836	68,477
Common data	3,060	1,300	10,904	15,121	10,714	49,802	66,606
Spam data	90	252	2,594	5,445	500	1,034	1,871

C. Data collection and uploading performance test

Figure 7 compares the collection time for data set shown in Table 2. Figure 8 compares the upload time for default datasets, datasets after removing duplication, and datasets filtering out spam s in HDFS. Although data collecting times of the three datasets are similar to each other, the uploaded times are different because duplication data and spam data are eliminated. The collection time and the HDFS upload time are increased as the amount of data are grows. The upload times are saved about 1 second to 4 seconds.

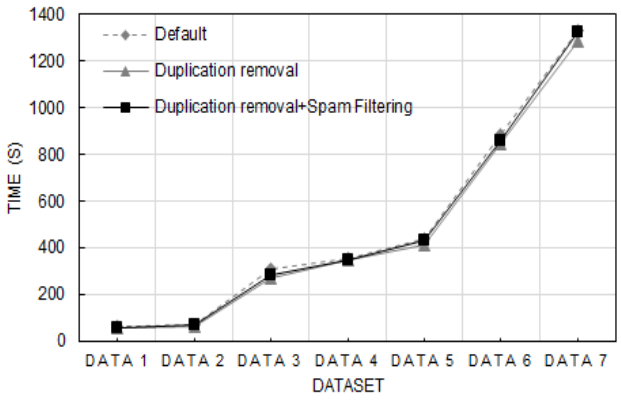

Figure 7: Data Collection Time

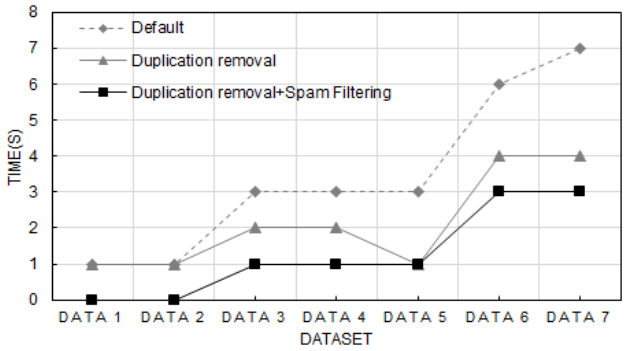

Figure 8: HDFS Upload Time

D. Analysis time

For Dataset 1, the analysis time is decreased from 12.11 seconds to 10.67 seconds. Also, when both duplication and spam data are removed, the analysis time is decreased to 10.29 seconds. For Dataset 6, the analysis time after duplication and spam data are removed, is reduced from 33.62 seconds to 24.72 seconds and 23.8 seconds, respectively. This means that the analysis time is reduced by 36 percent and 41.26 percent respectively. Therefore, it shows that the analysis time has increased stably according to the amount of data. The results of each data set are shown in Figure 9.

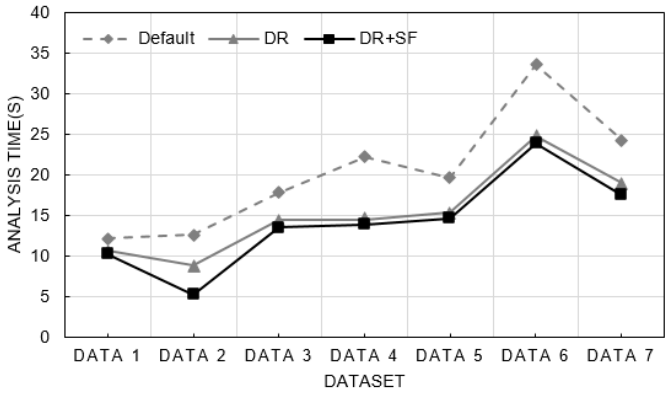

Figure 9: Analysis Time

E. Detection accuracy for spams and duplication data removal

The detection accuracy of duplication and spam data are evaluated. Dataset 3 and Dataset 4 are evaluated since those datasets contains a lot of duplication and spam data. In the case of duplication removal, it shows good performance since it is easy to figure out spaces or web link addresses. When users added meaningless characters such as "...", "!!!", they are not removed. In the case of spam filtering, when the spam classifier detects 2,594 spam data from 29,564 data for Dataset 3, it showed 481 wrong results. The incorrect spam detection is 18.45 percent. When the spam classifier detects 5,445 spam data from 35,320 data for Dataset 4, it showed 1,118 wrong results. The incorrect spam detection is 20.53 percent. The results for Dataset 3 and Dataset 4 are shown in Figure 10.

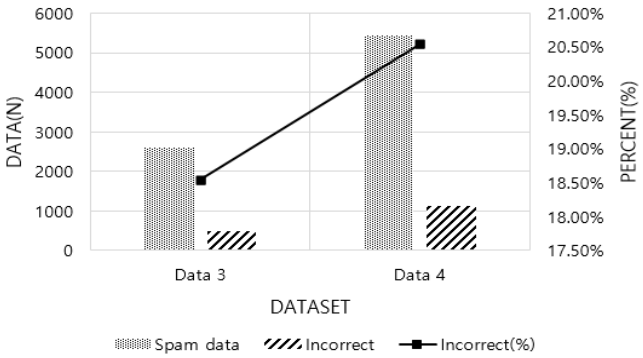

Figure 10: Practical Assessment of Spam Filtering

CONCLUSION

We have proposed an efficient method by removing lots of duplication and spam data before sentiment analysis for SNS. The proposed method is built on HDFS and MapReduce method to collect data from Twitter and eliminate duplication and spam data. When duplication and spam are removed, the upload time is reduced about 4 seconds for 133,500 data. The analysis time after duplication and spam data are filtered out, is reduced by 36 percent and 41.26 percent respectively for 133,500 data. The incorrect spam detection is 20.53 percent for 35,320 data.

It is necessary to improve the accuracy for spam filtering. To reduce the spam filter errors, we need to study more language specific studies for real life such as new words, slangs and so on. Also it is required to improve the spam training algorithms.

REFERENCES

[1] Jae-Young Chang, SinYoung Lee, JongBin Han. "Machine-Learned Classification Technique for Opinion Documents Retrieval in Social Network Services." Porceedings of KISS, p.245-247, June 2013..

[2] Joa-Sang Lim, Jin-Man Kim. "An Empirical Comparison of Machine Learning Models for Classifying Emotions in Korean Twitter." Journal of Korea Multimedia Society, 17.2 (2014.02): 232-239.

[3] Jinju Hong, Sehan Kim, Jeawon Park, Jaehyun Choi. "A Malicious Comments Detection Technique on the Internet using Sentiment Analysis and SVM." Journal of the Korea Institute of Information and Communication Engineering, 20.2 (2016.2): 260-267.

[4] Phil-Sik Jang. "Study on Principal Sentiment Analysis of Social Data." Journal of the Korea Society of Computer and Information , 19.12 (2014.12): 49-56.

[5] An, Jungkook, and Hee-Woong Kim. "Building a Korean Sentiment Lexicon Using Collective Intelligence." Journal of Intelligence and Information Systems 21.2 (2015): 49-67.

[6] Choi, Sukjae, and Ohbyung Kwon. "The Study of Developing Korean SentiWordNet for Big Data Analytics: Focusing on Anger Emotion." Journal of Society for e-Business Studies 19.4 (2014).

[7] Thoits, Peggy A. "The sociology of emotions." Annual review of sociology (1989): 317-342.

[8] Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." LREC. Vol. 10. 2010.

[9] Apache Hadoop, http://hadoop.apache.org (accessed Dec 02, 2016).

[10] D. Borthakur, Apache Software Foundation, "The Hadoop Distributed File System: Architecture and Design", 2007.

[11] C. M. Bishop, Pattern recognition and machine learning, Vol. 1, New York: springer, 2006.

[12] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large cluster," Communications of the ACM, Vol.51, Iss.1, pp.107-113, 2008.

[13] A. Pak and P. Paroubek, "Twitter as a Corpus for Sentiment Analysis and Opinion Mining," Proc. of the LREC'2010, 2010.

[14] Bong-Hyun Back, Ilkyu Ha, ByoungChul Ahn. "An Extraction Method of Sentiment Infromation from Unstructed Big Data on SNS." Journal of Korea Multimedia Society, 17.6 (2014.6): 671-680.

[15] Mahmoud, Tarek M., and Ahmed M. Mahfouz. "SMS spam filtering technique based on artificial immune system." IJCSI International Journal of Computer Science Issues 9.1 (2012): 589-597.

[16] Wang, Alex Hai. "Don't follow me: Spam detection in twitter." Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on. IEEE, 2010.

[17] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?," IEEE Transactions on Dependable and Secure Computing, vol. 9, no.6, pp. 811-824, Dec. 2012.

[18] J. Kim, S. Lee and H. Yong, "Automatic Classification Scheme of Opinions Written in Korea," Journal of KIISE: Database, Vol.38, No.6, pp.423-428. 2011.