Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2015, Article ID 417502, 11 pages
http://dx.doi.org/10.1155/2015/417502

Research Article

MapReduce Functions to Analyze Sentiment Information from Social Big Data

Ilkyu Ha,[1] Bonghyun Back,[2] and Byoungchul Ahn[2]

[1] *Department of Computer Engineering, Kyungil University, Gyeongsan 712-701, Republic of Korea*
[2] *Department of Computer Engineering, Yeungnam University, Gyeongsan 712-749, Republic of Korea*

Correspondence should be addressed to Byoungchul Ahn; b.ahn@yu.ac.kr

Received 20 October 2014; Revised 11 February 2015; Accepted 16 February 2015

Academic Editor: Praveen Rao

Opinion mining, which extracts meaningful opinion information from large amounts of social multimedia data, has recently arisen as a research area. In particular, opinion mining has been used to understand the true meaning and intent of social networking site users. It requires efficient techniques to collect a large amount of social multimedia data and extract meaningful information from them. Therefore, in this paper, we propose a method to extract sentiment information from various types of unstructured social media text data from social networks by using a parallel Hadoop Distributed File System (HDFS) to save social multimedia data and using MapReduce functions for sentiment analysis. The proposed method has stably performed data gathering and data loading and maintained stable load balancing of memory and CPU resources during data processing by the HDFS system. The proposed MapReduce functions have effectively performed sentiment analysis in the experiments. Finally, the sentiment analysis results of the proposed system are very close to those of manual processes.

1. Introduction

With the variety of social networking site (SNS) uses, opinion mining [1] has been recently introduced to extract useful information from SNS text and multimedia data and evaluate the real intention of users. In particular, opinion mining has been used to understand the true meaning and intent of the users on social networks. It is necessary to develop technologies that rapidly extract meaningful information from the large amounts of data generated by SNSs. A new real-time method is required to extract the ideas and thoughts from this mass of data. Such information can prove useful in many fields, such as economics, politics, media, culture. However, there are a lot of data on SNSs, and they have an atypical nature, so technologies that can efficiently process large amounts of unstructured data to obtain the necessary information are very important as fundamental technologies in social media data processing.

Social media is social interaction among people, in which they create, share, or exchange information, ideas, and pictures/videos/texts in virtual communities and networks [2]. Social media contains many forms of multimedia information. In particular, Twitter contains various forms of information, such as video links, image links, and text data. In this study, we focus on text data and we study methods for extraction of sentiment information from big-data text.

Since the data format is relatively easy and free, most SNS data are unstructured. Unstructured data may be defined as data that has not been standardized, because its structure and shape are so complex, unlike video image data and document data [1]. In order to extract meaningful information from large amounts of unstructured data on an SNS, a structuring process is needed for the unstructured data. Up to now, various technologies for processing unstructured data have been studied, focusing on morphological analysis.

Thus, research on text mining [3–5] has been developed, which extracts some information from semistructured or atypical text data, based on natural language processing (NLP) techniques. These methods use a statistical, periodic algorithm based on machine learning to extract meaningful information and to extract the information from the mass of text data. In addition, based on text mining technology, some research has also been carried out on opinion mining to determine the sentiment tendency of social network users, such as positive, negative, and neutral preferences [6, 7].

Recently, various open sources associated with the processing of big data have been provided. The Hadoop ecosystem [8] is a famous big data processing system that is most commonly used. More information on Hadoop will be offered in the next section. In this study, a parallel Hadoop Distributed File System (HDFS) [8] and MapReduce [9] functions based on Hadoop are proposed, which can stably collect and store a variety of data generated by social networks and analyze the sentiments of users on networks that have a lot of unstructured data.

2. Related Works

2.1. Previous Works for Sentiment Analysis. Sentiment analysis is a process to discover and extract subjective information from the original data with the aid of text analytics, computational linguistics, and natural language processing [6]. Up to now, a lot of researches have been developed to analyze big data for the sentiments of the users [10–12]. There are two general methods to decide the polarity of a sentence: one method that uses thesaurus and dictionaries and another method that uses a semantic dictionary built into semiautomatic or manual work. The second method is excellent in terms of utilizing capacity and accuracy but still has some problems in terms of cost and classification objectivity [13–15].

Pak and Paroubek [10] presented a method that could collect a corpus automatically for sentiment analysis and opinion mining purposes. Using the corpus, they built a sentiment classifier that was able to determine positive, negative, and neutral sentiments for a document. Though they proposed an efficient classifier for sentiment analysis, they did not consider the hardware aspect to collecting and saving the initial data. Mukherjee and Bhattacharyya [12] investigated the utility of linguistic features for detecting the sentiment of Twitter messages and showed that part-of-speech (POS) features might not be useful for sentiment analysis in the microblogging domain. They found using hashtags to collect training data proved useful. They investigated the role of the hashtag and parts of speech in sentiment analysis. However, they also did not consider the hardware aspects related to data collection and processing. Mukherjee and Bhattacharyya [12] presented a lightweight method for using discourse relations for polarity detection of tweets. The method incorporates discourse information in the bag of words model to improve accuracy. There are other studies on sentiment analysis. In particular, some studies deal with sentiment analysis on Twitter, such as the following. Go et al. [16] described a distant supervision-based approach for sentiment analysis classification. They use hashtags in tweets to create training data and a multiclass classifier to decide the polarity of the sentence. Barbosa and Feng [17] proposed a method for sentiment analysis in Twitter. The method used POS-tagged n-gram features and hashtags.

2.2. Open Systems for Processing Social Big Data. A variety of open source projects for big data processing are in progress with the Hadoop ecosystem [8]. The database management system that is widely used for big data processing with the Hadoop system is Not-Only Structured Query Language (NoSQL) [18]. It stores big data and retrieves some data using the consistency model, which is less restrictive than traditional relational databases. It is a very flexible database because it provides horizontal and vertical scalabilities, does not use the join operation between tables and table schema, and serves up fast response for reads and writes. Currently, a number of database models based on NoSQL are being introduced in academia and industry: BigTable from Google [19], Amazon DynamoDB from http://amazon.com/ [20], HBase from the Apache Software Foundation [21], Cassandra [8], and MongoDB [22] are typical.

In particular, MonDB has no schema and provides both regular expression search and searches for whether the sentence flexibly includes a particular value. Because NoSQL can be scaled with the horizontal extension method (horizontal scalability), a big data processing system can be extended by adding the existing systems in parallel at a low cost and not upgrading the system with an expensive central processing unit (CPU). Therefore, it is possible to process a large amount of data in parallel, compared to a conventional relational database management system, using the MapReduce techniques, filtering, data clustering operation, statistics, and data extraction.

2.3. Sentiment Analysis on Hadoop. Up to now, there have been many studies related to sentiment analysis on Hadoop. Khuc et al. [23] described a large-scale distributed system for real-time sentiment analysis on Hadoop. The system consists of two components: a lexicon builder and a sentiment classifier, which are capable of running on a large-scale distributed system using a MapReduce framework and a distributed database system. Bautin et al. [24] introduced the TextMap Access system, Lydia, which provides ready access to a wealth of interesting statistics on millions of people, places, and things across a number of web corpora using the Hadoop system. Lydia consists of five primary components: spidering, NLP markup, sentiment analysis, entity analysis and aggregation, and visualization.

In this study, a parallel HDFS that can stably extract and save the necessary data from a variety of unstructured SNS data is proposed. And sentiment analysis functions based on MapReduce [9] can extract sentiment information of the user from the data received from the proposed parallel HDFS.

3. Parallel HDFS and MapReduce Functions

3.1. Configuration of the Parallel HDFS. To handle a variety of unstructured SNS data efficiently, a big data processing system is proposed. The proposed system is comprised of a parallel HDFS and MapReduce, as shown in Figure 1. Parallel HDFS, which is based on the Hadoop ecosystem, is used to reliably collect and store data from a large amount of

Table 1: HDFS servers.

Server	Components	Functions
Master node	Main name node, data node, MapReduce, crawler	Main server for parallel distribution process, name node (controlling other servers), data node, data loading
Slave node 1 (backup node)	Secondary name node, data node	Backup server for main server, data node, data loading, data analyzing
Slave node 2 (data node)	Data node	Data node, data loading, data analyzing
Slave node 3 (data node)	Data node	Data node, data loading, data analyzing

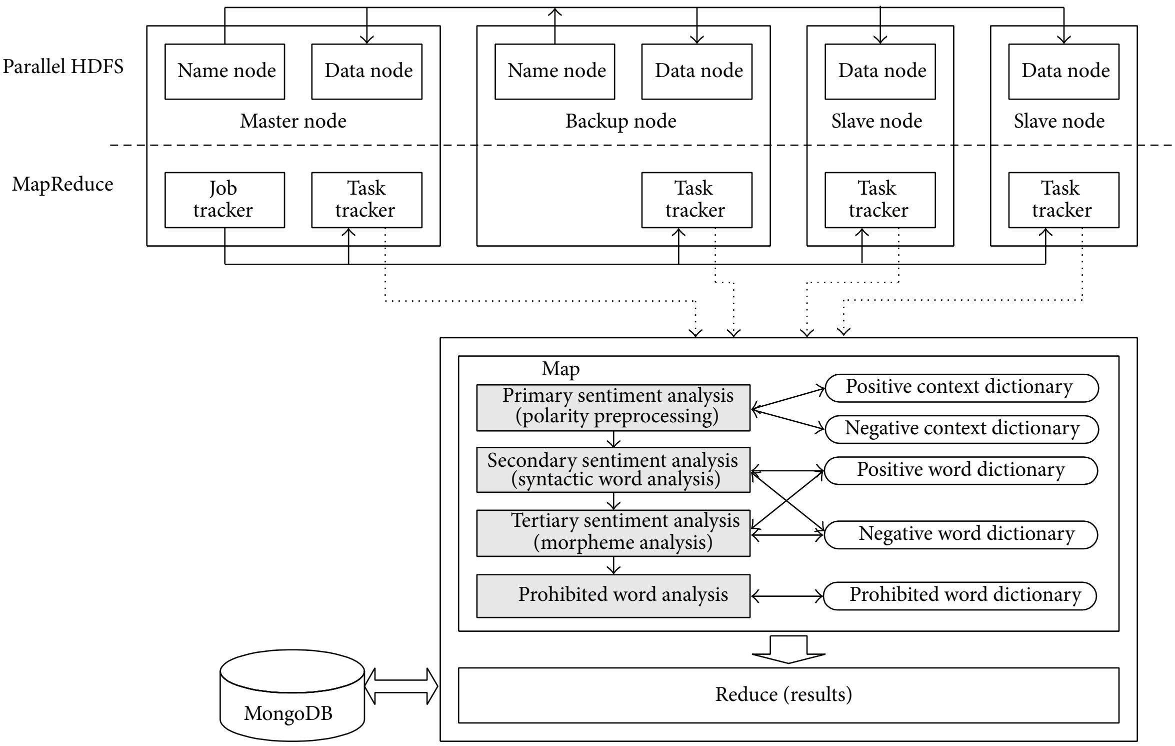

Figure 1: Data extraction method using parallel HDFS and MapReduce.

SNS data. MapReduce [9] is used to effectively analyze large amounts of unstructured data as to the sentiment of the user.

HDFS is a file processing system that has a distributed processing structure. It is configured in parallel, as shown in Figure 1, and uses four servers based on Linux, and each chunk node for storing data is set to 64 MB. It duplicates the name server using NFS for disaster recovery. Functions of the proposed servers are described in Table 1.

3.2. MapReduce Functions for Sentiment Analysis. MapReduce is a software framework developed by Google to support distributed computing, and it allows parallel programming using the function concept called Map. In this paper, it is classified into four special Map functions. They perform polarity preprocessing analysis, syntactic word analysis, morpheme analysis, and prohibitive word analysis, stage by stage. The lower part of Figure 1 shows the proposed MapReduce functions and the processes for sentiment analysis. Table 2 lists the four proposed functions and their operations. And Figure 2 shows the process of sentiment analysis using four sentiment analysis functions. Sentiment analysis is started from the loading of sentiment analysis dictionaries. And then, the SNS law data are input and loaded to the parallel HDFS. Later, the stepwise sentiment analysis is processed by four sentiment analysis functions. Sentences that do not determine the sensitivity in the previous step move on to the next step. The results of the final sentiment analysis are stored in the database; sentiment analysis is terminated.

First, sentiment analysis performs a polarity preprocessing function. This examines the context in each sentence to enhance the accuracy and subjects them to pattern matching with the negative context dictionary and the positive context dictionary. It counts the number of positive and negative contexts, and if the positives and the negatives are equal, the sentence is treated as positive but is transferred to

TABLE 2: The proposed sentiment analysis function.

Sentiment analysis function	Operations	Referenced dictionary
Polarity preprocessing function	Context analysis using sentence pattern matching	Positive/negative context dictionary
Syntactic word analysis function	Elimination of needless elements, calculation of the result count	Positive/negative word dictionary
Morpheme analysis function	Creation of tokens, calculation of the result count	Positive/negative word dictionary
Prohibited words analysis function	Calculation of the prohibited word score	Prohibited word dictionary

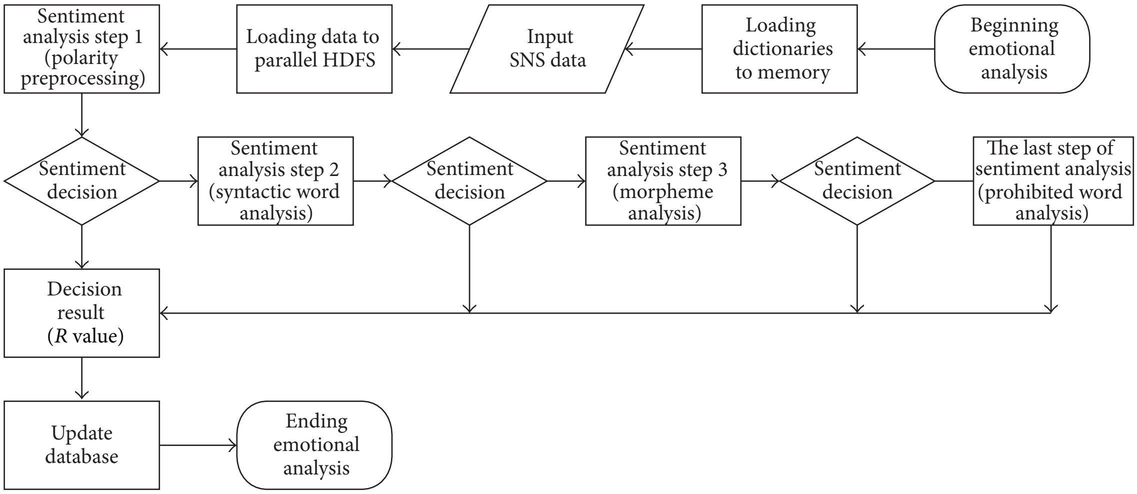
Beginning emotional analysis
Loading dictionaries to memory
Input SNS data
Loading data to parallel HDFS
Sentiment analysis step 1 (polarity preprocessing)
Sentiment decision
Sentiment analysis step 2 (syntactic word analysis)
Sentiment decision
Sentiment analysis step 3 (morpheme analysis)
Sentiment decision
The last step of sentiment analysis (prohibited word analysis)
Decision result (R value)
Update database
Ending emotional analysis

FIGURE 2: The process of emotion analysis using four emotion analysis functions.

morphological analysis if it is undetermined. The algorithm for polarity preprocessing is shown in Algorithm 1.

Second, sentiment analysis performs morphological analysis. This function removes any unnecessary components, such as special symbols, and it calculates the respective counters by comparing the sentence to positive and negative clause dictionaries. If the counter values of positives and negatives are equal, the sentence is treated as positive but is transferred to token analysis if it is undetermined.

Third is token analysis. After separating the source sentence into tokens by space, the function counts the positive and negative words by comparing the negative and positive dictionaries. If the counter value of the positives is equal to that of the negatives, the sentence is treated as positive but is transferred to prohibited word analysis if it is undetermined.

The prohibited word analysis function calculates a prohibition score based on a prohibited word dictionary. The algorithm for morphological analysis, token analysis, and prohibited word analysis is described in Algorithm 2.

3.3. Dictionaries for Sentiment Analysis. There are five proposed dictionaries used in the MapReduce functions. They are a positive context dictionary, a negative context dictionary, a positive word dictionary, a negative word dictionary, and a prohibited word dictionary. The prohibited word dictionary is composed of polarity and score. The roles of each dictionary are described in Table 3.

4. Sentiment Analysis Procedures

In this section, the procedure for sentiment analysis based on the proposed system and functions is explained. Sentiment analysis is processed with the following steps using parallel HDFS and MapReduce functions. First, social networking big data is gathered from some SNS services. Second, the necessary data is extracted from the gathered data. Third, the extracted data is processed to load into the HDFS. Fourth, the processed data is loaded into the parallel HDFS. Fifth, sentiment analysis is processed via the MapReduce functions using dictionaries for sentiment analysis. In Section 5, some experiments are processed for performance analysis of the proposed system and functions.

4.1. Data Gathering. The data collection method of the proposed system was processed through Twitter and Topsy for performance analysis. Topsy analyzes the activity of users in SNS services such as Google Plus and Twitter. Topsy provides data by analyzing about 500 million pieces of data per day. After the acquisition of the historical data, Twitter4j has been used to collect data for continuous incremental data. Twitter provides the most recent one week's worth of data and the key

Table 3: The proposed dictionaries for sentiment analysis.

Dictionary	Role	Application
Positive context dictionary	Set of positive context patterns, used to compute the number of positive contexts in a sentence	Polarity preprocessing
Negative context dictionary	Set of negative context patterns, used to compute the number of negative contexts in a sentence	Polarity preprocessing
Positive word dictionary	Set of positive word patterns, used to compute the number of positive words in a sentence	Syntactic word and morpheme analysis
Negative word dictionary	Set of negative word patterns, used to compute the number of negative words in a sentence	Synthetic word and morpheme analysis
Prohibited word dictionary	Set of prohibited words, used to compute the number of prohibited words in a sentence	Prohibited word analysis

```
//Polarity pre-processing
Inputs:
  $K$ - keyword
  $S$ - source data
  $S = \{S_1, S_2, \ldots, S_k\}$ - sentences
  $pc$ - positive count, set to zero
  $nc$ - negative count, set to zero
Outputs:
    $R$ - result
foreach ($S_n \in S$)
      Calculate $pc(K, S_n)$ //using positive/negative context dictionary
      Calculate $nc(K, S_n)$ //using positive/negative context dictionary
end
if $pc$ and $nc$ are zero then $R = 0$
$R = pc - nc$
If $R$ is zero then $R = pc$
return $R$
```

Algorithm 1: Polarity preprocessing.

Table 4: An example of data extraction.

Extraction data	Content
Create_at	Creation time of a tweet
Text	Contents of a tweet
Profile_image_url	Registered image in Twitter
Screen_name	Identification

that may be used to 450 queries for 15 minutes. In this study, a data collection module is to run every 4 hours using the crawler. Figure 3 shows the process of data gathering using Twitter4j.

4.2. Extraction and Processing of Necessary Data. The data collected from Twitter and Topsy contain a lot of unnecessary data. Therefore, only the necessary data needs to be extracted from the collected data. Table 4 shows an example of the type of data to be extracted from the raw data. And then, in the next step, the extracted data is processed to match the format of the extracted data from Twitter and Topsy.

4.3. Data Loading and Sentiment Analysis. The extracted data are stored in the proposed HDFS in parallel. Then, the stored data undergo sentiment analysis via the MapReduce functions. MapReduce functions consist of Reduce and Map. Figure 4 shows the general functionality of Map and Reduce. After the sentences to be analyzed are separated into data with their value and key, they are sorted based on the key value, then the sentences are sorted by their value. Finally the data are reduced on the key, and the final results are obtained.

However, in this study, the final scores are obtained by applying the four kinds of sentiment analysis functions proposed in Section 3.2 during the Map stage. Table 5 shows an example result from obtaining score values by applying the Map functions to the example sentences. It is possible to obtain a result that is configured to key and score, as seen in Table 6, as a result of the Map process. It is possible to obtain a Reduce processing result as seen in Table 7 by classifying the data on the basis of the key value again.

5. Experiment and Analysis

Several experiments are processed to check functions and performance of the proposed system. The following three

```
//Syntactic Word Analysis – if R is zero in previous stage
Inputs:
    S - source data
    $S = \{W_1, W_2, \ldots, W_k\}$ – syntactic words
    pc - positive count, set to zero
    nc - negative count, set to zero
Outputs:
    R - result
foreach ($W_n \in S$)
    Calculate $pc(W_n)$ //using positive/negative word dictionary
    Calculate $nc(W_n)$ //using positive/negative word dictionary
end
if pc and nc are zero then $R = 0$
$R = pc - nc$
If R is zero then $R = pc$
return R
//Morpheme Analysis – if R is zero in previous stage
Inputs:
    S - source data
    $S = \{M_1, M_2, \ldots, M_k\}$ – morphemes
    pc - positive count, set to zero
    nc - negative count, set to zero
Outputs:
    R - result
foreach ($M_n \in S$)
    Calculate $pc(M_n)$ //using positive/negative word dictionary
    Calculate $nc(M_n)$ //using positive/negative word dictionary
end
if pc and nc are zero then $R = 0$
$R = pc - nc$
If R is zero then $R = pc$
return R
//Prohibited Word Analysis – if R is zero in previous stage
Inputs:
    S - source data
    $S = \{M_1, M_2, \ldots, M_k\}$ – morphemes
    pc - positive count, set to zero
    nc - negative count, set to zero
Outputs:
    R - result
foreach ($M_n \in S$)
    Calculate $pc(M_n)$ //using prohibited word dictionary
    Calculate $nc(M_n)$ //using prohibited word dictionary
end
$R = pc - nc$
return R
```

Algorithm 2: Analysis of syntactic word, morpheme, and prohibited words.

Table 5: An example of sentiment analysis results.

Date	Keyword	Content	Score
20140204	YunaKim	YunaKim has no manner and prefers the money.	–1 (negative)
20140204	YunaKim	YunaKim is beautiful also, and it is okay to exercise well.	1 (positive)
20140204	YunaKim	Image was nice, and Yuna Kim was really beautiful.	1 (positive)
20140205	YunaKim	Yuna Kim, Samsung Galaxy S	0 (neutral)
20140205	YunaKim	Beautiful Yuna Kim Fighting!	1 (positive)

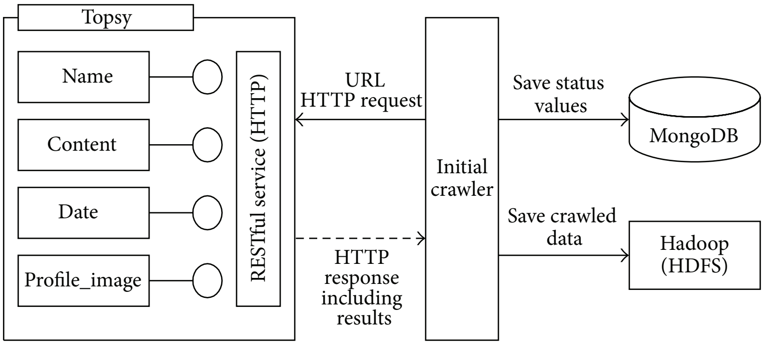

Figure 3: Data gathering using Topsy.

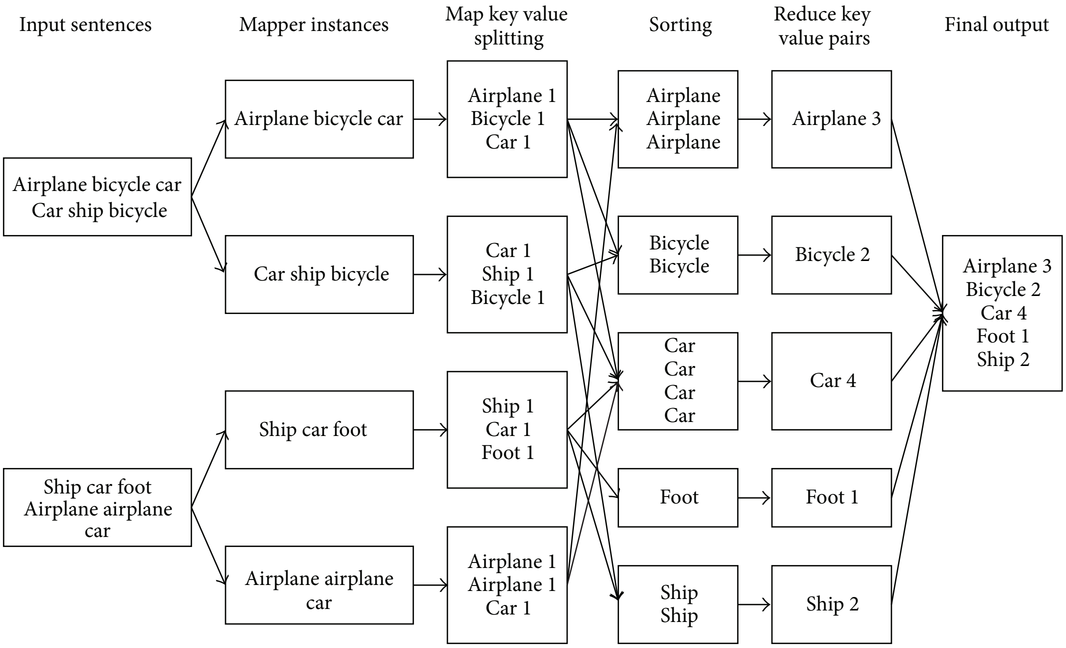

Figure 4: General Map and Reduce procedure.

Table 6: The results of Map processing.

Key	Score
20140204_YunaKim	−1
20140204_YunaKim	1
20140204_YunaKim	1
20140205_YunaKim	0
20140205_YunaKim	1

Table 7: The results of Reduce processing.

Key	Value
20140204_YunaKim	[−1, 1, 1]
20140205_YunaKim	[0, 1]

tests were carried out for performance analysis: performance test, time test, and accuracy test.

5.1. Experimental Environment. The experimental environment for performance analysis of the proposed system is described in Table 8. The proposed system consists of four Hadoop-based parallel servers using CentOS 6.3 x64 as an operating system.

The test for system load and acquisition time has been performed using the five Twitter data sets in Table 9. Each data set has been collected using the Topsy application programming interface (API).

5.2. Performance Test for Data Crawling and Loading. Among the performance tests of the proposed system for data crawling and loading into HDFS, first was an experiment for system performance according to the number of data items. Figure 5 shows a comparison of HDFS loading time and crawling time for each data set. For data set "A," the crawling time was 9 seconds and the HDFS loading time was 1 second. For data set "E," the crawling time was 753 seconds and the

Table 8: Experimental environment.

Components	OS	CPU	Memory	HDD
Primary server	CentOS_6.3 x64	Intel Xeon E5540 (8Core, 2.53 GHz)	8 GB	1 T
Secondary server	CentOS_6.3 x64	Intel Xeon E5540 (8Core, 2.53 GHz)	2 GB	1 T
Data server	CentOS_6.3 x64	Intel Xeon E3210 (4Core, 2.13 GHz)	2 GB	500 G
Data server	CentOS_6.3 x64	AMD Sempron 2800+ (1Core, 1.6 GHz)	1 GB	80 G

Table 9: Data sets for experiment and analysis.

Data set	Number of data items	Extraction period (day)	API
A	1,000	1	Topsy API
B	2,000	1	Topsy API
C	10,000	5	Topsy API
D	50,000	24	Topsy API
E	100,000	52	Topsy API

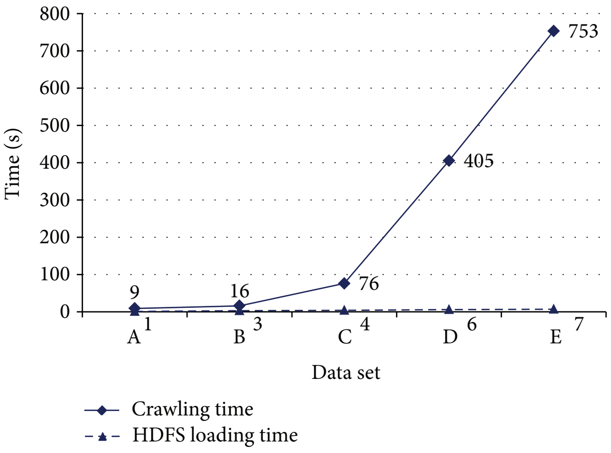

Figure 5: Crawling time and HDFS loading time.

HDFS loading time was 7 seconds, as shown in Figure 5. It is possible to see that the increases in HDFS loading time and crawl time are in proportion to the number of data items. Therefore, we can see that stable data collection and loading can be processed in a few seconds in the proposed system.

Figure 6 shows the memory load of each node in the HDFS when each data set is stacked and crawled. The memory usage of the SN1 to SN3 slave nodes used a maximum 2.5% and a minimum 0.10%, and the master node used a maximum 1.36% to a minimum 1.33%, as shown in Figure 6. The master node using the data distribution policy shows stable memory usage, regardless of the size of the data set. Slave node 1, used as a secondary server, also shows relatively stable memory usage. However, for slave node 2 and slave node 3, which are the data dedicated servers, memory usage increased in accordance with the size of the data set. Also, the loads for data crawling and loading were distributed in a balanced manner, and the memory resources were used efficiently according to the data distribution load policy on the master node.

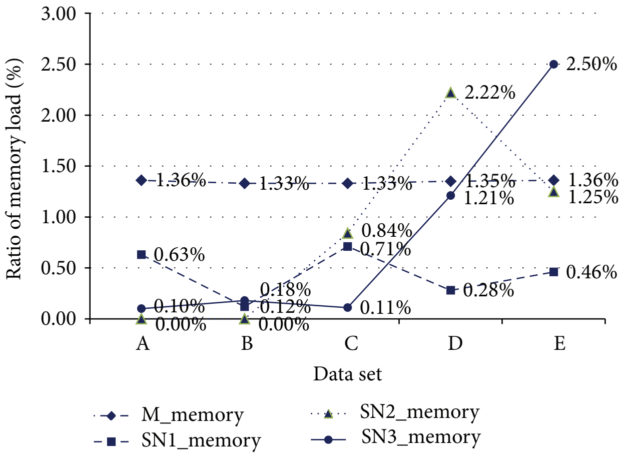

Figure 6: Memory usage for data crawling and HDFS loading.

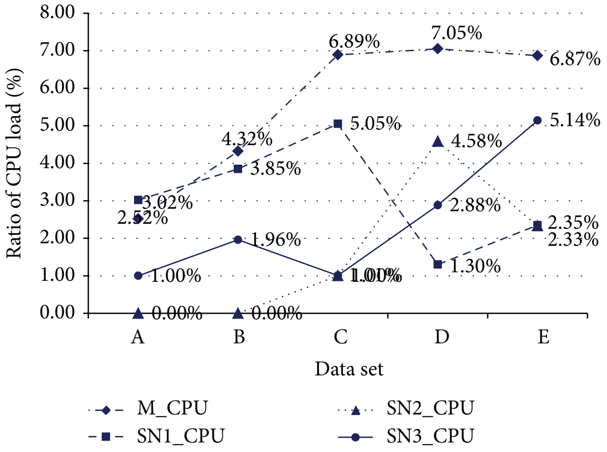

Figure 7: CPU load for data crawling and HDFS loading.

Figure 7 shows the CPU load of each node in the HDFS when each data set is stacked and crawled. For slave node 1 (SN1) and slave node 2 (SN2), their CPU loads covered a maximum 5.14% to a minimum 0.0%. However, for the master node, the CPU load shows a minimum 2.52% up to a maximum 7.05%.

As with the memory load and the CPU load, we can see that load balancing between the slave nodes is performed rather than focusing on a particular node in the course of automatic parallel processing of an HDFS load. The master node was found to provide a stable CPU operating environment during the collection of data.

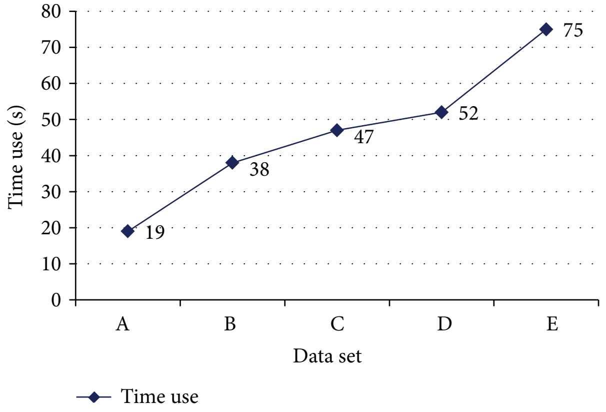

Figure 8: Time spent for MapReduce processing and sentiment analysis.

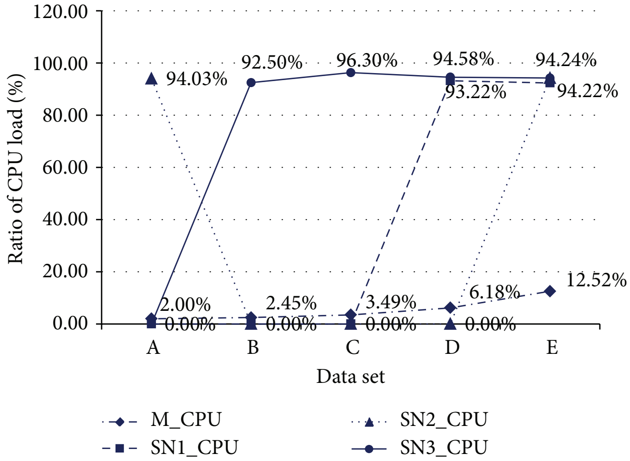

Figure 9: CPU load for MapReduce processing and sentiment analysis.

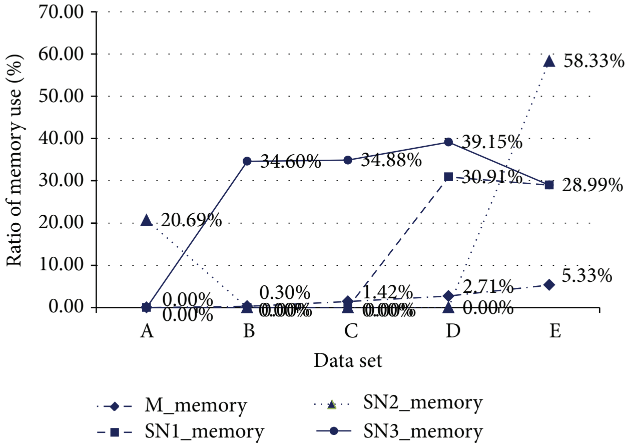

Figure 10: Memory load for MapReduce processing and sentiment analysis.

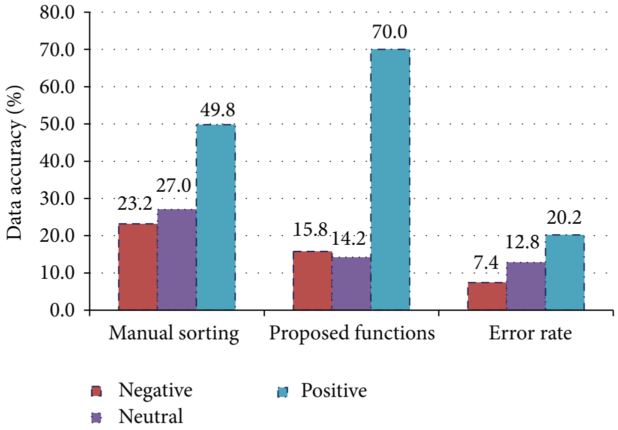

Figure 11: Comparing the results of the proposed functions with results of manual sorting.

5.3. Performance Test for MapReduce Processing and Sentiment Analysis. A performance test for MapReduce processing and sentiment analysis was conducted. Sentiment analysis time and system load were tested according to the number of data items. The experiment was executed for the degree of system load and the time required for sentiment analysis. Figure 8 shows the sentiment analysis time required for each data set. Figures 9 and 10 show CPU load and memory load, respectively, for each node during MapReduce processing and sentiment analysis. Sentiment analysis took from 26 seconds to 68 seconds, in accordance with the scale of the data sets. Analysis time increases linearly with the scale of the data set, as shown in Figure 8.

In Figure 9, the master node is not responsible for actual analysis processing but for managing the subslave nodes. Its CPU usage is low when the slave nodes use most of the CPU resources. When the number of items in the data set is less than 40,000, each slave node processes data in parallel. When the number of items in the data set is greater than 50,000, all slave nodes utilize high CPU resources, in accordance with the number of data items. Therefore, the proposed system performed stably as the number of data items increased. This is because the proposed system engages parallel mode if CPU load increases. In Figure 10, the memory usage of the master node is low, but the memory usage of slave nodes was partitioned mutually for parallel processing in each slave node. Therefore, the proposed system distributes workload equally among the slave nodes to ensure load balancing.

The algorithm of the proposed method shows $O(n)$ processing time. It provides a stable parallel analysis environment without operating on a single node only.

5.4. Accuracy Test. A test of the sentiment analysis has been conducted to measure accuracy. "Yuna Kim" (Korea's famous figure skater) was used to analyze sentiment. Figure 11 shows the comparison results of the proposed system against a manual process. In Figure 11, the error ratio for positive sentiment is relatively high, and the error rates for negative and neutral sentiments are relatively small. Therefore, the results of the sentiment analysis with the proposed system are very close to the results of sentiment analysis under a manual process.

5.5. Discussion. This experiment is based on the Korean language. The proposed HDFS in this study can be applied, regardless of language, for loading data in parallel. And the four kinds of MapReduce functions proposed in this study can also be applied to English, because they are based on a matching method basically using stored dictionaries. Furthermore, in morphological analysis, it is widely known that the accuracy in English is higher than in Korean. However, for the analysis of English, an English analyzer for morphological analysis and reserved dictionaries is required. In this study, we have implemented sensitivity analysis through morphological analysis, but we can consider a method of sensitivity analysis through machine learning with Apache Mahout in the future. In addition, in this study, we have focused on the text data of multimedia on social networks, and then extracted sentiment information. For future study, in order to more accurately determine sentiment, the use of a variety of multimedia information can be considered for sentiment analysis.

6. Conclusions

A multimedia data processing system and algorithms are proposed to analyze the sentiments of users from large amounts of unstructured text data generated by SNSs. The proposed method is composed of a parallel HDFS system based on the Hadoop ecosystem and on four MapReduce functions. In addition, it uses five types of data dictionary for sentiment analysis.

The proposed method stably processes data loading according to the increase in the number of data items. The system load is distributed to each node by parallel processing. When the proposed sentiment analysis functions have processed the data effectively, the system load is not concentrated on a single node but is evenly distributed among all nodes. The results of sentiment analysis with the proposed system are very close to the results of sentiment analysis with a manual process.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the 2014 Yeungnam University research grant.

References

[1] McKinsey, *Big Data: The Next Frontier for Innovation, Competition, and Productivity*, McKinsey & Company, 2011, http://www.mckinsey.com/.

[2] WIKIPEDIA, 2014, http://en.wikipedia.org/wiki/Social_media.

[3] A. Tan, "Text mining: the state of the art and the challenges," in *Proceedings of the PAKDD Workshop on Knowledge Disocovery from Advanced Databases*, pp. 65–70, 1999.

[4] Q. Mei and C. Xhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining," in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD '05)*, pp. 198–207, 2005.

[5] K. Park and K. Hwang, "A bio-text mining system based on natural language processing," *Journal of KISS: Computing Practices*, vol. 17, no. 4, pp. 205–213, 2011.

[6] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1–135, 2008.

[7] B. Kang, M. Song, and W. Jho, "A study on opinion mining of newspaper texts based on topic modeling," *Journal of the Korean Society for Library and Information Science*, vol. 47, no. 4, pp. 315–334, 2013.

[8] Hadoop, http://hadoop.apache.org/.

[9] J. Y. Chang, "Automatic retrieval of SNS opinion document using machine learning technique," *The Journal of The Institute of Internet, Broadcasting and Communication*, vol. 13, no. 5, article 27, 2013.

[10] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," *Proceedings of the 7th Conference on International Language Resources and Evaluation (LREC '10)*, 2010.

[11] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: the god the bad and the OMG!," in *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, pp. 538–541, 2011.

[12] S. Mukherjee and P. Bhattacharyya, "Sentiment analysis in twitter with lightweight discourse analysis," in *Proceedings of the 24th International Conference on Computational Linguistics (COLING '12)*, pp. 1847–1864, December 2012.

[13] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10760–10773, 2009.

[14] S. Gilbert and N. Lynch, "Brewer' s conjecture and the feasibility of consistent, available, partition-tolerant web services," *ACM SIGACT News*, vol. 33, no. 2, pp. 51–59, 2002.

[15] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[16] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," Project Report, Stanford University, 2009.

[17] L. Barbosa and J. Feng, "Robust sentiment detection on twitter from biased and noisy data," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling '10)*, pp. 36–44, August 2010.

[18] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NoSQL database," in *Proceedings of the 6th International Conference on Pervasive Computing and Applications (ICPCA '11)*, pp. 363–366, October 2011.

[19] F. Chang, J. Dean, S. Ghemawat et al., "Bigtable: a distributed storage system for structured data," *ACM Transactions on Computer Systems*, vol. 26, no. 2, article 4, 2008.

[20] S. Sivasubramanian, "Amazon dynamoDB: a seamlessly scalable non-relational database service," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '12)*, pp. 729–730, May 2012.

[21] L. George, *HBase: The Definitive Guide*, O'Reilly, 2011.

[22] K. Chodorow, *MongoDB: The Definitive Guide*, O'REILLY, 2nd edition, 2013.

[23] V. N. Khuc, C. Shivade, R. Ramnath, and J. Ramanathan, "Towards building large-scale distributed systems for Twitter sentiment analysis," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, pp. 459–464, March 2012.

[24] M. Bautin, C. B. Ward, A. Patil, and S. S. Skiena, "Access: news and blog analysis for the social sciences," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 1229–1232, April 2010.