International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 10 Issue 6 ǁ 2022 ǁ PP. 1589-1597

Highly Scalable MapReduce Application for Sentiment Analysis

Bonghyun Baek, Yunkyu Ryu, Huang J. Wangi
Department of Health and Medical Computing, Daegu Health University, China
Corresponding Author: H. J. Wangi

Abstract
Recently, opinion mining has been introduced to extract useful information from a large amount of SNS (Social Network Service) data and evaluate the user's true information. Opinion mining requires an efficient technique that collects and analyzes data from a large amount of data within a short period of time to extract information suitable for the purpose. To extract emotional information from various unstructured data generated from SNS, this paper proposes a Hadoop Distributed File System (HDFS)-based parallel and MapReduce-based sentiment analysis function. As a result of the experiment, it was confirmed that the proposed system and function process data collection and loading time faster, and stable load distribution for memory and resources.

Keywords: *Big Data, HDFS, MapReduce, Emotion Analysis (emotion analysis), Unstructed data analysis (unstructured data analysis)*

Date of Submission: 08-06-2022 Date of acceptance: 24-06-2022

I. INTRODUCTION

In a social atmosphere in which smartphone use is increasingly active, opinion mining, which extracts meaningful user information using a large amount of SNS (Social Network Service) data, is emerging. In particular, Opinion Mining is being used to understand the intention and truth of users on the platform. In such opinion mining, it is essential to quickly extract meaningful information from a large amount of data generated on the platform and to process it. In other words, there is a need for a technology that quickly analyzes the data of the business to extract meaningful information, and through this, the opinions and thoughts demanded by the public can be grasped in real time, and can be utilized in various fields that produce products and provide services. . In addition, it is necessary to more efficiently manage and visualize such refined, valid and diverse information through big data processing and analysis technology. Therefore, in this study, the Hadoop-based parallel HDFS (Hadoop Distributed File System) and MapReduce-based sentiment analysis that can analyze user emotions from various data generated on SNS, especially unstructured data function is proposed.

1.1 PREVIOUS WORKS

Since the method of interconnection between information delivery targets is easy and the data creation format is relatively free, most of the data generated on SNS is unstructured data. Unlike numeric data, unstructured data can be defined as unstructured data because of its complex form and structure, such as pictures, images, and documents [1]. In order to extract meaningful information from numerous unstructured data generated on SNS, processing of unstructured data is required first.

In the case of unstructured data analysis, various analysis methods [2-4] are being studied based on morphological analysis. However, there are factors that hinder data analysis, such as new buzzwords, unconfirmed words, symbolic words through symbolic characters, and pasting of words from various broadcasting media and young people. Accordingly, language analysis and sentiment analysis through a computer are becoming more difficult, and validation of these is becoming more difficult.

Therefore, research [6-7] on text mining [5], which extracts and processes information from unstructured or semi-structured text data based on natural language processing technology, is in progress. They use dictionary-based, machine learning-based statistical and regular algorithms to extract meaningful information from large amounts of text data or to refine related information. Also, research on Opinion Mining, which judges the preference of Positive, Negative, and Neutral in a text, is being conducted [8-9].

Currently, various open source projects for big data processing are under the name Hadoop ECO system [10]. The database used for big data processing uses NoSQL (Not-Only SQL) [11] for data storage and retrieval using a consistency model that is less restrictive than traditional relational databases. NoSQL uses a contextual database as well as a conventional relational database. In other words, it provides a form of horizontal scalability from vertical scale up that can be seen in existing relational databases, there is no join between table schema and tables, fast response time for read/write, and It provides extensibility. Currently, many studies on NOSQL databases are being conducted in industry and academia, representatively Google's BigTable [12], Amazon's Amazon DynamoDB [13], Apache HBase [14] of open source projects, Cassandra [10], MongoDB [15] is an example. In particular, MonDB used in this study is a CP-type database that satisfies Consistency and Partition tolerance when the database is classified according to the theory of Consistency, Availability, and Partition tolerance. JSON-type document data is stored in the method of), it is possible to expand the system at a low cost by adding the same equipment as the existing system in parallel, rather than upgrading expensive CPUs and memory. Therefore, compared to traditional RDBMS, large amounts of data can be processed in parallel, and data clustering operations, statistics, data extraction and filtering are possible using MapReduce techniques, etc.

Sentiment is “the mind or feeling that arises about a certain phenomenon or event [16]”. Emotional vocabulary mainly describes the meaning domain where inner or subjective emotions or psychology act rather than objective value evaluation. Sentiment analysis is a process of discovering and extracting subjective information from raw data using natural language processing, computational linguistics, and text analysis [16]. Studies [17-19] to analyze user emotions from big data are in progress. The work of analyzing and classifying the types of emotions can be divided into three stages. In the first step, sentences expressing subjective thoughts or feelings containing emotional information are extracted, and in the next step, the polarity (positive, negative) of the document or sentence is divided. The last step is intensity classification, which determines how much subjectivity a document or sentence has [20-21]. The polarity discrimination method using the English thesaurus and English-Korean dictionary and the method using the semi-automatic/manual semantic dictionary are used for the emotional analysis processing of Korean. Among them, the semi-automatic/manual semantic dictionary construction method is the best method in terms of accuracy and usability, but there are still problems in terms of cost and objectivity of semantic classification [22]. Therefore, in this study, a Hadoop system-based parallel Hadoop Distributed File system (HDFS) that can extract user's emotional information from various unstructured SNS data is proposed, and the user's emotional We propose a sentiment analysis function based on MapReduce [23] that can extract information.

II. EXTRACTION OF ATYPICAL EMOTIONAL INFORMATION

2.1 PARALLEL HDFS CONFIGURATION

Figure 1 shows the configuration of a parallel system for stably collecting unstructured data from a large amount of SNS data and extracting text-type data . The text-type data extracted from the system is used as an input to the MapReduce-based emotion analysis function, which is the next step for extracting emotion information.

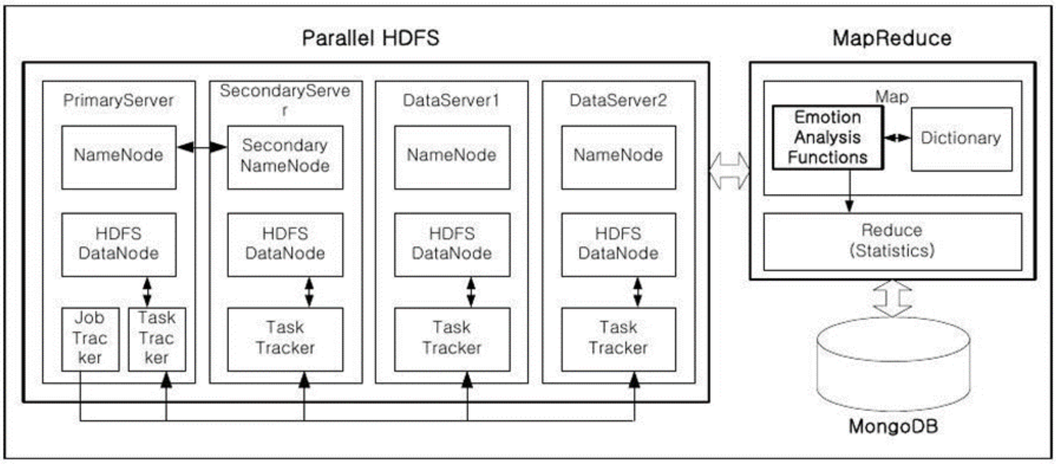


Figure 1: Parallel HDFS

HDFS is a file processing system with a distributed processing structure. As shown in Figure 1, the proposed HDFS is connected in parallel with 4 Linux-based servers, and each chunk of nodes to store data is composed of and the name server using is duplicated for failure recovery. The functions of the configured server are shown in Table 1.

Server	Components	Role
PrimaryServer (Master Node)	Namenode, DataNode MapReduce, Crawler	Main server for parallel distribution process Name node (controlling other servers) Data node, Data loading
SecondaryServer (Slave Node 1)	Secondary NameNode DataNode	Backup server of main server Data node, Data loading
DataServer1 (Slave Node 2)	DataNode	Data node, Data loading
DataServer2 (Slave Node 3)	DataNode	Data node, Data loading

Table 1: HDFS configuration

2.2 MAPREDUCE BASED SENTIMENT ANALYSIS

MapReduce is a software framework developed by Google for the purpose of supporting distributed computing and enables parallel programming using the concepts of map and reduce functions. This study proposes four types of MapReduce-based sentiment analysis functions as shown in [Table 2]. That is, it is a function of positive/negative context analysis, morpheme analysis, token analysis, and non-word analysis functions.

Emotion function	Role	Referenced dictionary
positive/negative context analysis function	context analysis using sentence pattern matching	positive/negative context dictionary
morphological analysis	elimination of needless elements, calculation of the result count	positive/negative word dictionary
creation of tokens calculation of the result count		
prohibited words analysis function	calculation of the prohibited word score	prohibited word dictionary

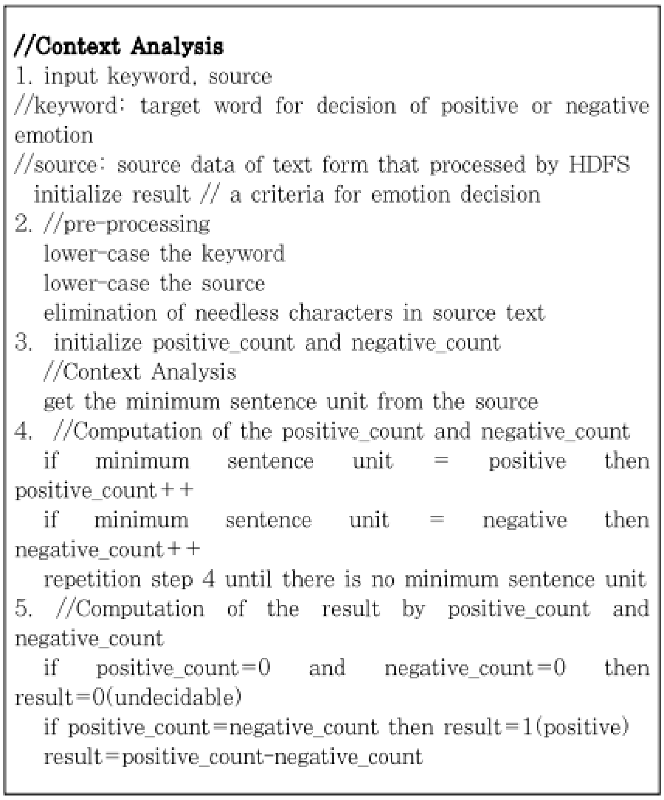
Table 2: Sentiment Analysis Functions Proposed

First, it is a positive/negative context analysis function. This function first checks the context in units of sentences to increase accuracy, performs pattern (regular expression) matching using positive and negative context dictionaries, and counts the original data (tweet text) as positive and negative, If the positive and negative counts are the same, it is treated as positive, and if it is impossible to judge, it is transferred to morphological analysis. Algorithm for context analysis is shown in [Figure 2]. Second, it is a morphological analysis function. In this function, Hannanum's Hangul morpheme analyzer is used to remove unnecessary elements for analysis, such as links and special symbols, and then compares positive word and negative word dictionaries to calculate each counter. Also, if the positive or negative counter values are the same, it is treated as positive, and if it is in an undetermined state, it is transferred to token analysis.

Third, it is a Totten analysis function. This function separates the token of the source (tweet text) with a space, analyzes the morpheme using the Hangul morpheme analyzer, and includes the result of positive word and negative word

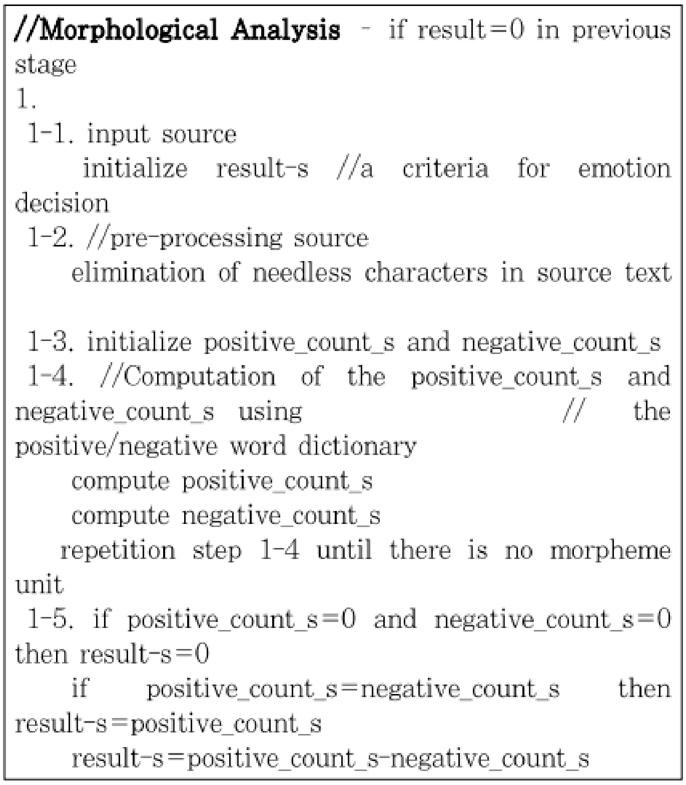
Each counter is calculated by comparing clause dictionaries, and if the positive or negative counter values are the same, it is treated as positive.

Fourth, it is a non-law analysis function. In this function, as the final analysis stage, if analysis is not performed in the upper process, the non-law score is calculated based on the non-law dictionary. Algorithms for morpheme analysis, token analysis, and non-word analysis are Figure 2, Figure 3 and Figure 4.



```
//Context Analysis
1. input keyword, source
//keyword: target word for decision of positive or negative emotion
//source: source data of text form that processed by HDFS
  initialize result // a criteria for emotion decision
2. //pre-processing
   lower-case the keyword
   lower-case the source
   elimination of needless characters in source text
3.  initialize positive_count and negative_count
   //Context Analysis
   get the minimum sentence unit from the source
4.  //Computation of the positive_count and negative_count
   if minimum sentence unit = positive then positive_count++
   if minimum sentence unit = negative then negative_count++
   repetition step 4 until there is no minimum sentence unit
5.  //Computation of the result by positive_count and negative_count
   if positive_count=0 and negative_count=0 then result=0(undecidable)
   if positive_count=negative_count then result=1(positive)
   result=positive_count-negative_count
```

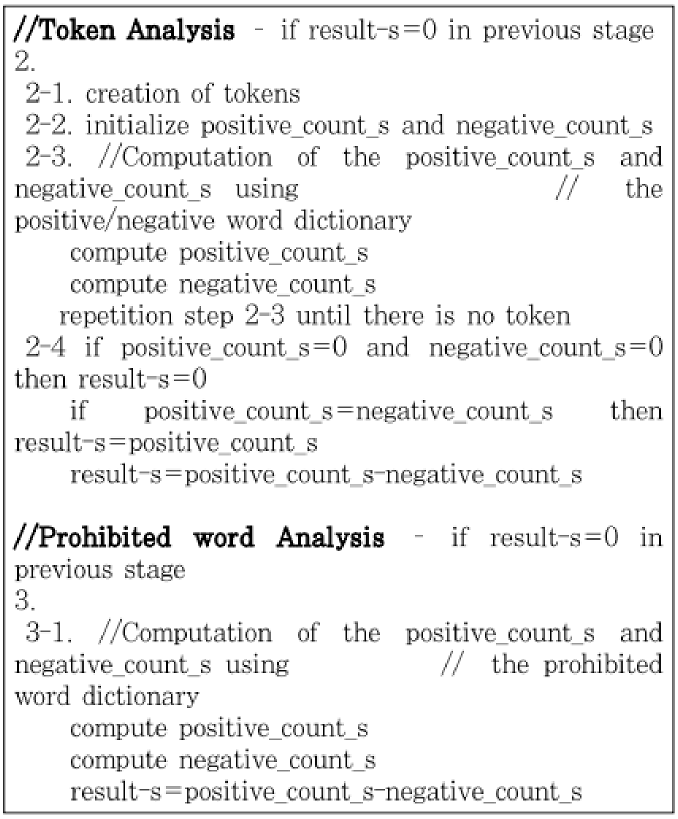
Figure 2: Context analysis function



```
//Morphological Analysis - if result=0 in previous stage
1.
 1-1. input source
      initialize result-s //a criteria for emotion decision
 1-2. //pre-processing source
     elimination of needless characters in source text

 1-3. initialize positive_count_s and negative_count_s
 1-4. //Computation of the positive_count_s and negative_count_s using // the positive/negative word dictionary
     compute positive_count_s
     compute negative_count_s
    repetition step 1-4 until there is no morpheme unit
 1-5. if positive_count_s=0 and negative_count_s=0 then result-s=0
     if positive_count_s=negative_count_s then result-s=positive_count_s
     result-s=positive_count_s-negative_count_s
```

Figure 3: Morphological analysis function

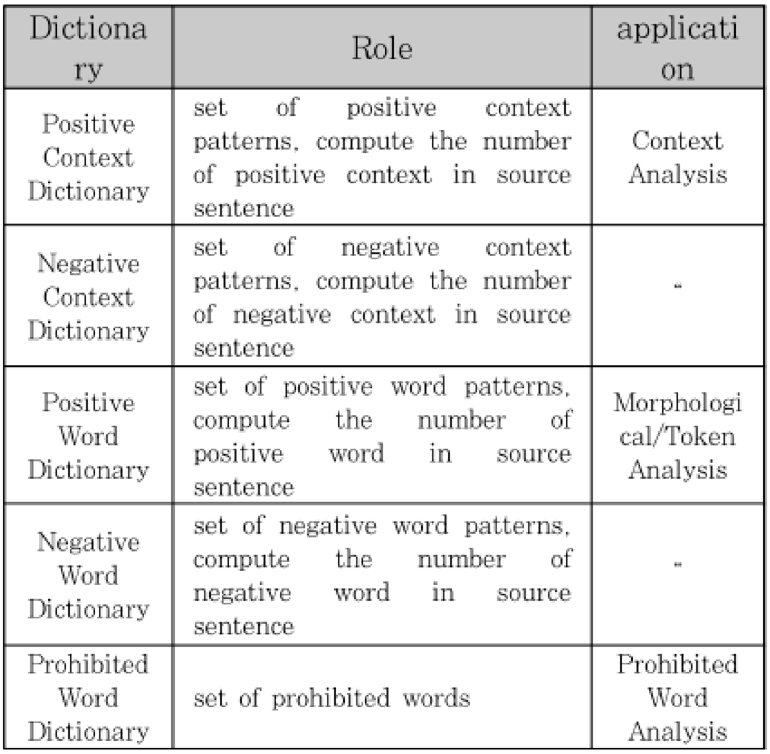


```
//Token Analysis  - if result-s=0 in previous stage
2.
 2-1. creation of tokens
 2-2. initialize positive_count_s and negative_count_s
 2-3. //Computation of the positive_count_s and negative_count_s using // the positive/negative word dictionary
      compute positive_count_s
      compute negative_count_s
     repetition step 2-3 until there is no token
 2-4 if positive_count_s=0 and negative_count_s=0 then result-s=0
      if positive_count_s=negative_count_s then result-s=positive_count_s
      result-s=positive_count_s-negative_count_s

//Prohibited word Analysis  - if result-s=0 in previous stage
3.
 3-1. //Computation of the positive_count_s and negative_count_s using // the prohibited word dictionary
      compute positive_count_s
      compute negative_count_s
      result-s=positive_count_s-negative_count_s
```

Figure 4: Token analysis function

The MapReduce function uses all five types of dictionaries as shown in Table 3 depending on the purpose. That is, it is a dictionary of positive words, negative words, positive context, negative contexts, and abuses.



Dictionary	Role	application
Positive Context Dictionary	set of positive context patterns, compute the number of positive context in source sentence	Context Analysis
Negative Context Dictionary	set of negative context patterns, compute the number of negative context in source sentence	〃
Positive Word Dictionary	set of positive word patterns, compute the number of positive word in source sentence	Morphological/Token Analysis
Negative Word Dictionary	set of negative word patterns, compute the number of negative word in source sentence	〃
Prohibited Word Dictionary	set of prohibited words	Prohibited Word Analysis

Table 3: Role of sentiment analysis dictionary

III. EXPERIMENTAL RESULTS

Table 4 shows the environment for performance analysis of the proposed system. The experimental environment consisted of 4 servers as a Hadoop-based parallel system, and CentOS was used as the operating system.

Components	Roles
OS, RE	Use of Hadoop for distributed storage, Supporting Java environment for processing some business logic
Crawler. HDFS Layer	Crawler: Gathering the source data from various SNSs HDFS: Distribution File system.
MapReduce Layer	Sentence Analysis. Text Mining. Emotion Analysis
MongoDB	Storing analyzed results by MapReduce in MongoDB
WAS. Web Server	Supporting Web applications using analyzed results

Table 4: Experimental environment

To analyze the performance of the proposed system, the following experiments were conducted. The experiment was conducted on 5 sets of actual Twitter data collected through a crawler.

First, it is a system performance experiment on crawling (data collection) and HDFS loading. Figure 6 compares crawling time and loading time for each data set. As shown in Figure 6, for the 1,000 data set, the crawling time was 9 seconds and the loading time was 1 second, and for the 100,000 data set, the crawling time was 753 seconds and the loading time was 7 seconds. As the number of data increases, both the crawling time and the HDFS loading time increase, but it was found that the crawling time is absolutely longer than the HDFS time. Therefore, it can be seen that the data loading load according to the amount of data in the proposed HDFS is insignificant. Also , it can be seen that crawling and HDFS loading are processed faster than time complexity in the proposed system .

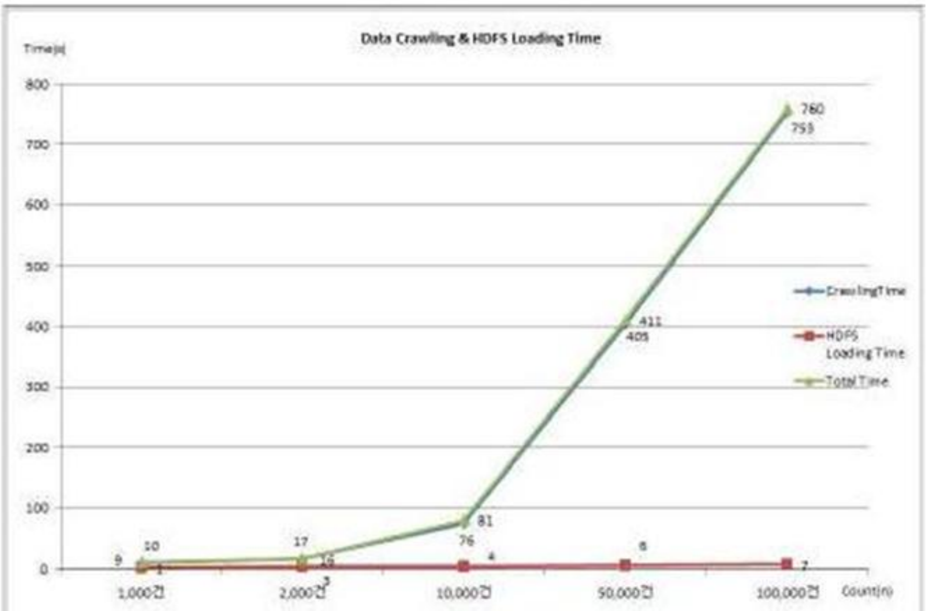


Figure 6: Crawl time and HDFS loading time

Figure 7 shows the memory load for each node when crawling by data set and loading HDFS. As shown in Figure 7, the nodes from the slave node to the slave node 3 (SN3) used the minimum to the maximum , and the master node (M) used the minimum to the maximum . The master node using the data distribution policy showed stable memory usage regardless of the size of the data set, and the slave node 1 used as the secondary server also showed relatively stable memory usage. However, the data-only servers, slave nodes 2 and 3, increased their memory usage as the size of the data set increased. In addition, according to the master node's data distribution loading policy, it can be seen that the load is distributed in a balanced way, rather than only on a specific node, so that the memory is used efficiently.

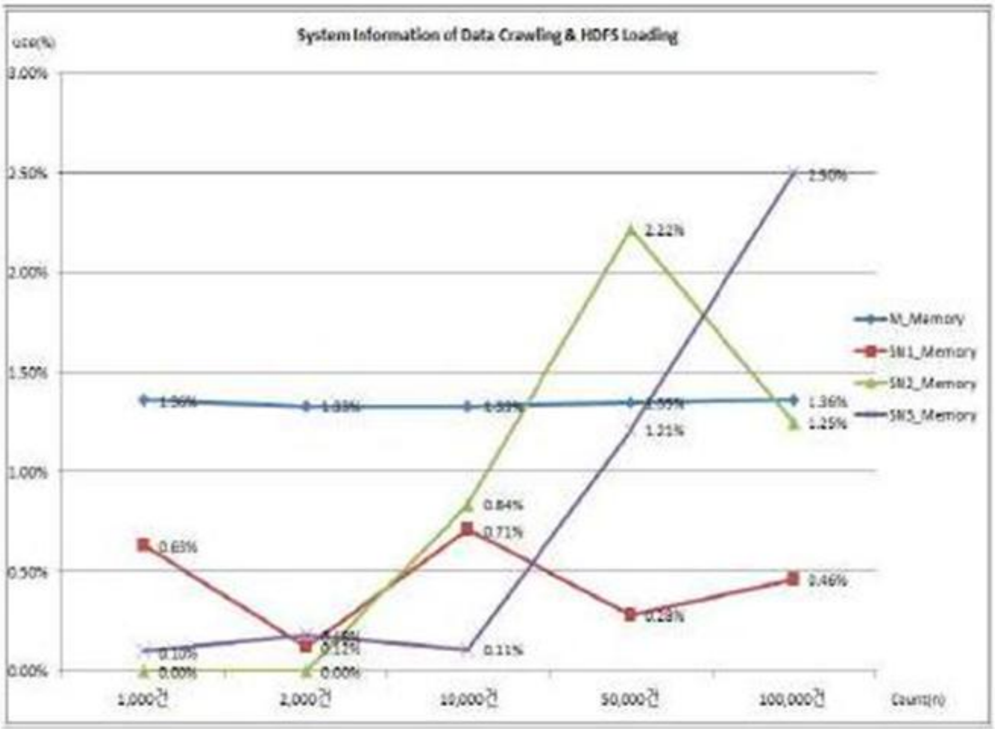


Figure 7: Memory load by node during crawling and loading

CPU load for each node during crawling and loading by data set . As shown in [Figure 8], in the case of the slave node and the slave node , the minimum to maximum usage was shown. In the case of the master node, the minimum to maximum usage is shown. As with memory usage, it was found that the load was not concentrated on a specific node, but balancing between slave nodes was achieved during the automatic parallel processing of HDFS. It was also found that the master node provides a stable environment for data collection and loading .

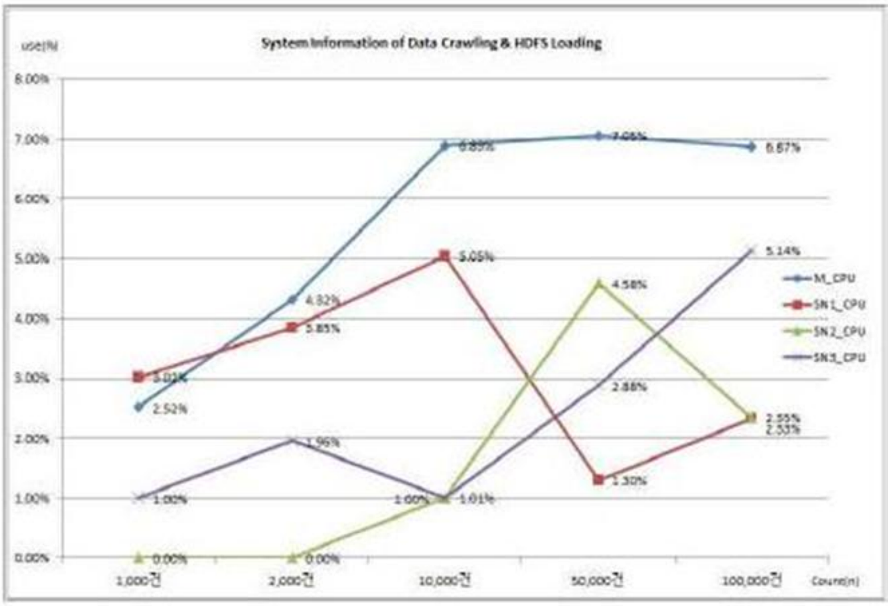


Figure 8: Load for each node during crawling and loading

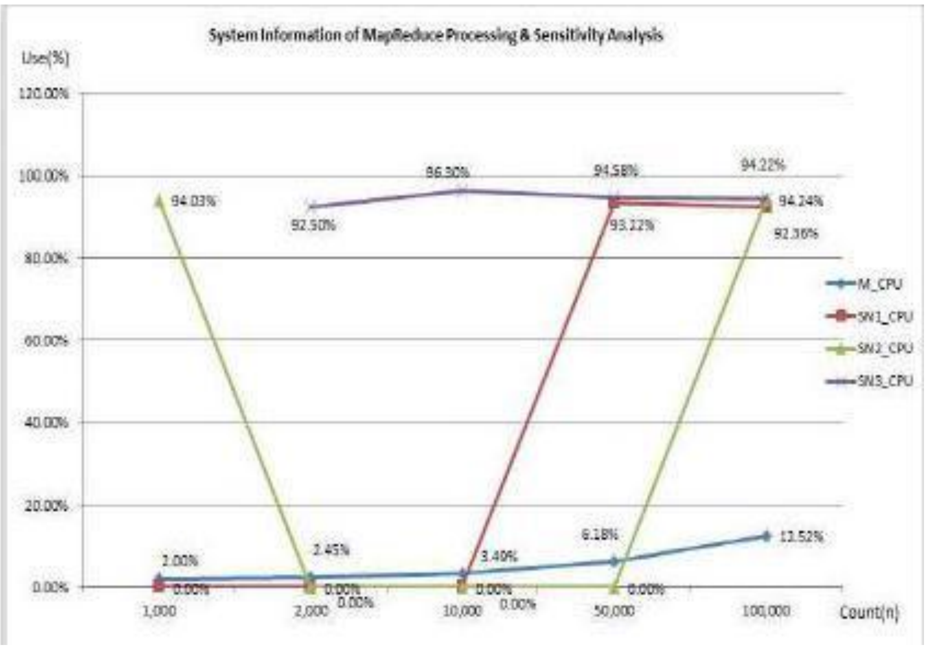


Figure 9: Load by node in MapReduce and sentiment analysis

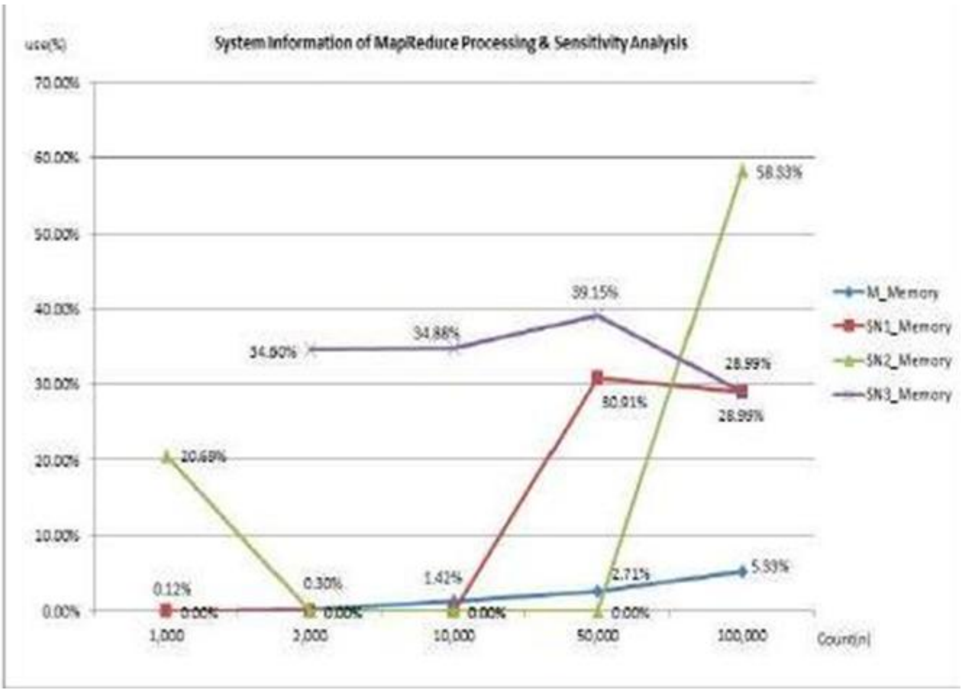


Figure 10: Memory load by node during MapReduce and sentiment analysis

Second, it is a system performance experiment on MapReduce and sentiment analysis. With the data set used in the first experiment, the degree of system load during MapReduce and sentiment analysis was tested. Figure 9 and Figure 10 compare the load and memory load of each node during MapReduce and sentiment analysis, respectively. As shown in the figures, in the case of the master node (M), the usage of the master node (M) is low because it is in charge of the management of the lower slave nodes rather than the actual analysis processing, whereas the slave nodes use most of the resources. It was shown that each slave node processes data in parallel with each other until the data set with the number of data sets of 10,000, but after the data set with the number of data sets of 50,000, as the amount of distributed data increases, all slave nodes show that the CPU is maximally used. Therefore, it can be seen that the proposed system achieves stable mutual parallel processing up to a certain level.

Similarly in Figure 10, it can be seen that the memory usage is low in the case of the master node, but the memory is distributed and used in the case of the slave node by mutual parallel processing. Therefore, in terms of memory usage, it can be seen that appropriate mutual parallel processing is being performed between slave nodes.

Therefore, when data were collected and processed using the proposed system and algorithm, the appropriate processing time was shown according to the number of data cases. It has been shown to provide an analysis environment.

IV. CONCLUSION

In this study, we proposed a MapReduce function with and which can analyze the user's emotions from a large amount of unstructured data generated from . The proposed system consists of four main servers operating in parallel based on the Hadoop system, and it consists of a MapReduce function with four main functions. Several experiments were conducted to analyze the performance of the proposed system. In other words, it was a system performance experiment related to crawling (data collection) and loading according to the amount of data, and a system performance experiment related to MapReduce and sentiment analysis. Through the experiment, the proposed method consumes less loading time according to the change in the amount of data , and shows stable performance as the system load is processed in parallel on each node without being concentrated on any one node. In addition, when data is processed using the proposed system and MapReduce sentiment analysis function in the performance experiment according to MapReduce and sentiment analysis, the load is not concentrated on a single node and is processed in parallel to provide a stable parallel analysis environment.

REFERENCES

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

[2]. Chang-Shing Lee, Mei-Hui Wang, "Automated ontology construction for unstructured text documents", Data & Knowledge Engineering, Vol.60, Iss.3, pp.547-566, 2007

[3]. B. Lee, J. Lim, J. Yoo, "Utilization of Social Media Analysis using Big Data", Jour. of the Korea Contents Association, Vol.13, No.2, pp.211-219, 2013

[4]. M. Song, S. Kim, "A Study of improving on prediction model by analyzing method Big data", The Journal of Digital Policy & Management, Vol.11, No.6, pp.103-112, 2013

[5]. Ah Tan, "Text mining: The state of the art and the challenges", Proc. of the PAKDD 1999, 1999

[6]. Q. Mei, C. Xhai, "Discovering evolutionary theme patterns from text: an exploration of temporal text mining" Proc. of the ACM SIGKDD international conference on knowledge discovery in data mining, pp.198-207, 2005

[7]. K. Park, K. Hwang, "A Bio-Text Mining System Based on Natural Language Processing", Jour. of KISS: computing practices, Vol.17, No.4, pp.205-213, 2011

[8]. B. Pang, L. Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends in Information Retrieval, Vol.2, No.1-2, pp.1-135, 2008

[9]. B. Kang, M. Song, "A Study on Opinion Mining of Newspaper Texts based on Topic Modeling", Jour. of the Korean Library and Information Science Society, Vol.47, No.4, pp.315-334, 2013

[10]. http://hadoop.apache.org/

[11]. Jing Han, Kian Du, "Survey on NoSQL database", Proc. of International Conference on Pervasive Computing and

[12]. Applications(ICPCA), pp.363-366, 2011

[13]. Fay Chang, R.E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", ACM Transactions on Computer System, Vol.26, Iss.2, 2008

[14]. S. Sivasubramanian, "Amazon dynamoDB: a seamlessly scalable non-relational database service", Proc. of the 2012 ACM SIGMOD'12, pp.729-730, 2012

[15]. Lars George, "HBase: The Definitive Guide", O'REILLY, 2011

[16]. Kristina Chodorow, "MongoDB: The Definitive Guide Edition", O'REILLY, 2013

[17]. B. Pang, ,L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval: Vol.2, No.1-2,pp.1-135,

[18]. S. Mukherjee, P. Bhattacharyya, "Sentiment Analysis in Twitter with Lightweight Discourse Analysis", Proc. of COLING 2012, pp.1847-1864, 2012

[19]. N. Godbole, S. Skiena, "Large-Scale Sentiment Analysis for News and Blogs", Proc. of the ICWSM'2007, 2007

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

[21]. H. Tang, S. Tan, X. Cheng, " A survey on sentiment detection of reviews," Expert Systems with Applications, Vol.36, pp.10760-10773,

[22]. Seth Gilbert, Nancy Lynch, Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services, ACM SIGACT New 33(2), pp. 51-59, 2002 .

[23]. E. Yu, Y. Kim, N. Kim, S. Jeong, "Predicting the Direction of the Stock Index by Using a Domain-Specific Sentiment Dictionary", Jour. of Intelligence and Information Systems, Vol.19, No.1, pp.

[24]. J. Dean, S. Ghemawat, "MapReduce; Simplified Data Processing on Large Clusters", Communications of the ACM, Vol.51, No.1, pp.107-113, 2008

[25]. Al-Khasawneh, Mahmoud Ahmad, Irfan Uddin, Syed Atif Ali Shah, Ahmad M. Khasawneh, Laith Abualigah, and Marwan Mahmoud. "An improved chaotic image encryption algorithm using Hadoop-based MapReduce framework for massive remote sensed images in parallel IoT applications." *Cluster Computing* 25, no. 2 (2022): 999-1013.

[26]. Sheheeda Manakkadu, Srijan Prasad Joshi, Tom Halverson, and Sourav Dutta. "Top-k User-Based Collaborative Recommendation System Using MapReduce." In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 4021-4025. IEEE, 2021.

[27]. Jeyaraj, Rathinaraja, and Anand Paul. "A Constrained 2D Bin Packing Model for MapReduce Task Scheduling." *IEEE Access* (2022).