A Real-time Knowledge Extracting System from Social Big Data using Distributed Architecture

Youngsub Han
Department of Computer and Information Sciences
Towson University
7800 York Road,Towson, MD, USA
1+443-991-9422
yhan3@students.towson.edu

Hyeoncheol Lee
Department of Computer and Information Sciences
Towson University
7800 York Road,Towson, MD, USA
1+410-704-2757
hlee23@students.towson.edu

Yanggon Kim
Department of Computer and Information Sciences
Towson University
7800 York Road,Towson, MD, USA
+1-410-704-3782
ykim@towson.edu

ABSTRACT

A huge amount of data is being generated by social media in real time. Accordingly, demands for extracting meaningful information from the social data have been dramatically increased. However, most of the previous research encompasses potential problems with data processing, management and analysis in real time. In this paper, we propose a distributed system architecture for generating meaningful information from text-based social data. The system collects data from multi-source channels, such as Twitter, YouTube, and The New York Times. Also, the system extracts terms and sentiment from each document using data mining technologies. In addition, the system uses HDFS, Map-reduce, and message service to handle the huge data. By analyzing keywords in texts and user account information, the system generates a summary of results including terms, sentiments and data variations for further analysis, including reputation, social trends, and customer reactions. The experiment results show that our approach is able to effectively process the social data in real time.

CCS Concepts

• Information systems→Data Mining; • Computing methodologies→Natural language processing; • Computer systems organization→Distributed architectures;

Keywords

crawling; data mining; sentiment analysis; natural language processing; distributed computing; Message driven processing; big data; Hadoop

1. INTRODUCTION

In these days, social data is being generated, produced and exchanged by the masses [1]. Demand for extracting meaningful information from social data has dramatically increased with the huge amount of data generated from social media. However, underlying architecture and the process to analyze the huge amount of social data has yet to be established. To extract meaningful information from the data, the system requires effective data management and various processing techniques that can handle huge amounts of data such as multi-processing, NLP (Natural Language Processing), machine learning, topic model, etc. Especially, distributed computing is one of the biggest issues processing the data. For these reasons, we propose a system for analyzing the social data in real time.

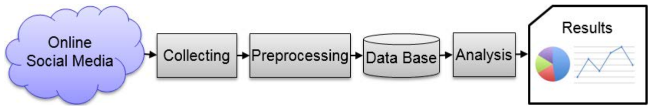


Figure 1. General Process of data collection and management tool for online social network analysis

Figure 1 shows the general process of data collection and management tool for online social network analysis. Firstly, it collects raw data from an online social network using API (application programming interface). The APIs are usually provided by the social network provider. Secondly, the data preprocessor extracts information in raw data, such as keywords and topic in texts. Then, the raw data and extracted information is saved into a data storage. Previous approaches mainly used a relational database as a main data storage. Finally, the data analyzer examines the data using data mining algorithms and generated knowledge.

Table 1. The Comparison of related approaches

	Source Channel	Data Store	Distributed Processing
Song et al[1].	Twitter	Relational, Key-value pairs	No
TwitterEcho[14]	Twitter	Not given	No
Byun et al[15].	Twitter	Relational	No
Twitter Zombie[16]	Twitter	Relational	No
TwitHoard[17]	Twitter	Graph DB	No
TrendMiner[18]	Twitter	Key-value pairs	No
TwitIE[19]	Twitter	Not given	No
ESA[20]	Twitter	Not given	No
Baldwin et al[21]	Twitter	Flat files	No

RACS'15, October 9–12, 2015, Prague, Czech Republic.
© 2015 ACM. ISBN 978-1-4503-3738-0/15/10 $15.00.
DOI: http://dx.doi.org/10.1145/2811411.2811481

A lot of researchers have developed data collection and management systems for online social network analysis, which is summarized in Table 1. Most of the previous research encompasses potential problems with data processing, management and analysis. Firstly, the data storages of the previous approaches are based on a relational database that may cause performance issues when a huge amount of datasets ranging from a few terabytes to multiple petabytes needs to be handled. Secondly, they do not support distributed processing, which may slow down processing time. Lastly, they collect data from only a single source channel, such as Twitter. To analyze trends of society accurately, data should be collected from multiple online social networks as opposed to only one. For these reasons, we propose a system architecture for social data analysis. The system consists of a multi-source channel data collector, term extractor, and sentiment analyzer. The system uses the HDFS (Hadoop Distributed File System) and message service to handle the data.

The following is in the next chapter of this paper: Section II presents previous related works. It addresses Apache Hadoop, topic modeling and sentiment analysis. Section III presents the system including the method for processing the social data. Section IV presents experiment results of the system with sample data. Section V presents our conclusion of this research and future works.

2. RELATED WORKS

2.1 Apache Hadoop

Apache Hadoop is an open source software that allows us to process and manage data in scalable and distributed manners [2]. The main characteristic that separates Apache Hadoop from other database management systems is that it allows distributed processing of large data sets across clusters of computers with simple programming models. It also supports job scheduling and cluster resource management. The distributed computing techniques provide high scalable processing capabilities that reduce processing time for big data [13]. Hadoop uses Hadoop Distributed File System (HDFS) that splits files into large blocks and distributes blocks into nodes in clusters. It runs a range of clusters of commodity machine. Map-Reduce is a distributed data processing model and execution environment that processes data parallel in the nodes. Hadoop is widely used in the industry to process and manage in big data. It can be considered the best candidate for social data analysis since it requires handling and analyzing a vast amount of data.

2.2 Java Massaging Service (JMS)

Messaging is an approach of loosely coupled distributed communication. Message-oriented middleware supports asynchronous communication between distributed components as senders and receivers using message-passing. A sender sends a message to designated queues on a server. A receiver receives the message from the queue asynchronously. The Java Message Service (JMS) API is a Java Message Oriented Middleware API. It provides a vast collection of interfaces to develop a message-oriented service. The JMS supports two models: point-to-point and Publish-subscribe. In the point-to-point model, messages are passed to an individual consumer which maintains a queue of incoming messages. Publish-subscribe model is based on the use of topics that can be subscribed to by clients. In this model, messages are sent to topics by other clients and are then received in an asynchronous mode by all the subscribed clients. The benefit of this approach is that a sender does not necessarily need to know about the status of a receiver to send a message. The JMS can be used to integrate heterogeneous platforms using JVM and it helps to reduce system bottlenecks, increase scalability, and respond more quickly to change.[22][23][24]

2.3 Sentiment Analysis

Several texts written by users in online social networks encompass the users' emotional states about specific topics, such as events, products, and services. The purpose of sentiment analysis is to extract people's opinion or emotional states towards certain topics from the texts [3][4]. Opinion and emotional state can be major features in online social network analysis. Thus, demands on automatic sentiment analysis of texts generated from online social network have dramatically increased and those topics have received a lot of attention from researchers. There are two main approaches to extracting sentiment from texts. The first approach is lexicon-based sentiment analysis, which is found using pattern matching with a pre-built lexicon [6][7][8][9]. The second approach is classification-based sentiment analysis, also known as supervised classification. It builds a sentiment classifier using a train set that contains labeled texts or sentences and tests new texts using the classifier [10][11]. Most approaches focused on classifying whether a given text contains a positive or negative opinion [5]. We developed our own sentiment analysis model in previous research, which guarantees higher accuracy than existing approaches and can be broadly used in any social networking sites without requiring additional information. It will be integrated into our system.

3. SYSTEM ARCHITECTURE AND IMPLEMENTATION

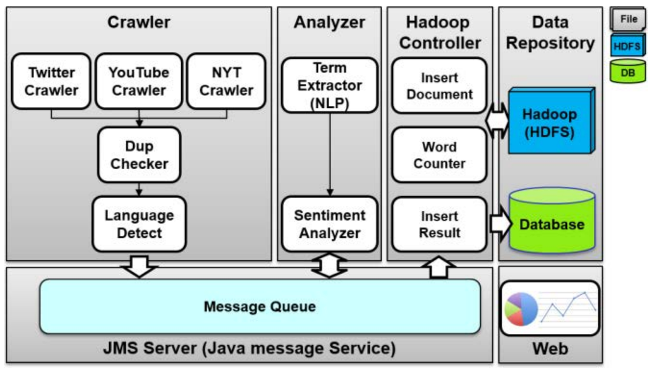


Figure 2. The system architecture and service flow

In this section, we describe our system. The system consists of three main phases which are the data collecting, the analyzing, and the Hadoop controlling. The first phase is data collection. The crawler collects data from Twitter, YouTube, and the New York Times using their API. The data is a bunch of Tweets, articles of The New York Times, and YouTube comments. We called each of them 'document'. In the case of Twitter, we collect tweets in two ways, which are the search-based collecting and seed-based collecting using Twitter API. In the case of The New York Times, they also provide search-based APIs to collect articles. In the case of YouTube, we manually find shared videos related to the target as seeds. Then, we collect comments of the videos as documents. After collecting the data, the duplication checker filters duplicated data and the language detector identifies the language of each document. The second phase is analysis. In this phase, the term extractor extracts terms using Stanford core NLP, and the sentiment analyzer made by Lee et al [12] analyzes the sentiments from each of the documents. The last phase is the Hadoop controlling. We designed our own directory architecture in HDFS (Hadoop distribute file system) to analyze and retrieve the data effectively.

All the phases are designed to support multi-processing depending on performances of each phase and hardware resources since there are gaps between the phases in performance. Also, all the data is communicated in the message server using JMS (Java messaging service).

3.1 Data Collection

3.1.1 Twitter Crawler

To collect data, we developed three crawlers. The first one is Twitter crawler. The crawler collects tweets in two ways: search crawler and seed crawler. The seed crawler uses Twitter Stream API with screen names of the user accounts as seeds. The search crawler uses Twitter search API with keywords related to a target to be analyzed such as the companies, the politicians, the products, or the issues.

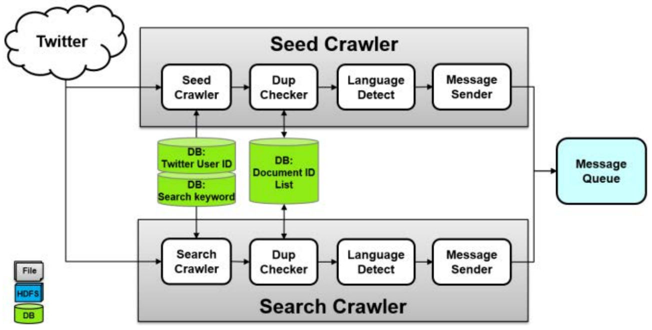


Figure 3. Twitter Crawler architecture and flow

3.1.1.1 Seed Crawler

The seed crawler retrieves tweets from the selected Twitter accounts. We call a Twitter account a seed. Also, all the accounts are equivalent to a Twitter screen name. Because the crawler uses only Twitter user id, the crawler automatically converts the screen name into the corresponding user id. While a screen name can be changed by the user, a user id is permanently used in Twitter. All account information is stored in the database with the screen name and the user ids. Twitter provides the latest 3200 tweets by each seed. The seed crawler needs Twitter development application keys. Twitter allows 180 requests per key in 15 minutes, and a request includes 200 tweets.

3.1.1.2 Search Crawler

The search crawler retrieves tweets by keywords related to the target companies. All the keywords are refined by hand. The crawler requests tweets with the keywords using Twitter search API, then Twitter gives tweets including the keywords. All keywords are stored in the database with the company information. Twitter provides the latest 9 days of tweets by each keyword in our experiment. The search crawler also needs a twitter development application key. Twitter allows 180 requests per key in 15 minutes, and a request includes 100 tweets.

3.1.2 YouTube Crawler

YouTube allows us to collect data such as video information, user profiles, and comments with APIs, which are provided by YouTube. In our research, we used a YouTube collecting tool, which was developed by Lee et al [12], that automatically collects comments posted on YouTube videos chosen as seeds. The crawler retrieves the comments from the video, which are searched by the keyword related to targets used in Twitter search crawler.

3.1.3 The New York Times News Crawler

The New York Times also allows us to collect the articles with APIs, which are provided by The New York Times. In our research, we developed a collecting tool for The New York Times that automatically collects articles using by the keyword related to 22 companies used in Twitter search crawler.

3.1.4 Duplicate tweets checking

To prevent collected data duplication, we developed a duplication checker using database. All the data should be checked before being sent to the analyzer, because all the data providers do not care about the duplication from our collecting requests.

3.1.5 Language detector

We developed a language detector for filtering out the data not written in English using Microsoft Language Detection Module to analyze, because we only used the English NLP (Natural Language Process) tool to extract terms and sentiments in this research.

3.2 Analysis

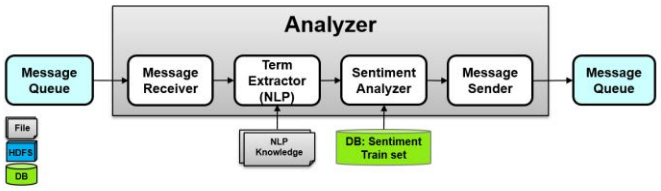


Figure 4. Analysis architecture and flow

3.2.1 Term Extractor

To extract terms in a document, we developed the term extractor. Term can be used to discover what issues are exchanged in the data. So, the word counter calculates the top 100 keywords which are most frequently used in the tweets of each company by date. Extracted terms is one of important features used to discover opinions. We used a Natural Language Processing (NLP) software, which is Stanford Core NLP, made by The Stanford Natural Language Processing Group. This software is an integrated suite of natural language processing tools for English. It provided refined and sophisticated results based on English grammar. The term extractor extracts specific POS (part of speech) such as noun, pronoun, adjective, verb, and adverb as a term because the POSs may have meaningful opinions from the document in general. Then, the extractor filters stop-word, and then, sends the data to the message queue with all the extracted results.

3.2.2 *Sentiment Analyzer*

Table 2. The examples of comparing sentiments by the analyzer with sentiments by human coders

Tweet	Sentiment			Correctness
	by analyzer		human coders	
	Score	Sentiment	Sentiment	
Save 30% on Flu vaccines at Walmart	60	P	P	O
RT @abbytucks: I love target	58	P	P	O
I think I just met the love of my life at Walgreens	52	P	P	O
Comcast sucks!	47	N	N	O
@comcast GO TO HELL	44	N	N	O
@ComcastMike Nope. I hate @comcast and I'm leaving for better.	49	p	N	X

Sentiment analysis means to discover an opinion, which has a positive or negative meaning from text. So, we used Sentiment Analyzer, which is an automated sentiment analysis tool made by Lee et al [12] for short-term text in social media. In our research, we used 1,000 document for train sets, which are labeled by hand. The analyzer provides us the score of positivity from 100% to 0%. As following our experiment, we find a threshold point, which is 48% of positivity [12].

As shown in Table 2, we compared 100 sample documents, which are randomly sampled from collected data, by the analyzer with sentiments by human coders. The accuracy of correctness from the comparison is 90 %. It is a similar to Lee's results (89%). After analyzing sentiment, the analyzer sends all the data to message queue with sentiment score. We added a module for communicating the results with the documents between the message server and the analyzer.

3.3 Hadoop Controller

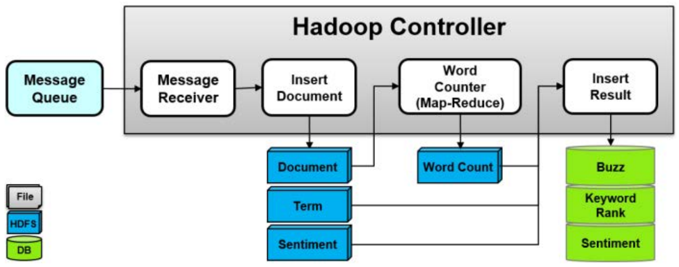


Figure 5. Hadoop controller architecture and flow

We developed Hadoop Controller to store the data, calculate the word counts and sentiment scores using HDFS. Figure 5 shows the architecture and flow of the Hadoop controller. The message receiver gets the data from the analyzer through the message queue; the data is stored in HDFS by the document, term, and sentiment; and then, the word counter calculates the word count and their ranks. Afterward, all the results are stored in the database.

3.3.1 *Insert Document*

We designed our own directory architecture to store, analyze, and retrieve effectively, in HDFS. All the data is stored by channel, company, category and date. Document, term, and sentiment are stored separately into the directory, because when a user requests data of specific channel, company and date, the system easily provides the data to the user without additional processing.

3.3.2 *Word Counter*

The word counter counts and sorts how frequently a word appears in each context of the data using Map-Reduce. All the counted word are saved in "term" directory in HDFS. Then, the Hadoop controller stores the top 100 terms into our database to show in our demo web page. In our results, the terms of about 15 million of documents can be calculated in about 20 minutes by channel, company and date as shown in the Figure 5.

3.4 Messaging Service and Multi-processing

We uses the JMS to communicate data between the crawler, the analyzer, and the Hadoop controller to minimize delay time among all processes because of the performance gap between them. Because of the characteristics of the JMS, we applied multi-processing to our system easily. It also provided maintainability and flexibility of processes controlling.

4. EXPERIMENT RESULT

We select 22 companies as targets to analyze from S&P 100, a stock market index of United States stocks maintained by Standard & Poor's. The 22 companies were categorized into three groups by their business types as shown in the Table 3.

Table 3. 22 target companies for analyzing

Category	Company
Information Technology	Apple, Amazon, Cisco, eBay, Facebook, Google, HP, IBM, Intel, Microsoft, Oracle, Qualcomm
Retail	CVS, Costco, Home Depot, Lowe's, Walmart, Target, Walgreen
Telecommunications	AT&T, Verizon

To collect the data, we assigned 6 app keys for seed crawler. 6 app keys allows 1,080 queries per 15 minutes. Because a query includes 100 tweets, 108,000 tweets can be crawled in every 15 minute. It means that the crawler can collect 7,200 tweets per minute. We also assigned 28 app keys for search crawler. The 28 app keys allow 5,040 queries per 15 minutes. So, 504,000 tweets can be crawled in every 15 minutes. This means that the crawler can collect about 33,600 tweets per minute. Table 4 shows examples of keywords and user accounts to collect data.

Table 4. The examples of keywords and user accounts for collecting

Category		Count	Example
Keyword	Company	55	AT&T, Csco, Cisco, Home depot,
User account	Company	409	AT&T Business (@ATTBusiness), Cisco Services (@CiscoServices), Home Depot Inc (@HomeDepot)
	News	119	Wall Street Journal (@WSJ), Reuters Business (@ReutersBiz), USA today (@USATODAY)
	Investment company	98	Fidelity Investments (@Fidelity) Vanguard (@Vanguard), Goldman Sachs(@GoldSachsNews)

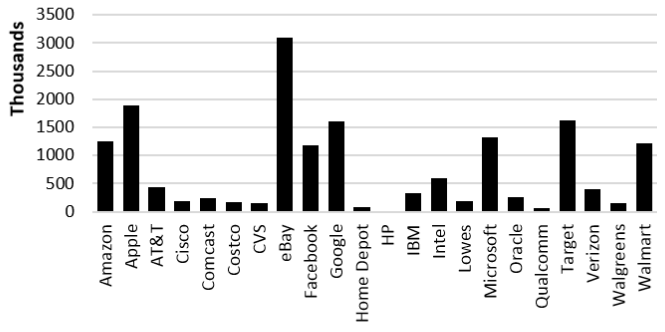


Figure 6. The number of data by the company

From Dec 19, 2014 to Jan 18, 2015, the crawler collected 16,479,483 documents. On average, that is about 532,226 documents daily. We can expect that about 2 billion documents will be collected per year in our system. Figure 6 shows the total amount of documents by the companies.

Table 5. The comparison of the number of processed documents by the crawler, analyzer, and Hadoop controller

Processing / 10 min	Crawler	Analyzer	Hadoop Controller
test 1	2,109	599	1,967
test 2	2,035	566	2,149
test 3	2,004	455	2,161
test 4	2,120	489	2,294
test 5	2,485	448	2,171
test 6	2,464	443	2,155
test 7	2,593	451	2,180
test 8	2,563	491	2,152
test 9	2,698	483	2,260
test 10	2,670	616	2,727
average	**2,374**	**504**	**2,222**

Table 5 shows comparisons of the performance by 10 minutes per documents between crawler, analyzer, and Hadoop controller. On average, the crawler handles about 2,374 tweets in 10 minutes. The analyzer handles about 504.10 tweets per 10 minutes. This is about 4.7 times slower than the crawler in the average as shown in the test 5. The Hadoop controller handles about 2221.60 tweets in 10 minutes. This is about 1.43 times slower than the crawler in the average as shown in test 1. So, we need multi-processing (parallel) to minimize the gaps.

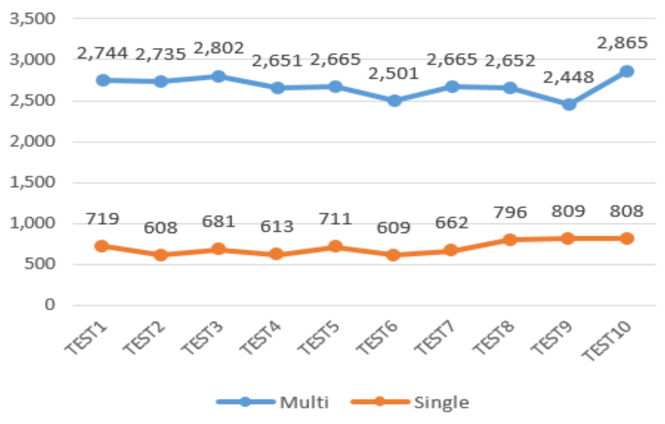


Figure 7. Comparison between Multi-processing and Single-processing.

According to the result, we applied multi-processing to our system. Also, we examined how the system performed to process data in real-time. As shown in the Figure 7, results of the multi-processing is about 2,673 documents per 10 minutes on average. This means that the system can cover an average of the crawler (2,374 documents per 10 minutes in average). If the crawler has more capacity, for example more app keys or multi crawler, we can execute analyzers and Hadoop controller following as the result. All the stored terms will be counted by the word counter using the Map Reduce technology every 10 minutes.

As shown in Figure 8, 9, 10, we developed a web-based system to report summaries of results including the number of documents, word counts, and sentiments by the companies and dates. The summaries are stored in the database for our web system.

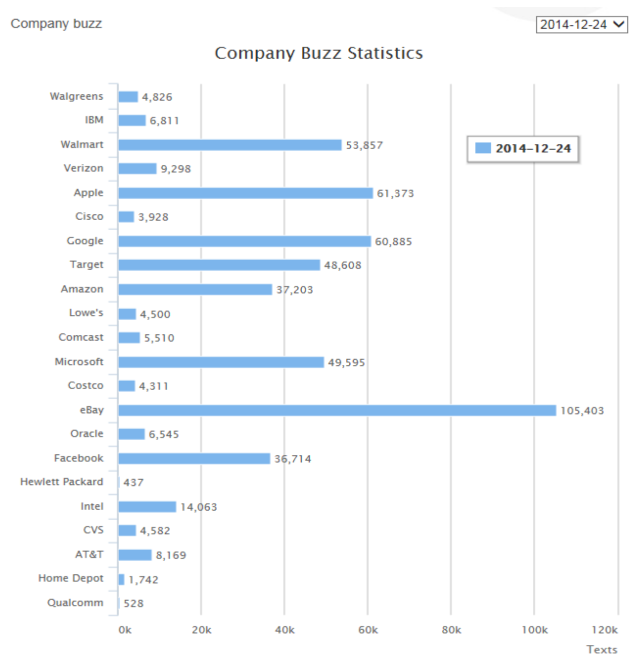
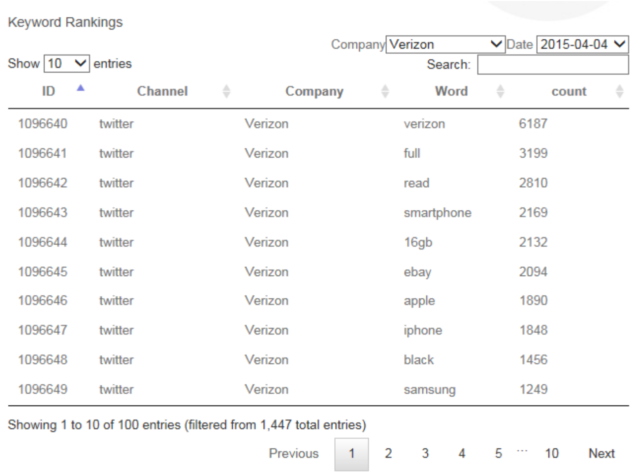


Figure 8. Company buzz statistics on web-based system.



Keyword Rankings

Company Verizon Date 2015-04-04

Show 10 entries Search:

ID	Channel	Company	Word	count
1096640	twitter	Verizon	verizon	6187
1096641	twitter	Verizon	full	3199
1096642	twitter	Verizon	read	2810
1096643	twitter	Verizon	smartphone	2169
1096644	twitter	Verizon	16gb	2132
1096645	twitter	Verizon	ebay	2094
1096646	twitter	Verizon	apple	1890
1096647	twitter	Verizon	iphone	1848
1096648	twitter	Verizon	black	1456
1096649	twitter	Verizon	samsung	1249

Showing 1 to 10 of 100 entries (filtered from 1,447 total entries)

Previous 1 2 3 4 5 … 10 Next

Figure 9. Top 100 Keywords on web-based system.

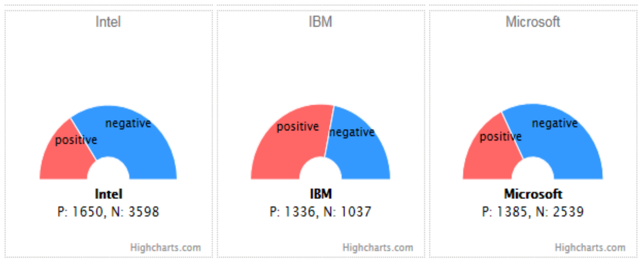


Figure 10. Sentiment analysis on web-based system.

5. CONCLUSION

In our research, we proposed a system to discover meaningful information from social data in real time. To discover the information, the system extracts terms and sentiments using a NLP tool and sentiment analyzer. Also, the system is designed to support the multi-processing and the message service improves maintainability, capability of multi-processing, and flexibility of processes controlling. Most importantly, the system uses the Hadoop and Map Reduce technologies to handle the mass amount of data. We also designed a data directory architecture, which is designed to retrieve the data effectively from HDFS. Now, the system can collect and process about 532,226 documents daily related to the 22 target companies in real-time, and about two-billion of the documents are stored in the system repository. The experiment results show that our approach is able to effectively process the social data in real time.

For future work, we are developing a process controller named "Dynamic Process Controller" to improve the efficiency of multi-processing. The dynamic process controller will work as a load balancer in our system to mitigate the gaps depending on the system resources such as usages of memory and CPU. We expect that the controller dynamically will control the number of processes according to their hardware resources.

6. REFERENCES

[1] M. Song, M. Kim and Y. Jeong, "Anlyzing the Political Landscape of 2012 Korean Presidential Election in Twitter", Intelligent Systems, IEEE, vol 29, Issue 2, pp.18-26, March 2014.

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

[3] A. Sharma and S. Dey, "A comparative study of feature selection and machine learing techniques for sentiment anlaysis", Proceedings of the 2012 ACM Research in Applied Computation Symposium, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3.

[4] P. Goncalves, M. Araújo, F. Benevenuto, and M. Cha, "Comparing and combining sentiment analysis methods", Proceedings of the first ACM conference on Online social networks, October 2013, pp. 27-38, ISBN: 978-1-4503-2084-9.

[5] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification", Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, June 2009, pp. 1275-1284, ISBN: 978-1-60558-495-9.

[6] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From Tweets to Polls:Linking Text Sentiment to Public Opinion Time Series", Proceedings of the International AAAI Conference on Weblogs and Social Media, May 2010, pp. 122-129.

[7] O. Kucuktunc, B. B. Cambazoglu, I. Weber, and H. Ferhatosmanoglu, "A Larege Scale Sentiment Analysis for Yahoo! Answers", Proceedings of the fifth ACM international conference on Web search and data mining, February 2012, pp. 633-642, ISBN: 978-1-4503-0747-5.

[8] M. Taboada, J. Brooke, M. Tofiloski, K. Voll and M. Stede, "Lexicon-Based Methods for Sentiment Analysis", Computational Linguistics, vol 37, Issue 2, June 2011, pp. 267-307, doi: 10.1162/COLI_a_00049

[9] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends Information Retrieval, vol 2, Issue 1-2, January 2008, pp.1-135, doi: 10.1561/1500000011

[10] A. Sharma and S. Dey, "A Boosted SVM based Sentiment Analysis Approache for Online Opinionated Text", Proceedings of the 2013 Research in Adaptive and Convergent Systems, October 2013, pp. 28-34, ISBN: 978-1-4503-2348-2

[11] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis", Proceedings of the 2012 Research in Adaptive and Convergent Systems, October 2012, pp. 1-7, ISBN:978-1-4503-1492-3

[12] H. Lee, Y. Han and K. Kim, "Sentiment Analysis on Online Social Network Using Probability Model", Proceedings of the The Sixth International Conference on Advances in Future Internet, November 2014, ISBN:978-1-61208-377-3

[13] X. Liu, N. Iftikhar and X. Xie, "Survey of real-time processing systems for big data", Proceedings of the 18th International Database Engineering & Applications Symposium, July 2014, pp.356-361, ISBN: 978-1-4503-2627-8

[14] M. Boanjak and E. Oliveira. "TwitterEcho - A distributed focused crawler to support open research with twitter data", International conference companion on World Wide Web, April 2012, pp. 1233–1239, ISBN: 978-1-4503-1230-1

[15] C. Byun, H. Lee, Y. Kim, and K. K. Kim. "Twitter data collecting tool with rule-based filtering and analysis module", International Journal of Web Information Systems, Vol 9, Issue 3, pp. 184-203, 2013

[16] A. Black, C. Mascaro, M. Gallagher, and S. P. Goggins. "Twitter Zombie: Architecture for capturing, socially transforming and analyzing the Twittersphere", International conference on Supporting group work, October 2012, pp. 229–238. ISBN:978-1-4503-1486-2

[17] Y. Stavrakas and V. Plachouras, "A platform for supporting data analytics on twitter challenges and objectives" Intl. Workshop on Knowledge Extraction & Consolidation from Social Media, (Ict 270239), 2013.

[18] D. Preotiuc-Pietro, S. Samangooei, and T. Cohn, "Trendminer : An architecture for real time analysis of social media text", Workshop on RealTime Analysis and Mining of Social Streams, 2012,pp. 4–7.

[19] K. Bontcheva and L. Derczynski, "TwitIE: an opensource information extraction pipeline for microblog text", International Conference on Recent Advances in Natural Language Processing, 2013.

[20] J. Yin, S. Karimi, B. Robinson, and M. Cameron "ESA: emergency situation awareness via microbloggers" Proceedings of the 21st ACM international conference on Information and knowledge management, October 2012, pp. 2701–2703.

[21] T. Baldwin, P. Cook, and B. Han "A support platform for event detection using social intelligence," Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, April 2012, pp 69–72.

[22] Mark Richards, Richard Monson-Haefel, and David A Chappell, "The Advantages of Messaging," in Java Message Service, 2nd ed., Ed. California: O'Reilly Media, May 2009, pp. 3-5, ISBN:978-0-596-52204-9

[23] Michael Menth, Robert Henjes, Christian Zepfel, and Sebastian Gehrsitz, "Throughput Performance of Popular JMS Servers," Proceedings of the joint international conference on Measurement and modeling of computer systems, June 2006, pp 367-368, ISBN:1-59593-319-0

[24] Mirco Musolesi, Cecilia Mascolo, Stephen Hailes, "Adapting asynchronous messaging middleware to ad hoc networking," Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing, October2004,pp121–126,ISBN:1-58113-951-9