International Journal of Research in Advent Technology, Special Issue, March 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Sentiment Analysis for Demonetization using Big Data and Hadoop Framework

Abir Ojha and Nirmal Kumar Gupta
Department of Computer Science and EngineerinG, Jaypee University Anoopshahr
Email: coolabir123@gmail.com ,*nirmalgpt@gmail.com*

Abstract- Social media websites have emerged as one of the platforms to raise users' opinions and influence the way any business is commercialized. Opinions of people play an important role in analyzing how the propagation of information impacts the lives in a large-scale through the network like Twitter. Tweets which are posted by millions all over the world can be used to analyze consumers' opinions about a specific product, its services and campaigns. In the recent years, these tweets have proven to be a valuable source of information, which are important for the success of any brand, business or the career of a politician. Here, in our research, we have adopted Sentiment Analysis with an approach to extract positive and negative tweets by using parts of the speech. This approach manifests in the design of a software toolkit that facilitates the sentiment analysis. The purpose of our research paper is to find out the views of different people on demonetization by analyzing their tweets from Twitter. The first phase of our research involves the extraction of data from twitter and the final phase of the work is aimed at using text-mining techniques, like tokenization, and how to use this to build classifier which can predict the sentiment of people of each tweet.

Index Terms- Social media, Big data, Sentiment Analysis, Tokenization, Hadoop.

1. INTRODUCTION

Sentiments of human beings fundamentally indicate their attitude, feelings or opinions about a specific matter. The analyses of such sentiments can postulate very beneficial and insightful information. In our research, a binary opposition in opinions has been assumed (e.g., for/against, like/dislike, good/bad, etc.); that means, a polarity is presumed. As humans are subjective creatures, their opinions are significant. Being able to interact with the people on that level has many advantages for information systems. Sentiment analysis involves Neuro-Linguistic Programming, statistics or machine learning methods to extract, identify or characterize the sentiment content of a text unit or sentiment content in the text of tweet or speech in our research work. Sometimes, it is mentioned as opinion mining; although in this case the emphasis is on the extracted meaningful insights from raw twitter data. The insights from sentiment analysis can provide the answers to questions like, "Is this product review a positive or negative?" or "Is this email of the customer satisfied or dissatisfied?",etc. Based on the sample of tweets, by using opinion mining, we can find the answers of the questions like "How the people are responding to a particular advertisement, marketing, product release, news item?" and"How have bloggers' attitudes towards the president changed since the election?" But the key exciting task involves extracting and analyzing the useful information obtained from this content. The unstructured nature of the content and the informal language used in writing these contents added up the complication and it opened a new area of research called Opinion Mining and Sentiment Analysis. There are several ways by which sentiment analysis can be performed on raw data. But, following two techniques are mainly used for opinion mining and sentiment analysis:

(1) Machine learning based techniques
(2) Lexicon based techniques

In machine learning based techniques, various machine learning algorithms are used in classifying the sentiment. Both supervised and unsupervised learning algorithms can be used in classifying the content. In Lexicon based techniques, a sentiment dictionary containing sentiment words is used for sentiment classification. This dictionary contains polarity of each word whether they are positive, negative and objective words. Polarity of the opinion words can be determined by matching those opinion words with dictionary words. In our research, we use a dedicated framework known as Hadoop. Hadoop is an Apache open source framework written in Java that allows distributed processing of large datasets across clusters of computers using simple programming models. A Hadoop frame-worked application works in an environment that provides distributed storage and computation throughout clusters of computers. Hadoop is designed to scale up starting from single server to thousands of machines, each offering local computation and storage. Hadoop can be used to work directly with any mountable distributed file system (e.g., Local FS, HFTP FS, S3 FS, etc.), but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS). HDFS is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters of thousands of computers of small computer machines in a reliable, fault-tolerant manner. HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more

slave DataNodes that store the actual data. A file in an HDFS namespace splits into several blocks and those blocks are stored in a set of DataNodes. The NameNode determines the mapping of blocks to the DataNodes. The DataNodes manage the read and write operation within the file system and also take care of block creation, deletion and replication of databased on instructions given by NameNode. HDFS provides a shell-like file system and a list of commands are available to interact with the file system. In our research, we use the Flume tool of Hadoops' framework in extracting data based on few keywords. In the next step, we use HDFS to store our metadata. We also use the piglatin language and the pig tool of Hadoop to internally run classification in classifying the individual tweets into positive and negative. Then, the polarity of individual tweets can be aggregated to finally compare them and to find out the polarity of the majority.

2. RELATED WORK

With the population of blogs and social networks, opinionmining and sentiment analysis became a field of interest for many researches. A very broad overview of the existing work was presented in Pang and Lee [7]. In their survey, the authors describe the existing techniques and approaches of an opinion-oriented information retrieval system. However, there are very few researches in opinion mining considered blogs and even much less addressed microblogging. In [10], the authors use web-blogs to construct a corpora for sentiment analysis and use emotion icons assigned to blog posts as indicators of users' mood. The authors applied SVM and CRF learners to classify sentiments at the sentence level and then investigated several strategies to determine the overall sentiment of the document. As a result, the winning strategy is defined by considering the sentiment of the last sentence of the document as the sentiment at the document level. Read in [9] authors used emotion icons, such as ":-)" and ":-(", to form a training set for the sentiment classification. For this purpose, the author collected texts containing emotion icons from Usene newsgroups. The dataset was dividedinto "positive" (texts with happy emoticons) and "negative" (texts with sad or angry emoticons) samples. Emotion icon strained classifiers (SVM and Naive Bayes) were able to obtainup to 70% of an accuracy on the test set. In [11], authors used Twitter to collect training data and then to perform a sentiment search. The same approach is defined by [9]. The authors construct a corpora by using emoticons to obtain "positive" and "negative"samples, and then use various classifiers. The best result was obtained by the Naive Bayes classifier with a mutual information measure for feature selection. The authors were able to obtain up to 81% of accuracy on their test set. However, this method exhibited a poor performance with three classes: "negative", "positive" and "neutral". As sentiment analysis is one of the most popular trend in today's world, lot of work has been done in this segment.There has been a lot of research work in the area of sentiment analysis. Recent works in this area also involve the use of mathematical approach that uses formula for the sentiment valuedepending on the proximity of the words with adjectives like "excellent", "worse", "bad", etc. In our research, we use the Hadoop cluster for distributed processing of the textual data.

3. RESEARCH METHODOLOGY

3.1 Hadoop

We use the Hadoop platform as it is designed to solve problems which involves a lot of data for processing. Hence, it is best suited for our work which requires the storage and processing of large twitter data sets and uses the divide and concur methodology for processing. It is used to handle large and complex unstructured data or semi-structured data (as data used in our research). Twitter data being relatively unstructured or semi-structured can be best stored using Hadoop. Hadoop also finds a lot of applications in the field of online retailing, search engines, finance domain for risk analysis, etc.

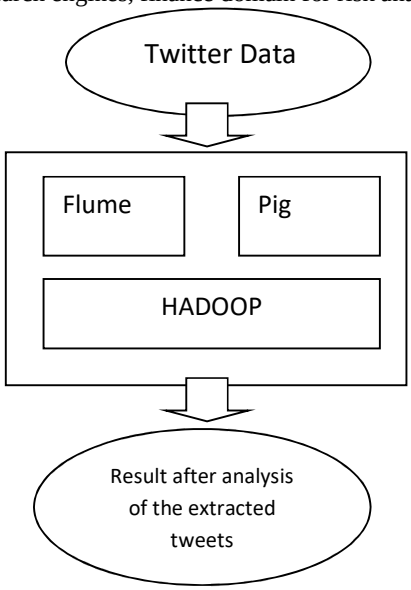

Fig. 1. Procedure to process the data

3.1.1 HDFS

A major part of Hadoop's ecosystem is its distributed file system which is HDFS. We use HDFS in our research that runs on commodity machines. It is highly fault tolerant and is designed for low-cost machines. HDFS has a high throughput access to application and is suitable for applications with large amount of data. Therefore, it is highly suitable for us as we are working with large amount of raw twitter data. HDFS has a master and slave architecture which has a single name node.This name node regulates the file system access. Data nodes handle read and write requests from the file

system's clients. They also perform block creation, deletion and replication of data upon instruction from the name node. Replication of data in the file system increases the data integrity and the robustness of the system.

3.1.2 Apache Flume
We use Apache Flume which is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of streaming data into the Hadoop Distributed File System (HDFS). It can be used for dumping twitter data in Hadoop HDFS.
3.1.3 Apache Pig
Then we proceed with Apache Pig which is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets.

3.1.4 Map-Reduce Algorithm
For our research, a faster real time processing can be obtained by using the clustered architecture set up by Hadoop. The program contains chained map-reduce algorithm which is used to process every tweet and assign the polarity of each sentiment to each remaining word of the tweet and then summing it up to decide final sentiment's polarity. Here, special care should be taken for the phrasal sentenceswhere sentiment of phrase matters rather than sentiment of each word.

3.1.5 Tokenization
In the tokenization process, all the words in a tweet are broken down into tokens. For example, "Jack that is an awesome car!" is broken down into individual tokens such as "Jack", "That", "is", "an", "awesome", "car". Emoticons, abbreviations, hash tags and URLs are recognized as individual tokens. Each word in a tweet is separated by a space. Therefore, on encountering a space, a token is identified.

3.1.6 Classification
Finally, the system will classify the processed data into Positive and Negative reviews with the help of AFINN dictionary and Piglatin language of Pig tool.

4. RESEARCH WORK AND ANALYSIS

So far, we have given the basic introduction and the basic methodologies required for our research. In this section, we will understand the process and steps required for our research to find out the views of different people on demonetization by analyzing their tweets from Twitter.We begin by collecting real-time tweets from twitter using Flume tool which belongs to Hadoop's ecosystem. We had to configure Flume before it can be used to extract data from twitter. To configure it, we open the configuration (configuration) file of Flume and replace the application program interface keys with the keys generated from our twitter account (access tokens). Then, we insert the keywords accordingly based on which our tool will fetch data from twitter. Finally, we store data in HDFS.

Our next step is to make the use of Pig tool of Hadoop's ecosystem to process and analyze the extracted twitter data.A Pig relation is similar to a table in a relational database, where the tuples in the "bag" correspond to the rows in a table. In order to do this, first we invoke the pigs grunt shell to operate with pig tool. We then import the extracted tweets and the AFINN dictionary to PigStorage.

Then, we extract only the tweet id and tweet text and store these in a bag which is "extract". We will use the command in piglatin which is "extract = FOREACH tweets GENERATE $0 as id,$1 as tweet;" to do so. Once done, we will then check the schema for the bag and the dictionary. We have to pay close attention to it as it should be similar in nature. After that, in our next step, we use the pig command "tokens = FOREACH extract GENERATE id,tweet, FLATTEN(TOKENIZE(tweet)) as word;". With the help of this command, we will be invoking internal function tokenization which extracts each word from the tweet text and stores it in a different column under the same tweet id. Simultaneously, we store all of these information in a different bag which is "tokens" as we can see from the above code. After that, we use the code
"rating = join tokens by word left outer, afinn by word us ing 'replicated';" to perform left outer join by word to classify and assign rating to each word of the tweet.

Moreover, we group the bag "wordrating" by id and tweet textin a new bag and we name it wordgroup and we use the command
"wordgroup = group wordrating by (id,text);" to do so. Then, foreach value in wordgroup named bag we use average function to perform average, and put them in a separate bag. In order to make this happen, we use the command
"avgrate = foreach wordgroup generate AVG(wordrating .rateing) as tweetrate;". After doing so, we may use the "dump" command to see the contents of the bag.Then, we will proceed to our next step which is to filter the negative and positive values and store them in separate bags. We perform this using the command "positivetweets = filter avgrate by tweetrate>0;"
which will store all the ratings of positive tweets in positivetweets bag and command "negativetweets = filter avgrate by tweetrate<0"
which will store all the negative tweet ratings in the negativetweets bag. We can use the "dump" command along with the name of the bag to see the contents of the bag. When we use "dump negativetweets", it shows us all the tweets that have got negative reviews. When we

use “dump positivetweets”, it gives us all the tweets that have a positive rating. In our next step, we use the count function along with the group-all function for each bag (negativetweets and positivetweets) to count the number of total positive and negative values using the command “postweetsgroup = group positivetweets all;” and “negtweetsgroup = group negativetweets all;” first to group them and then we use the command “postweets_count = foreach postweetsgroup generate COUNT(positivetweets.rate)”.Similarly,we use “negtweets_count = foreach negtweetsgroup generate COUNT(negativetweets.rate)” to count the total number of values in the bag under the column name “rate”. We can finally use the “dump negtweets_count” and “dump postweets_count” commands to show the values in the bags which were 1372 for negative tweets and 2195 for positive tweets, respectively. Now, we compare the values of both of the bags to conclude.From the majority from our outcome, we can say that we have more positively rated tweets and less negatively rated tweets.

5. RESULTS AND CONCLUSION

Huge collection of unstructured or semi-structured data that are accumulated on the web can be effectively extracted and analyzed by using Opinion Mining and/or Sentiment Analysis. In our research, we collected a little over eight thousand (8000) real time tweets using Flume tool and stored it in HDFS. Using Pig tool and PigLatin language, we analyzed the tweets and calculated the polarity of each tweet. We finally aggregated them to conclude the following:

The number of negative tweets that we get after we check the contents of negtweets_count bag using the “dump” command is 1372. Similarly,the number of positive tweets that we get after we check the contents of postweets_count bag using the “dump” command is 2195.

Hence, Negative tweets=1372 and Positive tweets = 2195.

As 2195 is greater than (>) 1372, we can say that we have,

2195 – 1372 = 823.

823 number of more positive tweets than the number of negative tweets.

This shows us that the majority of polarity is on the positive direction and in favour of demonetization. Hence, we can finally conclude that majority of our collected tweets have a positive opinion on demonetization. This information can be very useful when trying to make business or political decisions that are directly or indirectly related to demonetization. When people have positive sentiments about a certain event like demonetization, then it means that the occurrence of such events must have positive impacton the majority of lives and majority feels positive about it. Hence, conclusions like the reoccurrence of such events may have a high probability.

REFERENCES

[1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In *Proceedings of the Workshop on Language in Social Media (LSM 2011)* (pp. 30-38).

[2] Ingle, A., Kante, A., Samak, S., & Kumari, A. (2015). Sentiment analysis of twitter data using hadoop. *International Journal of Engineering Research and General Science*, *3*(6).

[3] Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (Vol. 10, No. 2010, pp. 1320-1326).

[4] Mane, S. B., Sawant, Y., Kazi, S., & Shinde, V. (2014). Real time sentiment analysis of twitter data using hadoop. *IJCSIT) International Journal of Computer Science and Information Technologies*, *5*(3), 3098-3100.

[5] Ingle, A., Kante, A., Samak, S., & Kumari, A. (2015). Sentiment analysis of twitter data using hadoop. *International Journal of Engineering Research and General Science*, *3*(6).

[6] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, *2*(1–2), 1-135.

[7] Pang, B., Lee, L., & Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*(pp. 79-86). Association for Computational Linguistics.

[8] Yang, C., Lin, K. H. Y., & Chen, H. H. (2007, November). Emotion classification using web blog corpora. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'07)* (pp. 275-278). IEEE.

[9] Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *ACL*. The Association for Computer Linguistics.

[10] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. 2007. Emotion classification using web blog corpora. In *WI ’07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 275–278, Washington, DC, USA. IEEE Computer Society.

[11] Alec Go, Lei Huang, and Richa Bhayani. 2009. Twitter sentiment analysis. Final Projects from CS224N for Spring 2008/2009 at The Stanford Natural Language Processing Group.