International Journal of Pure and Applied Mathematics
Volume 118 No. 18 2018, 2883-2889
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
Special Issue

Map Reduce based Key-Word Extraction and Bagging Algorithm for Review Classification in e-Commerce Websites

Pradeepa. S[1]
Associate Professor,
School of Computing,
SASTRA Deemed University

Sri Ram. E[2]
Student CSE,
School of Computing,
SASTRA Deemed University.

Swathi. G[3]
Student CSE,
School of Computing,
SASTRA Deemed University.

Mail: [1]pradeepa.pradee@gmail.com [2]sriram.060698@gmail.com [3]swaathyg98@gmail.com

Abstract **-** Customer reviews play an important role in sentiment analysis and opinion mining. With the rapid increase in the growth of e-commerce, understanding the mindset of the customer is very critical. To understand the reach of a product we can perform sentiment analysis on the user-generated reviews that are available on the internet. Since an array of information is available on the internet we need an efficient method to extract the necessary information before actually devising a method for sentiment analysis. In this work, classification of Amazon user reviews has been done using the same Key-word extraction bagging algorithm and Relative classifier for classification. To classify huge sets of reviews, components of the Hadoop such as Map-Reduce, Hive, and Pig are used in a systematic manner. Term Frequency-Inverse Document Frequency is implemented as it is highly efficient in Key-Words extraction. This algorithm classifies huge sets of data under seconds.

Keywords: Sentiment Analysis, Key word Extraction, Bagging Algorithm, Hadoop MapReduce, Relative Classifier.

I.INTRODUCTION:

Internet advancements are growing exponentially and e-commerce is the next big thing. It is crucial for online retailers to keep up with the latest field advancements and changing user sentiments to gain the cutting edge on the modern highly competitive online market. More often than not online shoppers tend to rely on the opinions and suggestions of other fellow users when making purchase decisions. Most of the reviews are long and this makes it difficult for the user to interpret the other user's mindset and also does not provide clarity on the quality of the product. Our proposed methodology generates review analysis within seconds as we use Hadoop and it can handle really large datasets efficiently. Sentimental analysis helps customer visualize satisfaction while purchasing by simple summarization of these reviews into positive or negative-two broader classified classes. Feedbacks are mainly used for helping customers purchase online and for knowing current market trends about products which is helpful for developing market strategies by merchants. These reviews play a vital role in determining potential customer for the products as well as market trends for products. Since reviews are highly unstructured, machine learning approaches are applied including Naive Bayes and support vector machine algorithms by first taking inputs as unstructured product reviews, performs preprocessing, calculates polarity of reviews, extracts features onto which comments are made and also plots graph for the result.. We

review Amazon products using this method to prove the efficiency in an effective manner.

II.RELATED WORK:

Many researchers have worked in the field of sentiment analysis, each one proposing new way of getting better efficiency from machine learning approaches. Fang, X. & Zhan[1] proposed means to handle the problem of sentiment polarity categorisation. In this method first steps were followed to identify the negative comments and then a mathematical approach was proposed for sentiment score calculation. And finally a feature vector method was adopted for the sentiment polarity categorisation. But this method failed to address a lot of critical issues. Firstly, this algorithm could not classify the comments based on star based reviews and also the algorithm proposed by them relies completely on sentiment tokens. Hence this method does not work that effectively for comments that contain implicit innuendo.

Lianghao[2] proposed an active learning method based on novel multi-domain. This framework selects text data from all categories such as books, DVDs, and electronics and from amazon.com. The data set used by this framework is the Multi-Domain Sentiment Dataset. During data preprocessing, they converted all words in upper case to lower case and removed the English stop words from the data. The authors of this paper defined the term frequency for weighting features. They have used joint query instances using a hierarchical structure among domains. In this paper, authors presented a framework in the linearly-separable manner and leave the non-separable case to our future work Manvee Chauhan and Divakar Yadav[3] went one step ahead and examined the effectiveness of applying machine learning techniques to sentiment classification problem. This method analysed and predicted product-based reviews using Naive Bayes and SVM. It was studied that the Naive Bayes produced an accuracy of 84.02% whereas the Support Vector Machine produced an accuracy of just 80.2%. Hence, concluding that the Naive Bayes is an efficient method. However large text files took quite a while for processing in this work. Kuat Yessenov[4] et al. presented an empirical study of efficacy of machine learning techniques in classifying text messages by semantic meaning. They have proposed numerous approaches for extracting text features such as bag-of-words model, using large movie reviews corpus, strictly adjectives and adverbs, handling negations, and using WordNet synonyms knowledge. They have evaluated their effect on accuracy of four machine learning methods - Naive Bayes, Decision Trees, Maximum-Entropy, and K-Means clustering. This was implemented using Python because all four algorithms are available in NLTK format. Taking into consideration the different methods proposed for sentiment analysis and analysing the critical necessity to handle large databases we have implemented the concept of opinion mining using Hadoop. The various modules used and the steps involved are discussed.

III.PROPOSED MODELING:

A. Dataset

The corpus used for our analysis is a dataset containing about 50,000 reviews of products belonging to the same category collected from Amazon. All these reviews are unstructured and not processed.

B. Module Description:

The modules used in the modelling are given below:

a) Reviews purification using Python
b) TF-IDF implementation using JAVA MapReduce
c) Pre-Processing of the given data using HIVE
d) Relative Classification using PIG UDF

This is also shown in Fig.2 as how each module interacts with each other to classify the data. The following subsections briefly explain each module.

C. Reviews purification using Python

Once the data set is obtained from Amazon, it is purified to extract the required semantics. We concentrate on the words that best express user's opinions such as good, bad, not good and not bad. First, we pick products that have almost 50,000 reviews and belong to the same category. The purification and filtering of this dataset can be done easily using the nltk package of Python. The product reviews are first divided into tokens and then the unwanted stop words are eliminated. The tokens thus obtained are then purified to obtain the required semantic words like good, bad, etc. We then proceed to move this purified dataset into the Hadoop Distributed File System of understanding the data. Thus we move onto the second module.

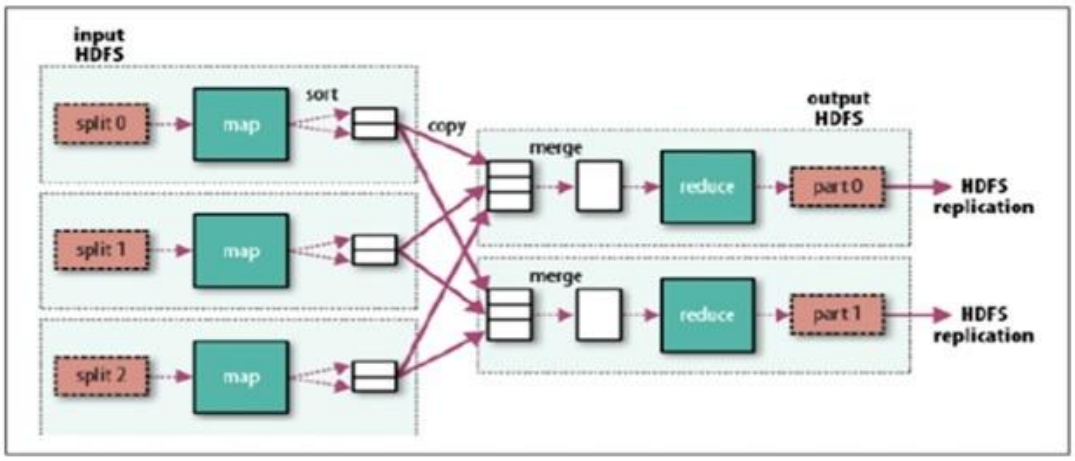

Fig 1: Architecture of HDFS

D. TF-IDF implementation using JAVA MapReduce:

Fig.1 describes the architecture of the Hadoop File System. The input data is split in a suitable way by the mapper and given to multiple workers. The output files from those workers are then reduced by the reducer to get the desired output for the given input data set. The input of this module is the purified dataset. We shall now work on understanding the data. The Term Frequency-Inverse Document Frequency (TF-IDF) bagging algorithm gives the positive and negative content available in the corpus. We need to get an understanding of the data on the product. Once all the words have been subject to this process, we need to compute statistical values for both positive and negative reviews which will help us in identifying the number of positive and negative content that is present in the reviews.

The general TF-IDF implementation will be as follows:

- Term Frequency (TF):-

$$\frac{Number\ of\ times\ (n)\ term\ t\ appears\ in\ a\ \ comment}{Total\ number\ of\ terms\ (N)\ in\ the\ comment}$$

- Inverse Document Frequency (IDF) :-

$$\log\left(\frac{Total\ number\ of\ comments}{Number\ of\ comments\ with\ term\ t\ in\ it}\right)$$

- TF-IDF = TF * IDF

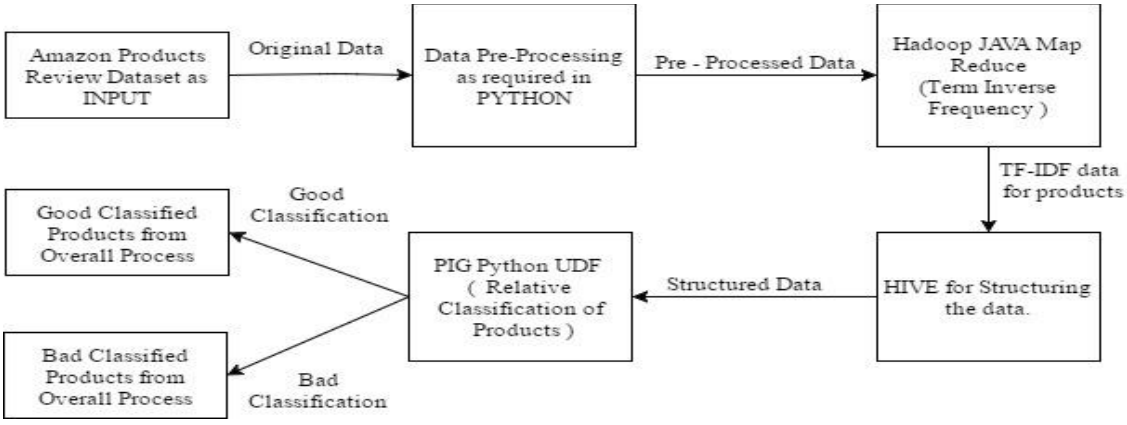

Fig 2: The method of classifying data

This TF-IDF is effectively implemented in the Hadoop environment using the MapReduce in JAVA language. MapReduce using <key, value> pairs involves 4 phases as follows:

- Phase One:

The purified raw data is fed as an input to the mapper. The data is sent as <key, value> pairs t both reducer. The role of the reducer is to compute the total counts of semantics in the data set and produce an output of the form:

<(word, product name, Total Comments Count),n>

Here, n is the count of the word for the respective product.

- Phase Two:

The output for the first phase is taken as input and likewise <key, value> pairs are sent to the reducer. The Term Frequency pairs are calculated and output is of the form:

<(product name, word, Tot Count),(N, TF)>

- Phase Three:

The output from the previous phase is taken as input and the IDF is calculated to be sent to the reducer. In the reducer the multiplication of the TF and IDF produces a product of the form:

*<(product name, word)(TF,IDF,TF*IDF)>*

- Phase Four:

In this phase, we preprocess the data as required using the mapper and reducer units. The pre-processed data is used to reduce repetition in the names of products. This finally leaves us with the output:

<(product name),(positive TF-IDF, negative TF-IDF)>

E. Pre-Processing of the given data using HIVE:

The HIVE component in Hadoop is used to structure the data. It is in this module that the unstructured data is converted into the structured table and all necessary SQL operations are performed on that table.

The steps involved are as follows:

- The output of the second module is first moved into the Hadoop Distributed File System.
- In Hive, we create an external table to which the data to be structured is loaded from the HDFS directory.
- Next, we move onto creating an Oracle table in HIVE and it is mapped to the previously created external table because in HIVE data cannot be directly moved into the Oracle table.
- The Oracle table is further preprocessed using SQL queries.

F. Relative Classification using PIG UDF:

This is the most important module and it is here that we use a relative classifier to separate and classify the products. In this relative classifier if the positivity of the TF-IDF value for a product is higher then, the product is said to be a good product and similarly, if the negativity is high they get into the bad classification. In some cases where both the values are same, the decision can be made using

$$LDT_p = \log\left(\frac{Pos_p + 0.01}{Neg_p + 0.01}\right)$$

Where,

LDT_p = Logarithmic differential TF-IDF

Pos_p = Positivity of a product p

Neg_p = Negativity of a product p

Now, based on the LDT_p the polarity of the product whether it has to be classified as a positive or negative is decided. This can be done as follows.

$$Pol_p = \begin{cases} positive \text{ if } LDT_p = 0 \\ negative \text{ if } LDT_p = -1 \end{cases}$$

This classification is done using PIGLatin language in the PIG component of the Hadoop. However, the User Defined Function(UDF) is written in Python language in which the classifier is implemented.

In the end, this model classifies the products into a good category or bad category. We create 2 separate files, one for good and the other for bad. The respective products are moved into the corresponding category to which they have been classified.

A pictorial representation such as bar charts and pie charts can be used to indicate the various reviews of each product and easily analyse the user's mindset.

IV.EXPERIMENTAL RESULTS:

This paper tackles the fundamental problem in sentimental analysis. Huge sets of reviews can be classified using Hadoop components such as MapReduce, Hive, and Pig. We implement the Term Frequency-Inverse Document Frequency is obtained using Hadoop MapReduce. Pre-processing involving removing outliners is done using HIVE component in Hadoop. The data obtained from HIVE is in a structured form. We then use PIG to relatively classify the products based on their reviews whether they are good or bad.

A. Hadoop MapReduce:

We employ Hadoop MapReduce to process large amounts of data. This involves two important tasks namely Map and Reduce. Large sets of data are taken and broken down into Key-Value pairs or tuples. This is Mapping. Then for the Reduce step, the output from the Map task is taken and further simplified to form a simple set of tuples. The Mapper is followed by the Reducer.

B. Apache HIVE:

This is a data warehousing software and facilitates us to read, write and manage datasets of

huge volumes using SQL. All that the user needs to connect to the HIVE is a command line tool and a JDBC driver. HIVE supports ACID transactions. And this is most efficient to analyse static data that is relative. Here the response time is very less.

C. Apache PIG:

In this platform, we use a high-level language to manage large sets of data and to analyse it. We use PigLatin to program in this platform and this is very effective as it is textual. PigLatin comes with a lot of advantages namely; it is very extensive, provides better optimisation, and allows simple programming.

The code was used to estimate the satisfaction level of users and the good, bad reviews for each product were found out as shown in Fig. 3. Table 1 shows the time taken for the mapper reducer approach.

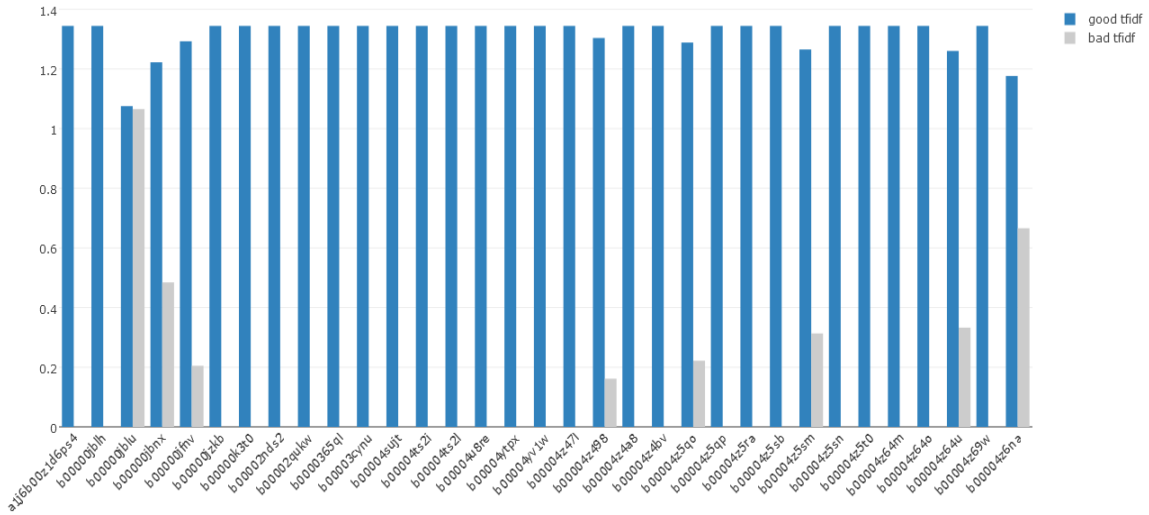

Fig 3. Bar Graph representation of good and bad for each product

Fig.3 shows the graphical representation of review for each product. It shows the exact classification of user reviews into good or bad opinion by using TF-IDF. Fig. 4 shows the time taken for each phase of activity to be performed. As the number of reviews increase, the time taken does not rise considerably due to the distributed nature of mapper reducer in the HDFS. The time taken is almost constant after a certain point which is also evident in Table-1.

TABLE-1.Running Time Analysis of the module.

Number of Reviews	Time for Phase 1 (ms)		Time for Phase 2 (ms)		Time for Phase 3 (ms)		Total Time
	Mapper	Reducer	Mapper	Reducer	Mapper	Reducer	Mapper + Reducer
1,000	3073	3171	2975	3329	2905	3224	18677
2,000	3205	4368	3692	3245	3351	3506	21367
3,000	3363	3762	3399	3308	3040	3356	20228
4,000	3338	3283	3200	3462	3373	3410	20066

Table 1. Time Analysis of TF-IDF module with different data sets

V. CONCLUSION

With the parallel growth in technology and research day by day, we come up with newer methods of performing sentiment analysis. Our model helps classify the products about whether they are good or bad. Since we are working in a Hadoop environment, all available documents can be reviewed with seconds in the most efficient manner available. However, we need to make more advancements in the field and come up with a more efficient and modern software that can identify and deal with sarcastic comments and also be one step forward, taking into account comparative opinions of the users. This page tackles a fundamental problem of sentimental analysis where you need not worry about giving unstructured data as input and you will still end up getting your desired output which either belongs to the good category or the bad category. To enhance the proposed methodology more segregations in the type of products can be included.

REFERENCES

[1] Fang, X., & Zhan, J. (2015). Sentiment analysis using product review data. Journal of Big Data.

[2] Li, Lianghao, Multi-domain active learning for text classification. Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2012.

[3] Manvee Chauhan & Divakar Yadav ,Sentimental Analysis of Product Based Reviews Using Machine Learning Approaches. Journal of Network Communications and Emerging Technologies (JNCET) Volume 5, Special Issue 2, December (2015)

[4] Kuat Yessenov and Sasa Misailovic, Sentiment Analysis of Movie Review Comments 6.863 Spring 2009 final project.

[5] Ghag, K., & Shah, K. (2014). SentiTFIDF – Sentiment Classification using Relative Term Frequency Inverse Document Frequency. International Journal of Advanced Computer Science and Applications.

[6] Saif H., He Y., Alani H. (2012) Semantic Sentiment Analysis of Twitter. In: Cudré-Mauroux P. et al. (eds) The Semantic Web – ISWC 2012. ISWC 2012. Lecture Notes in Computer Science, vol 7649. Springer, Berlin, Heidelberg.

[7] Bo Pangand Lillian Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval Vol. 2, No 1-2 (2008) 1–135.

[8] Sarvabhotla K, Pingali P, Varma V (2011) Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents. Inf Retrieval14(3): 337–353.