International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 11 104 – 108

Sentiment Analysis for Stock and Share Marketing

Shubham Jain
Student, Department of Information Technology, Maharaja Agrasen Institute of Technology
Delhi, India
shjain6670@gmail.com

Mark Kain
Student, Department of Information Technology, Maharaja Agrasen Institute of Technology
Delhi, India
markkain24@gmail.com

Neha Singh
Assistant Professor, Department of Information Technology, Maharaja Agrasen Institute of Technology
Delhi, India
join7neha@gmail.com

Abstract **-** Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. sentiment analysis aims to determine the attitude of a speaker, writer, or other subject with respect to some topic or the overall contextual polarity or emotional reaction to a document, interaction, or event. This paper will showcase how to perform sentimental analysis on stock/share market twitter data using pig. To begin with, we will be collecting real time tweets from twitter using flume. With the help of AFINN dictionary, we can find positive sentiments from the data which we have dumped into our HDFS/FLUME.

Keywords **:** *Sentiment Analysis , Natural Language Processing , Text Analysis , Computational Linguistics , Biometrics , Stock/share market , HDFS, Flume, AFFIN.*

*****

I. INTRODUCTION

Sentiment analysis is the process of determining the emotional tone behind a series of words, used to gain an understanding of the the attitudes, opinions and emotions expressed within an online mention.. Generally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall functionality of a document. In recent years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind Sentiment Analysis today. The Web is a huge repository of structured and unstructured data. The analysis of this data to extract latent public opinion and sentiment is a challenging task.

Sentiment Analysis identifies the phrases in a text that bears some sentiment. It identifies the polarity and degree of the sentiment. Sentiments are classified as *objective* (facts), *positive* (denotes a state of happiness, bliss or satisfaction on part of the writer) or *negative* (denotes a state of sorrow, dejection or disappointment on part of the writer).

The sentiments can further be given a score based on their *degree* of positivity, negativity or objectivity.

Sentiment analysis has aroused the interest of many researchers in recent years, since subjective texts are useful for many applications. Studies on sentiment analysis mainly focus on framework and lexicon construction, feature extraction, and polarity determination. Zhu Nanli presents a survey on the latest development in sentiment analysis, and makes an in-depth introduction of its research and application in business and Blogsphere. The methods used in current research are especially emphasized and the existing problems of those studies are discussed. Finally, some possible future directions of research are pointed out[5].

Sentiment Analysis has many applications in various fields. According to Ramteke the application from a user's standpoint is the application related to review websites. Tools that help summarize the sentiment regarding a product or service help users in identifying their product of choice. Similarly, vendors build tools that analyze customer feedback which help improve user experience. The future might see applications wherein a system gauges the human emotion through sensory means and then creates an environment that helps improve the human life in general. This section describes a few of these applications that have been built or are possibilities in the near future[1].

Ramteke et al. used domain ontology to handle thwarting. Domain ontology comprises of features and entities from the domain and the relationships between them depicted by a hierarchy. For building the ontology the features and entities need to be identified and then linked in the form of an hierarchy. The authors have built the domain ontology manually[1].

Balahur et al presented a comparison on the techniques as well as resources which may be utilized for mining opinions

from quotations in news articles. The challenges in the task were shown, motivated by the possibility of various targets as well as a huge set of affect phenomena which quotes comprise. The proposed methods evaluated were evaluated utilizing annotated quotations taken from news given by the EMM news collecting engine. A general OM system needs usage of both huge lexicons as well as specialized training as well as testing data[3].

Veeraselvi and Saranya presented opinion detection and organization subsystem, which had already been integrated into proposed larger question-answering system. The subjectivity classification system used Genetic-Based Machine Learning (GBML) technique that considered subjectivity as a semantic problem. The classification of a review was estimated through the average semantic orientation of phrases in the review which comprise adjectives or adverbs. Experimental results of the proposed techniques were efficient and generated eminent evaluations[4].

II. METHODOLOGY

Step 1

Register the downloaded jars in pig by using the below commands:

Grunt> REGISTER '/home/edureka/Desktop/elephant-bird-hadoop-compat- 4.1.jar';

Grunt> REGISTER '/home/edureka/Desktop/elephant-bird-pig-4.1.jar';

Grunt> REGISTER

'/home/edureka/Desktop/json-simple-1.1.1.jar';

Step 2

The tweets are in nested Json format and consists of map data types. We need to load the tweets using JsonLoader which supports maps, so we are using **elephant bird JsonLoader** to load the tweets.

Below is the first Pig statement required to load the tweets into Pig:-

load_tweets = LOAD '/user/flume/tweets/' USING com.twitter.elephantbird.pig.load.JsonLoader('-nestedLoad') AS myMap;

Step 3

Now, we shall extract the id and the tweet text from the above tweets. The Pig statement necessary to perform this is as shown below:

extract_details = FOREACH load_tweets GENERATE myMap#'id' as id,myMap#'text' as text;

We can see the extracted **id** and **tweet text** from the tweets in the below screen shot.

Figure 1 Dumping extract_details

Step 4

We have the tweet id and the tweet text in the relation named as **extract_details**.

Now, we shall extract the words from the text using the TOKENIZE key word in Pig.

tokens = foreach extract_details generate id,text, FLATTEN(TOKENIZE(text)) As word

From the below screen shot, we can see that the text got divided into words.

Figure 2 Dumping Tokens

Step 5

Now, we have to analyse the Sentiment for the tweet by using the words in the text. We will rate the word as per its meaning from +5 to -5 using the dictionary AFINN. The AFINN is a dictionary which consists of 2500 words which are rated from +5 to -5 depending on their meaning. You can download the dictionary from the above given link.

We will load the dictionary into pig by using the below statement:

dictionary = load '/AFINN.txt' using PigStorage('\t') AS(word:chararray,rating:int);

ISSN: 2321-8169
104 – 108

Figure 3 Loading AFFIN.txt

Step 6

Now, let's perform a map side join by joining the **tokens** statement and the dictionary contents using this command:

word_rating = join tokens by word left outer, dictionary by word using 'replicated';

Figure 4 Performing Join Operation on *tokens* and *dictionary*

In the above screenshot, we can see that the word_rating has joined the **tokens(**consists of id, tweet text, word**)** statement and the **dictionary**(consists of word, rating).

Step 7

Now we will extract the **id,tweet text** and **word rating(**from the dictionary**)** by using the below relation:

rating = foreach word_rating generate tokens::id as id,tokens::text as text, dictionary::rating as rate;

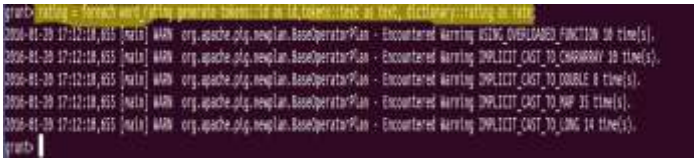

Figure 5 Extracting Rating of Each Word

Step 8

Now, we will group the **rating of all the words in a tweet** by using the below relation:

word_group = group rating by (id,text);

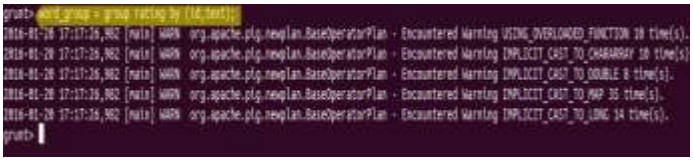

Figure 6 Grouping rating of all the words in a tweet

Step 9

Now, let's perform the **Average** operation on the **rating of the words per each tweet**.

avg_rate = foreach word_group generate group, AVG(rating.rate) as tweet_rating;

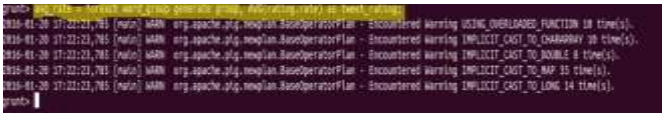

Figure 7 *Average* Operation on the Rating of the Words Per Tweet

Now we have calculated the Average rating of the tweet using the rating of the each word. You can refer to the below image for the same.

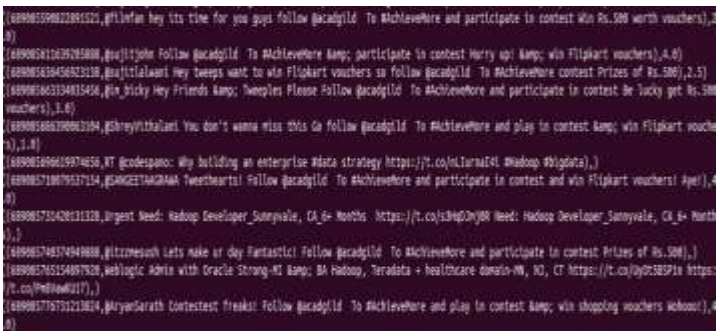

Figure 8 Dumping avg_rate

Step 10

From the above relation, we will get all the tweets i.e., both positive and negative. Now we will filter the positive tweets using the below statement:

positive_tweets = filter avg_rate by tweet_rating>=0;

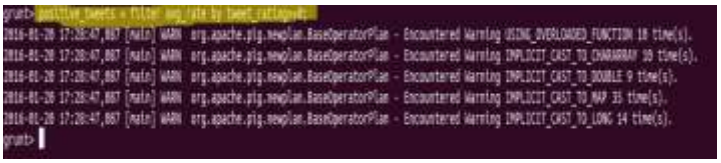

Figure 9 Filtering Positive Tweets

We can see the positive tweets and its rating in the below screen shot.

Figure 10 Dumping positive_tweets

III. RESULT

We can classify the positive tweets by taking the rating of the tweet which can be from **0-5.** We can classify the negative tweets by taking the rating of the tweet from **-5 to -1.**

We have now successfully performed the Sentiment Analysis on Twitter data using Pig.

IV. DISCUSSION/ANALYSIS

A. Application of Sentiment Analysis

- Applications to Review-related Websites

Today Internet has an entire gamut of reviews and feedbacks on almost everything. These include product reviews; feedbacks on political issues etc. Thus there is a need for a sentiment engine that can extract sentiments about a particular entity. It will provide a consolidated feedback or rating for the given topic. Such applications would not themselves contain any opinions, but they would fetch the opinionated text from various resources and provide an effective polarity. This would serve the need of both the users and the vendors.

Another application of Sentiment Analysis is in automatic summarization of user reviews. Automatic summarization is the creation of a summary of the entire review using an automated program. In case of user reviews, it is difficult for a new user to look at all the reviews thoroughly and understand what aspect of the product is not appreciated. Thus, there is a need of a summarizing application that will briefly inform the user about the polarity of the reviews, for example, thumbs up or thumbs down for the topic.

It is assumed that all user ratings are accurate. However, there are cases where users have accidentally selected a low rating when their review indicates a positive evaluation, or vice versa. Moreover, there is some evidence that the user ratings can be biased, based on a previous experience or otherwise in need of correction. Automated sentiment classifiers can help us correct such cases by identifying sentiments corresponding to the relevant features of the product.

- Applications *in Business Intelligence*

It has been observed that more and more people nowadays tend to look upon reviews of products online before buying them. And for many businesses the online opinion can make or break their product. Thus, Sentiment Analysis finds an important role in businesses. Businesses wish to understand the online reviews in order to improve their products and in turn their reputation.

- Applications *in smart homes*

Smart homes are supposed to be the technology of the future. It is speculated by leading scientists in all fields that eventually the entire homes would be networked and people would be able to control any part of the home using a tablet device. In such homes, Sentiment Analysis would also find its place. Based on the current sentiment or emotion of the user, the home could alter its ambiance to create a soothing and peaceful environment.

B. Challenges

Sentiment Analysis is a very challenging task. It requires deep understanding of the problem. We discuss some of the challenges faced in Sentiment Analysis.

• Identifying subjective portions of text: The same word can be treated as subjective in one context, while it might be objective in some other. This makes it difficult to identify the subjective (sentiment-bearing) portions of text. For example:

– The language of the author was very *crude*.

– *Crude* oil is extracted from the sea beds.

The same word "crude" is used as an opinion in first sentence, while it is completely objective in the second sentence.

• Associating sentiment with specific keywords: Many sentences indicate an extremely strong opinion, but it is difficult to pinpoint the source of these sentiments. Hence an association to a keyword or phrase is extremely difficult. For example:

– Every time I read 'Pride and Prejudice' I want to dig her up and beat her over the skull with her own shin-bone.

In this example, "her" refers to the character in the book "Pride and Prejudice", which is not explicitly mentioned. In such cases the negative sentiment must be associated with the character in the book.

• Domain dependence: The same sentence or phrase can have different meanings in different domains. The word unpredictable is positive in the domain of movies, but if the same word is used in the context of a vehicle's steering, then it has a negative connotation.

• Sarcasm Detection: Sarcastic sentences express negative opinion about a target using positive words. For example:

– Nice perfume. You must marinate in it.

The sentence contains only positive words but still the sentence expresses a negative sentiment.

V. CONCLUSION

Sentiment analysis has become an important factor in decision making process in a particular field. In this paper we discussed techniques for pre-processing and information retrieval of tweets through twitter. We also discussed how through Pig we can tokenize each tweet into individual words, assign each word rating, calculate average rating for a tweet and then, filter all the positive tweets from the data. Also, we analyzed various prospective, features, applications and challenges to sentiment analysis.

ACKNOWLEDGEMENT

First and foremost, I wish to express my profound gratitude to **Ms. Neha Singh (Assistant Professor)** for giving me the opportunity to carry out the project . My heartfelt thanks for her invaluable guidance, immense help, support & useful suggestions throughout the course of my project work.

Last but not least I thank the Almighty for enlightening me with his blessings .

REFERENCES

[1] A. Ramteke, A. Malu, P. Bhattacharyya, J. S. Nath, “Detecting Turnarounds in Sentiment Analysis: Thwarting”, Association for Computational Linguistics, vol. 2, pg. 860-865, 2012.

[2] S. Mukherjee, P. Bhattacharyya, “Sentiment Analysis in Twitter with Lightweight Discourse Analysis”, International Conference on Computational Linguistics, pg. 1847-1864, 2012.

[3] M. Wiegand, A. Balahur, B. Roth, D. Klakow, A. Montoyo, “A feature dependent method for opinion mining and classification”, Association for Computational Linguistics, vol. 2, pg. 1-7, 2008.

[4] S. J. Veeraselvi, C. Saranya, “Semantic orientation approach for sentiment classification”, Green Computing Communication and Electrical Engineering (ICGCCEE), 2014.

[5] Z. Nanli, Z. Ping, Li Weiguo, “Sentiment analysis: A literature review”, Management of Technology (ISMOT), 2012