International Journal of Engineering &Technology, 7 (3.12) (2018) 434-436

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Sentiment Analysis of Tweets Using Hadoop

Pranav Seth[1], Apoorv Sharma[2], R. Vidhya[3]

Department of Computer Science Engineering, SRM Institute of Science and Technology
SRM Nagar, Kattankulathur, Kancheepuram District, Tamil Nadu
**Corresponding Author Email: [1]pranav_sanjeev@srmuniv.edu.in*
[2]apoorv_sharma@srmuniv.edu.in, [3]vidhya.r@ktr.srmuniv.ac.in

Abstract

Blogging and networking platforms like Facebook, Reddit, Twitter and LinkedIn are social media channels where users can share their thoughts and opinions. Since online chatter is a vital and exhaustive source of information, these thoughts and opinions hold the key to the success of any endeavour. Tweets which are posted by millions all over the world can be used to analyse consumers' opinions about individual products, services and campaigns. These tweets have proven to be a valuable source of information in the recent years, playing key roles in success of brands, businesses and politicians. We have tackled Sentiment Analysis with a lexicon-based approach for extracting positive, negative, and neutral tweets by using part-of-speech tagging from natural language processing. The approach manifests in the design of a software toolkit that facilitates the sentiment analysis. We collect dataset, i.e. the tweets are fetched from Twitter and text mining techniques like tokenization are executed to use it for building classifier that is able to predict sentiments for each tweet.

Keywords: *Sentiment Analysis, tokens, tweets, Hadoop, emotion.*

1. Introduction

With increase in penetration of social media in everyone's daily lives, people can share their opinions on social networking websites like Facebook, Twitter, etc. People freely share their unbiased opinions on these platforms and thus, an opportunity has arisen where these free thoughts can be utilised to cater to the needs of the general public. Sentimental Analysis is the process of determining the emotion of any form of expression. Sentiment Analysis is used to gain an understanding of the attitude of the targeted audience using computational methods such as computational linguistics, data mining, database management, text mining, natural language processing, neural networks and statistics. Data is collected from various sources of information and then cleaned and structured for Sentiment Analysis to take place. Sentimental Analysis is often used to gauge the reception of a new product or campaign in the market and to know the general consensus behind the product or experience being considered. By having the knowledge of the general opinion of the public and effectively the reasoning behind an online mention, one can use this knowledge to their advantage in order to improve the product or experience. However, one faces many problems while performing Sentiment Analysis. Named Entity Recognition, Informal and Improper Language, Slang, Sarcasm, Anaphora Resolution are some of the issues faced while performing Sentiment Analysis. Sentiment Analysis can help in improving user experience by analysing negative emotions. It can generate additional revenue streams for a business by determining market for a new product/experience which will improve the emotion of the tweets by the target audience.

In order to improve accuracy and precision, we first need to answer a simple question; what are the pre-determined problems It can improve customer service and thus, determine the success of the product or experience in question. This feedback loop results in corrective production and design, leading to continuous improvement of product or experience.

Our approach to tackle the problem of analysing each tweet is to perform Sentiment Analysis by tokenisation of each tweet (user opinion on twitter). We will split each tweet into its building blocks called tokens and then compare these tokens with dictionaries of positive and negative words. By comparing each token with these dictionaries, we will allot a positive, negative or neutral score for each token to the tweet. We will then calculate the overall weight of each tweet by addition of all the score of a tweet, thus obtaining the overall emotion of the tweet. By using this approach, we can have a fairly accurate idea of the overall emotion of the user and then analysis can be performed to take the input of the user for the next iteration of the product/experience. Using this approach can deliver fairly accurate results as dictionaries of positive and negative tokens are fairly exhaustive in nature. This will allow us to take all the tokens of a tweet into account while performing Sentiment Analysis.

2. Analysis

An exhaustive research has started taking place in the field of Sentiment Analysis after realisation of the opportunities offered by the analysis of user reviews and opinions. This research is largely centred on arriving to a conclusion from any user review or opinion. However, Sentiment Analysis of tweets has proven to be a difficult task.

when performing Sentiment Analysis? As cited by Ghag and others [1], following are some of the most formidable challenges faced while performing Sentiment Analysis;

- Hidden Sentiment Identification
- Named Entity Recognition
- Informal and Improper Language
- Anaphora Resolution
- Dealing with Big Data being some of the others.

Hidden Sentiment Identification is defined as the science behind realising the actual intention behind a tweet as a tweet may not necessarily be restricted to a singular emotion. Hidden Sentiment Identification continues to be a challenge as discussed in [2].

Named Entity Recognition procures information from unstructured text and rearranges it into groups. It answers questions like, for example, Is 300 Spartans a group of Greeks or a movie? What is the person trying to express in this tweet? Which meaning should be considered in this particular context? How can we infer the subjective emotion behind similar phrases? Similarly, if there's a mention of "New Delhi" in a tweet, computing Named Entity Recognition would predict that as "Location".

Informal Language is widespread on social networks. It includes Elision and Assimilation. They are common when encountering informal ways of communication online. Elision of sounds is common in formal and informal language in abbreviated forms like isn't, I'll, who's, they'd, haven't and so on. It is very necessary to remember that sounds do not simply vanish. Some examples of Elision in informal language are 'em, sorta, coulda, woulda, shoulda and many more. Meanwhile, Assimilation is commonly found in languages all over the world by which one sound becomes similar to a nearby sound. This can occur either inside the structure of a word or between more than one words, for example, gotta, s'pposed. Complete assimilation takes place when the assimilated words' sound is a replica of the sound that it assimilates to. Partial assimilation is very common and very difficult to differentiate from others. When it occurs, the assimilated word creates the same sound, but disguised to be more similar, phonetically, to another word. Intermediate Assimilation is common in many languages, including English. It just means that the assimilated words' sound changes in some regards to become a different sound, but the word's sounds doesn't become completely the same as the sound it assimilates to. This kind of assimilation occurs in contractions and with the common word suffixes -s and -ed, and it is mandatory.

Anaphora Resolution is the resolving of problem where the parser cannot decide what a pronoun refers to. In the example mentioned below, both statements are utterances; and together, they form a discourse. Vidhi helped Vijay; she was helpful. As human, readers and listeners can quickly and unconsciously work out that the pronoun "she" refers to "Vidhi" in the former statement. Question that arises is, how do we reconstruct this into something meaningful? The tweets collected from Twitter can range from hundreds to thousands in number. Thus, we need a HDFS distributed over a large number of nodes to ensure that parallel processing power enables voluminous amounts of data to be processed.

The aforementioned problems need to be converted into the corresponding formal counter parts and the records in the database modified accordingly to reflect the changes in the final scores. As discussed in [3], we propose an approach that classifies tweets into 3 different sentiments. The approach proposed is scalable and can be run to classify tweets into more sentiments given enough training data exists.

3. Design

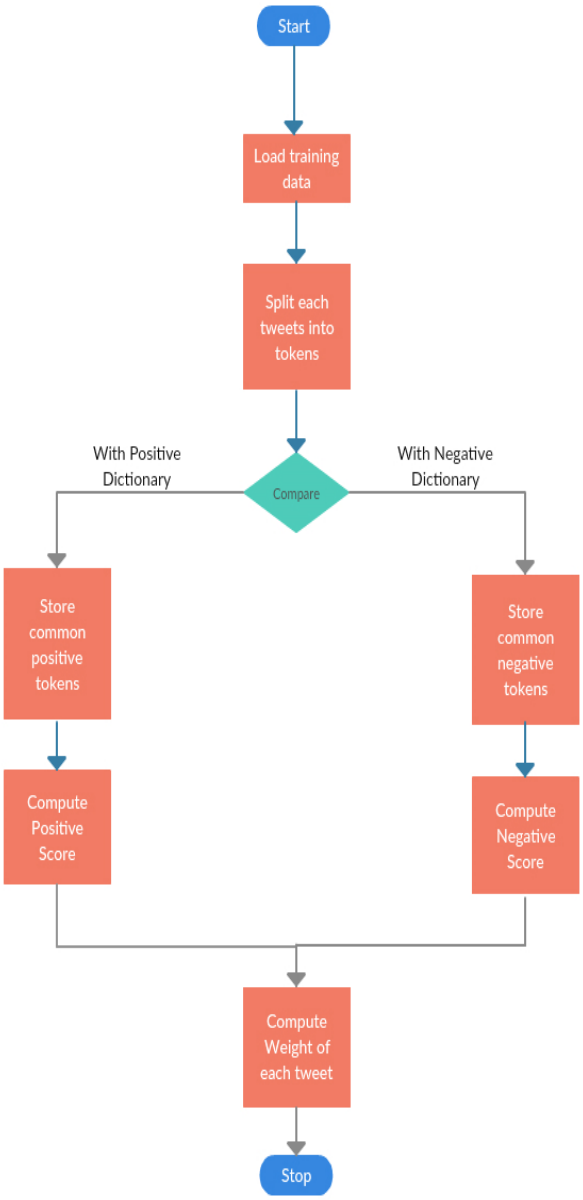

Fig. 1: Mind map of our proposed solution

Figure 1 shows a mind map of how data will flow in our proposed solution. We will use the Hadoop framework for the successful implementation of each step of the process. Our training dataset consists of about five hundred tweets from different users on Twitter on various topics of discussion in .csv (Comma Separated Values) format. The term "Classifier" refers to a mathematical function, implemented by a classification algorithm that organises input data to a category. Text mining is the scientific process of analysis of structured data, preferably in natural language. After the training data has been stored, we will perform tokenisation of each tweets and store the tokens. These tokens will then be compared twice; once with a lexicon of positive sentiments and then again with a lexicon of negative sentiments. The matches in both the comparisons will be stored in two separate tables and a positive or negative score will be given for each match. We will then compute the weight by summation of the positive and negative scores for each tweet. This will give us a weight of the tweet which will help us identify whether the tweet is positive, negative or neutral in nature.

4. Development

We will load the training data i.e. tweets into the Hadoop File Distributed System (HDFS) [4] using terminal in Cloudera distribution of Hadoop. This training data consists of the name of the user who has posted the tweet, the time and date the tweet was posted and many other details which are not relevant to this discussion. HDFS will only be used for storing the training data so that the data is centrally located and available to all workstations in case of scalable operations. The next phase requires the use of another Hadoop framework called Hive [5]. Hive allows operations on data stored in HDFS in the form of tables from any node with the help of a terminal and SQL based commands. We will proceed to create another table where we will store only the pertinent data i.e. tweets. After the tweets have been stored in a separate table, we will focus on splitting each tweet into individual words i.e. tokens. Tokenisation is the process of splitting sentences into words. We will perform tokenisation of these tweets using Hive framework provided in Hadoop.

5. Implementation

We will use the explode() function to perform tokenisation of tweets and store the tokens in another table. Then we will compare tokens with lexicons of positive and negative sentiments. We will create table for both the comparisons and store the tweet, and the common word in the training data and dictionary by using Left Join SQL command. We will then combine these two tables, thus obtaining a total weight from the positive and negative scores. This will allow us to classify the tweet into any of the three emotions, positive, netural or neutral. We will do this calculation by uploading the final table to HDFS alongwith a Java program in .jar format. We will then use Hive by terminal to execute the .jar and the result will be saved on HDFS. We can then download this file and see the result of the Sentiment Analysis.

6. Evaluation

After downloading the results of the Sentiment Analysis from HDFS, we can see the holistic polarity of each tweet, with the weight deciding the true sentiment behind the tweets. We can then observe for each tweet whether the result is positive, negative or neutral. This allows us to know the sentiment behind the tweet in question and its classification. This classification allows us to draw important observations regarding the product, campaign, experience or service in question. Are the customer's needs satisfied by the product? If so, how can we reproduce these needs in the next iteration of the product or experience? How can this breakdown of user's thoughts and opinions be beneficial to provide them with better services and experiences? We can summarise the answer to these questions for benefit for all the parties involved from the creator to the end user.

7. Conclusion

In this work, we proposed a distributed solution to process sentiment analysis for twitter using lexicon-based algorithm. Different components of the proposed solution have been discussed. We tested the proposed solution on two real data set. The results showed significant performance improvement in running time compared to the implementation of the lexicon-based algorithm in java.

In the future, we aim to test the code with larger datasets and look for some of the complex issues discussed earlier. We plan to include emoji as well during classification in future. This method of Sentiment Analysis can be extrapolated to other Indian languages like Hindi, Marathi, Tamil and their regional dialects as well.

References

[1] K. Ghag and K. Shah, "Comparative analysis of the techniques for sentiment analysis," in Proc. Int. Conf. Advances in Technology and Eng., pp. 1–7, Jan. 2013.

[2] M. Bouazizi and T. Ohtsuki, "Sentiment Analysis: from Binary to Multi-Class Classification - A Pattern-Based Approach for Multi-Class Sentiment Analysis in Twitter," in Proc. IEEE ICC, May 2016.

[3] M. Bouazizi and T. Ohtsuki, "Sentiment Analysis in Twitter: from Classification to Quantification of Sentiments within Tweets," in Proc. IEEE ICC, May 2016.

[4] https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[5] https://hive.apache.org/