Annals of Pure and Applied Mathematics
Vol. 7, No. 1, 2014, 65-70
ISSN: 2279-087X (P), 2279-0888(online)
Published on 9 September 2014
www.researchmathsci.org

Annals of
Pure and Applied Mathematics

A Model to Compute Degree of Polarity of Review Titles

***Sohom Ghosh*[1], *Santanu Modak*[2] and *Abhoy Chand Mondal*[2]**

[1]Department of Computer Science and Engineering, Heritage Institute of Technology
Kolkata – 700107, West Bengal, India
E-mail: sohom1ghosh@gmail.com

[2]Department of Computer Science, University of Burdwan
Burdwan–713104, West Bengal, India.
E-mail: modaksantanu@gmail.com; abhoy_mondal@yahoo.co.in

Received 12 July 2014; accepted 22 August 2014

Abstract. Review Polarity Computation has been a flourishing frontier in the Natural Language Processing community. In this paper, we thoroughly study review titles of electronic products and compute the sentiment scores. Firstly, we conduct our experiment by collecting the review titles from a popular e-commerce website to build our dataset. Our dataset contains more than 1000 positive and negative review titles. For preprocessing, several NLP operations like tokenization, stop-word removal, stemming and so on have been done on the dataset. We build our own unique word corpora separately for positive and negative words. Finally, we design a new innovative model which automatically generates the scores by analyzing the review title. The score vary from -5 to +5. A score of -5 indicates that the review title is extremely negative and that of +5 indicates that it is highly affirmative. Experimental results confirm the high efficiency of our model. A product can be rated automatically as soon as a user writes the title of the review. Thus, the company can decide which reviews to display in their front page just by analyzing the title of the review.

***Keywords*:** opinion mining, sentiment analysis, linguistics, natural language processing, polarity computation, ordinal regression

1. Introduction

Tasks of Sentiment Analysis are classified into two different categories. First one is, taking unstructured reviews from user, processing it by natural language processing techniques and classify that it is as positive or negative. Several researchers have also studied about neutral opinion as neutral opinion does not play any important role for decision making process. So detection of neutral opinion and eliminate that from dataset is also an important job. In this type of problem, sentiment analysis called as "Text Classification" problem. Another type of problem is also possible, where system can rate the product by processing reviews. A scale from 1 to 5 is defined and system detect the ratings, where 1 or 2 means that review is negative and 4 or 5 means review is positive. A score of 3 can be considered as neutral review. This type of problem is called "Regression" problem.

2. Basic terminologies

Tokenization

The art of extracting words from a sentence is called Tokenization. It is the act of splitting a corpus into its constituent words. Natural Language Toolkit, a package of Python 2.7 provides us with Punkt sentence tokenizer. But, we have to split all sentences presented in our dataset**.** So, in this project we use strip and split functions of Python for tokenizing. We eliminate the trailing blank spaces using strip() function. After that, we use the split() function to extract the individual words from the sentences. Finally, we store these words in a list.

Stop word removal: There are certain words in English which doesn't contribute to the meaning of sentences like 'is', 'am', 'the' and so on. Thus, there is no point keeping them in our corpus. So, we eliminate them using Python's NLTK package stopwords.words('english') . Now our corpus is free from stop words. Since we are reducing the corpus here, the time required for analyzing it will be sufficiently less. This contributes to the high efficiency of our model.

Stemming: There are certain word with structural affixes, e.g. 'produce', 'producing' and 'produced'. They all mean the same but their affixes are different. Here, we can easily guess that for the ease of processing, it will be better if we convert all of them into a single form. This is what is done by a stemmer. NLTK provides us with Porter stemmer, Snowball stemmer, and Regular Expression stemmer and so on. We use porter stemmer to design our model.

Corpus: Corpus is a collection of machine-readable text, which is much needed for Natural Language Processing Research. The corpus we used in this case contains more than six thousand words and symbols. We are not able to list the whole of it here. So, let's look at a sample we made from it:

Positive Symbols and Words: [[':) ', ':-)', '=)', '(:', '(-:', ':-D', ':D', ':d', ':-d', ':> ', ':->', ':-))', 'x-D', 'X-D', 'LOL', '(lol)', '(LOL)', '':'D'', 'accomplishment', 'achievement', 'motivating', 'moving', 'natural', 'simple', 'skilled', etc.

Negative Symbols and Words: '':'c'', '':'-C'', '':'-c'', 'T_T', ':-O', ':O', ':o', ':-o', '8-o', ':[', ':C', ':-C', '>-@', '>:0', '>-0', '>:-0', '‘>-o', '‘'abysmal', 'adverse', 'alarming', 'ignorant', 'ill', 'immature', 'unlucky', ', 'vindictive', 'wary' etc.

Polarity: Polarity helps us to detect positivity and negativity of review. Polarity of each sentiment word is calculated by our positive and negative word corpus. We assign +1 for every positive word and assign -1 to every negative word. Average polarity of all words presented in the sentence helps us to detect actual sentiment.

Regression: Regression Analysis is a statistical process which shows the relationship between Dependent Variable, say y and one or more independent variables, say x. If the unknown parameters, which represents vector, then Regression problem defined as

$E(Y \mid \mathbf{X}) = f(\mathbf{X}, \boldsymbol{\beta})$, which is an initial approximation. In General Binary Logistic Regression Model, the response variable has two levels, 1=success and 0= failure. But in this paper we tried to calculate polarity. So, we use Ordinal Regression, which is also called Ordinal Classification, to set a fixed, discrete rating scale. The ordinal outcome variable coded from -5 to +5 based on sentiment word present in the review title.

3. The model

Our model takes a text file containing the review title as input. We use this text file as our dataset. Firstly, we preprocess it by traditional Natural Language Processing techniques. Our preprocessing steps include Tokenization, Stop word removal and Stemming. We do the same for our positive and negative word corpora. After that, we compare each word of the processed dataset with our word corpora and generate the score. Finally, we use our own technique to scale the score from-5to+5.

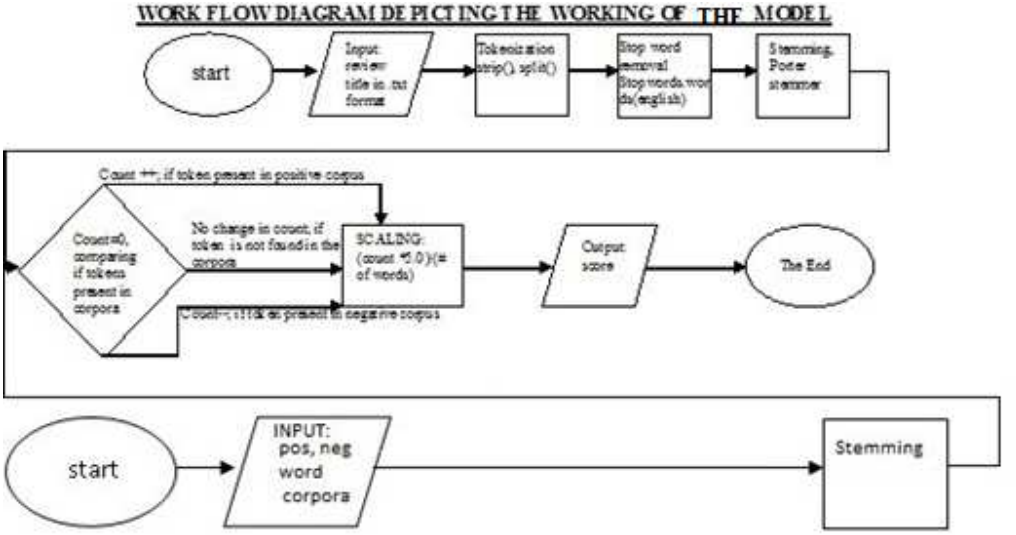

Figure 1:

4. Experiments and results

(Note: - For Better Experimental Results, in these examples we have omitted the scaling process. The scaling process works well if review title are ideal i.e. contains 1-5 words)

CASE I:

Dataset: Samsung Galaxy S Duos 2 Loses The Game With Slow Apps And Bad Battery

Dataset after preprocessing: Samsung Galaxi S Duo 2 Lose Game Slow App Bad Batteri

Score: -3

CASE II:

Dataset:hello guys this is the best tablet ever thnk u flipkart

Dataset after preprocessing:hello guy best tablet ever thnk u flipkart

Score: 4

CASE III:

Dataset:Samsung Galaxy S Duos 2: A Huge FailureS

Dataset after preprocessing: Samsung Galaxi S Duo 2: A Huge Failur

Score: -1

CASE IV:

Dataset: A good phone under 10k but strech your budget little & go for Moto G

Dataset after preprocessing: A good phone 10k strech budget littl & go Moto G

Score: 0

CASE V:

Dataset:A Good Budget Android Dual SIM Phone by Samsung.

Dataset after preprocessing: Good Budget Android Dual SIM Phone Samsung.

Score: 1

5. Analysis

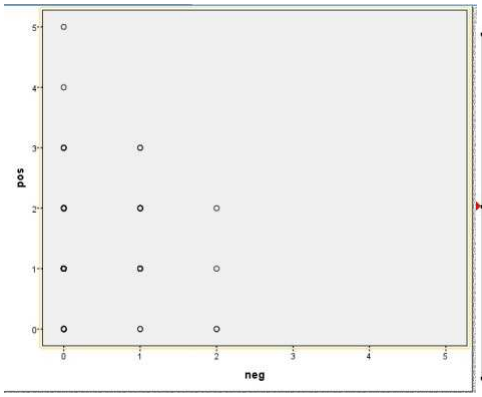

Figure 2:

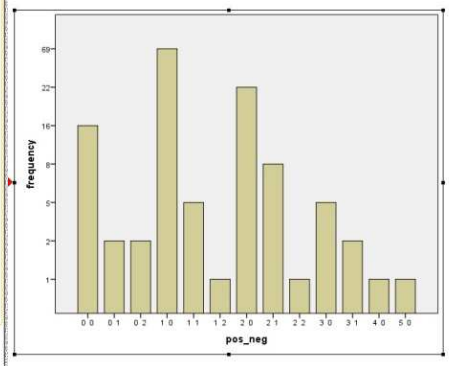

Figure 3:

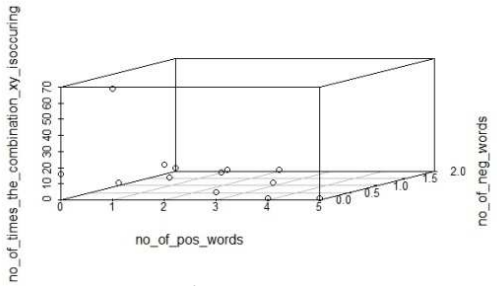

Figure 4:

Fig. 2 shows the distribution of positive and negative words per title. X-axis represents the # of negative words while the Y-axis represents the # of positive words per title. In Fig. 3, we plot the # of times the combinations are occurring. A combination 0 1

represents 0 positive word and 1 negative word per title. We combine Fig. 2 and Fig. 3 in Fig. 4. Here, we observe that number of titles with one positive word and one negative word are more. Thus we can conclude that most users write one positive word in the title of a positive reviews [# means number].

6. Recommendation system and word prediction

By calculating the scores from the review titles, we make a recommendation system. We use collaborative filtering here. We look for reviews having same title scores and suggest words to the user during the reviewing process. The following figure describes this in detail:-

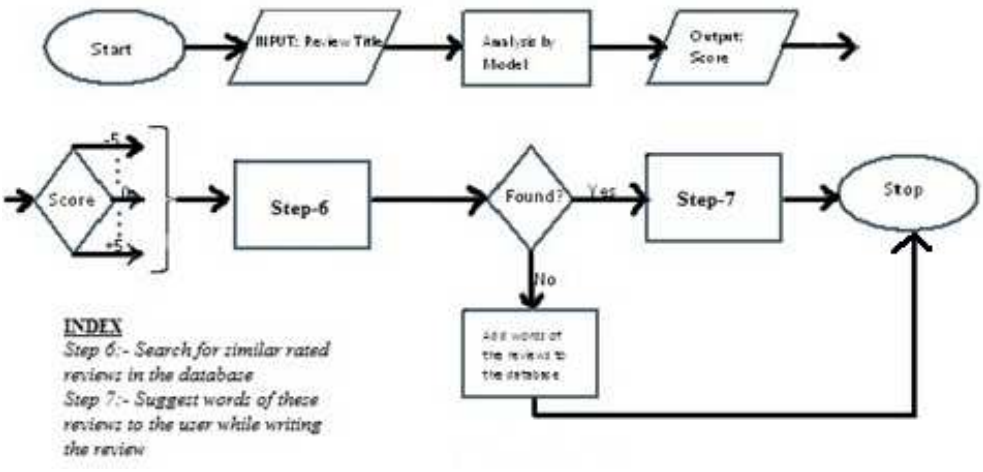

Figure 5:

7. Conclusion and future works

The model which we made can automatically generate the score of a product as soon as users write the title of the reviews. It works with high efficiency. It will save both time and resources of the reviewer and the company. An extra step during the process of reviewing can be eliminated. Thus, the process of reviewing will become easy and simple. This will involve more users reviewing the products. Eventually, both the company and the users will be benefitted. The company will be able to understand whether to display the review in its homepage or not just by scanning the title. It will thus save time and space required for computation. Customers can be suggested with words while writing reviews by using our recommendation system. This will help them to write the reviews.

We have trained our model using about one thousand reviews. It will be better to train it with more number of reviews. The corpora which we are using to classify contain around eighteen thousand words. More words can be appended to it to ensure that it works with greater efficiency. Porter stemmer has certain flaws such as it changes descriptive words like “awesome” to “awesom”, “little” to “littl”. The latter words don’t

convey any meaning. So, it is be appreciable to build a new stemmer which is more efficient than porter stemmer.

Acknowledgements: We are thankful to the popular e-commerce website www.flipkart.com from where we collected all the reviews. Natural Language Toolkit with Python 2.7 has been used for processing and analyzing the dataset. We sincerely acknowledge the developers of NLTK.

REFERENCES

1. Jacob Perkins, Python Text Processing with NLTK 2.0 Cookbook, *PACKT publishing* I.
2. Bird, Steven, Edward Loper and Ewan Klein (2009), *Natural Language Processing with Python*. O□Reilly Media Inc.
3. http://www.enchantedlearning.com/wordlist/positivewords.shtml, http://www.enchantedlearning.com/wordlist/negativewords.shtml; List of positive and negative words.
4. http://computer-ease.com/emotposi.htm, http://computer-ease.com/emotneg.htm List of positive and negative emoticons.
5. Srivastava, Ritesh, et al., Exploiting grammatical dependencies for fine-grained opinion mining, Computer and Communication Technology (ICCCT), 2010 International Conference on IEEE, 2010.
6. Bing Liu, Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers, May 2012.