Annals of Pure and Applied Mathematics
Vol. 7, No. xx, 2014, xx-xx
ISSN: 2279-087X (P), 2279-0888(online)
Published on X xxxxxxx XXXX
www.researchmathsci.org

A Model to Compute Degree of Polarity of Review Titles

Sohom Ghosh[1], Santanu Modak[2] and Dr. 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 XX xxx XXXX; accepted X xxxx XXXX

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

AMS Mathematics Subject Classification (XXXX): *xx*Xxx

1. Introduction

E-commerce industries have flourished as of late. People now find it easier to buy their necessities online. The e-marketing industries ask its customers to review the products they buy. Future customers get guided from these reviews. Thus, in today's world it has become extremely important for e-retailers to make their customers review the product they buy. This gives rise to the need of developing a fast and simple reviewing procedure. In this paper, we discuss about the innovative model we developed which can rate products by itself from the title of the reviews written by the customers. So, the customers don't have spent their time in rating the products they reviewed. This model reduces the process of reviewing by one step thereby making it easier and quicker.

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. 3 can be considered as neutral review. This type of problem is called "Regression" problem.

This paper explores a new innovative model which automatically generates the scores by analyzing the review title. We use unigram bag-of-words feature and proposed a fixed discrete rating scale (-5, +5). A score of -5 or near to -5 indicates that the review is extremely negative and that of +5 or close to +5 indicates that review is highly affirmative. The remaining sections of the papers are organized as follows. In section 2, the existing work done in the field of Sentiment

Analysis has been discussed. In section 3, Basic Terminologies which are used in this paper, are discussed. Section 4 represents our dataset and model. In section 5, we discuss some experimental results. Section 6 deals with analysis and 7 deals with recommendation system and word prediction. The conclusion and future work direction are presented in Section 8.

2. Related Works

In [5], they proposed a new definition of Opinion for structured analysis of unstructured opinion. In [6], they proposed a three phase approach. Phase 1: Corpora Acquisition Learning Phase. Phase 2: Adjective Extraction Phase. Phase 3: Classification. Phase 1, automatically extract positive or negative document from web for a specific domain. Phase 2 detect positive and negative adjectives. Phase 3 classify new document with a previous set of adjectives. In [7], they examine the problem of automatic sentiment analysis at sentence level. They compile a set of conjunction rules to determine relevant phrases for sentiment analysis. Finally use support vector machines to conclude that linguistic analysis plays a significant role in sentiment determination. Jusoh and Alfawareh applied Fuzzy Sets for Opinion Mining [8]. They used fuzzy lexicon and fuzzy sets in deciding the degree of positive and negative. If sentiment word is negative then we assign negative fuzzy set to that word, otherwise assigned positive fuzzy set. Then calculated degree of sentiment and visualized the output. Srivastava1 et al focused on binary grammatical relation or dependency (BGD) of words, the pattern in which each word in a sentence possesses grammatical corporations with other words for correct utterance of meaning [9].

3. Basic Terminologies

3.1 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.

3.2 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.

3.3 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, Regular Expression stemmer and so on. We use porter stemmer to design our model.

3.4 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.

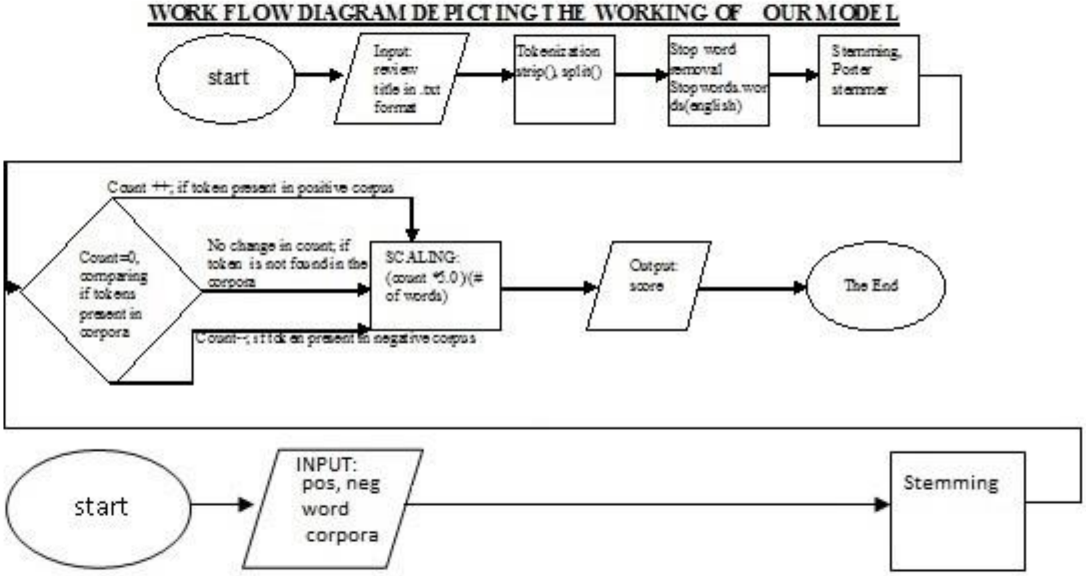
3.5 Regression

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.

4 The Model & The Corpora

4.1 The Model

Our model takes a text file as input. This text file contains the title of review as written by user. 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 -5 to +5.


WORK FLOW DIAGRAM DEPICTING THE WORKING OF OUR MODEL
start
Input: review title in .txt format
Tokenization strip(), split()
Stop word removal Stopwords.words(english)
Stemming, Porter stemmer
Count ++; if token present in positive corpus
Count=0, comparing if tokens present in corpora
No change in count; if token is not found in the corpora
SCALING: (count *5.0)/(# of words)
Output: score
The End
Count--; if token present in negative corpus
start
INPUT: pos, neg word corpora
Stemming

4.2. The Corpora

The corpora we developed to compute the sentiment score are:-

4.2.1 The Corpus of positive words

The corpus we used in this case contains more than six thousand words. We are not able to list the whole of it here. So, let’s look at a sample we made from it:

[[':) ', ':-)', '=)', '(:', '(-:', ':-D', ':D', ':d', ':-d', ':> ', ':->', ':))', ':-))', 'x-D', 'X-D', 'LOL', '(lol)', '(LOL)', ":'D", ":'-D", 'LMAO', '(lmao)', "x'D", "X'D", "x'-D", "X'-D", 'ROFL', '(rofl)', ":')", ":'-)", '(})', '({)', '>:D<', '>:d<', ""l",

'(K)', '(k)', ':*', ':-{}', ':{}', ':-^', ';-^', 'X-^', ';;)', ';;-)', ':-x', ':X', ':-X', ':x', 'l-)', 'l:-)', 'l:)', '\\:D/', '\\:d/', 'x-)', 'X-)', '=D>', '=d>', ':)', '>-', '$D', '$-D', '$-)' ,'absolutely', 'adorable', 'accepted', 'acclaimed', 'accomplish', 'accomplishment', 'achievement', 'action', 'active', 'admire', 'adventure', 'affirmative', 'affluent', 'agree', 'agreeable', 'amazing', 'angelic', 'appealing', 'approve', 'aptitude', 'attractive', 'awesome', 'beaming', 'beautiful', 'believe', 'beneficial', 'bliss', 'bountiful', 'bounty', 'brave', 'bravo', 'brilliant', 'bubbly', 'calm', 'celebrated', 'certain', 'champ', 'champion', 'charming', 'cheery', 'choice', 'classic', 'classical', 'clean', 'meritorious', 'miraculous', 'motivating', 'moving', 'natural', 'simple', 'skilled', , 'willing', 'wonderful', 'wondrous', 'worthy', 'wow', 'yes', 'yummy', 'zeal', 'zealous', '']

4.2.2 The Corpus of negative words

This corpus is too huge to accommodate it here. It has more than six thousand words in it. So, we are presenting a sample we made from it:

[':(', ':-(', '=(', ":'(", ":'-(", ':((', ':-((', ":'s", ":'S", ":'-s", ":'-S", ':-<', ':<', ':-[', ':[', ':C', ':-C', ':-c', ':c', ":'C", ":'c", ":'-C", ":'-c", 'T_T', ':-O', ':O', ':o', ':-o', '8-o', '8-O', 'O.o', ':Z', ':z', ':-Z', ':-z', ':-S', ':-s', ':S', ':s', '>-[', '>-(', '8o|', '8-|', 'X-(', 'x-(', 'X(', 'x(', ':-@', ':@', '>:o', '>-@', '>:0', '>-0', '>:-0', '>-o', '>-O', '[-(', ':-L', ':L', '[-x', '[-X', ':^o', '(U)', '(u)', 'abysmal', 'adverse', 'alarming', 'angry', 'annoy', 'anxious', 'apathy', 'appalling', 'atrocious', 'awful', 'bad', 'banal', 'barbed', ', 'hard', 'hard-hearted', 'harmful', 'hate', 'hideous', 'homely', 'horrendous', 'horrible', 'hostile', 'hurt', 'hurtful', 'icky', 'ignore', 'ignorant', 'ill', 'immature', 'imperfect', 'impossible', 'inane', 'inelegant', 'infernal', 'injure', 'injurious', 'insane', 'insidious', 'insipid', 'jealous', 'junky', 'lose', 'lousy', 'lumpy', 'malicious', 'mean', 'menacing', 'messy', 'misshapen', 'missing', 'misunderstood', 'moan', 'moldy', 'monstrous', 'naive', 'nasty', 'naughty', 'stressful', 'stuck', 'stupid', 'unfair', 'unfavorable', 'unhappy', 'unhealthy', 'unjust', 'unlucky', ', 'vindictive', 'wary', 'weary', 'wicked', 'woeful', 'worthless', 'wound', 'yell', 'yucky', 'zer']

5 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

CASE VI:

Dataset: Totally In Love With The Phone & Micromax,And Flipkart Hats Off To You

Dataset after preprocessing: total in love with the phone & micromax,and flipkart hat off to you

Score: 3

CASE VII:

Dataset: Micormax Canvas Doodle A111 - Mobile that could satisfy your needs

Dataset after preprocessing: micormax canvas doodle a111 - mobile could satisfy need

Score: 2

CASE VIII:

Dataset: Lagging Apps And Other Disadvantages Of My Samsung Galaxy S Duos 2

Dataset after preprocessing: Lag App And Other Disadvantag Of My Samsung Galaxi S Duo 2

Score: -2

6 Analysis:

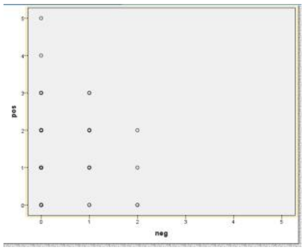


Fig-I

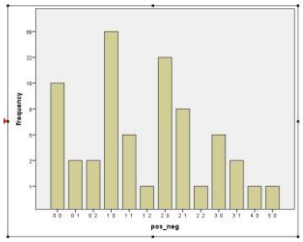


Fig-II

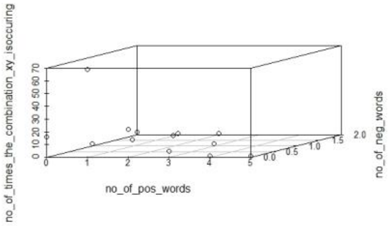
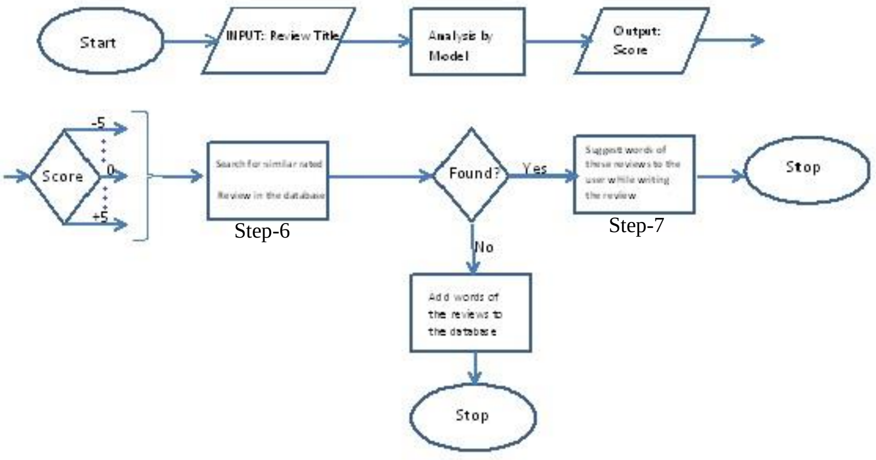


Fig-III

Fig-I 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-II, 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-I and Fig-II in Fig-III. 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 review. Thus, we may say a user who has already written a positive word have high tendency to use neutral words thereafter. So, from the review title we can predict what words a user will use while writing the review.

7 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:-


Start
INPUT: Review Title
Analysis by Model
Output: Score
Score
-5
0
+5
Search for similar rated Review in the database
Step-6
Found?
Yes
No
Suggest words of these reviews to the user while writing the review
Step-7
Stop
Add words of the reviews to the database
Stop

Step-6: Search for similar rated review in the database; Step-7: Suggest words of these reviews to the user while writing the review

8 Conclusion and Future Work

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.

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.

We get some aspect based reviews like “battery life is not good”, “camera is outstanding”- that is not considered as perfect review of that particular products. In future we will extend our model, which will considered full review, in place of only title of review and we will try to detect aspect based score.

9 Acknowledgement

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] Modak, S., & Mondal, A. C “A Model of Structured Opinion Format” International Journal of Advanced Research in Computer Science and Software Engineering, ISSN (Online): 2277 128X, ISSN (Print): 2277 6451, Volume 4, Issue 5, May 2014. Page 147-151

[6] Ali Harb, Michel Plantié, Gerard Dray, Mathieu Roche, François Trousset, Pascal.Poncelet *Web Opinion Mining: How to extract opinions from blogs?* Published in "CSTST'08: International Conference on Soft Computing as Transdisciplinary Science and Technology,.

[7] Ritesh Agarwal, T. V. Prabhakar, Sugato Chakrabarty. *“I Know What You Feel”:Analyzing the Role of Conjunctions in Automatic Sentiment Analysis*,6th International Conference, GoTAL 2008 Gothenburg, Sweden, August 25-27, 2008 Proceedings pp 28-39

[8] Jusoh, S.; Alfawareh, H.M., “Applying fuzzy sets for opinion mining,” Computer Applications Technology (ICCAT), 2013 International Conference on , vol., no., pp.1,5, 20-22 Jan. 2013 doi: 10.1109/ICCAT.2013.6521965

[9] Srivastava, Ritesh, et al. “Exploiting grammatical dependencies for fine-grained opinion mining.” Computer and Communication Technology (ICCCT), 2010 International Conference on. IEEE, 2010.

[10] Bing Liu, “Sentiment Analysis and Opinion Mining”, Morgan and Claypool Publishers, May 2012.