Filtering Malicious Comments with NLTK

Team 23

20170660 Daeun Choi
School of Computing, KAIST
cde1103@kaist.ac.kr

20170305 Hyoungjo Bhang
School of Computing, KAIST
bhanghj3094@kaist.ac.kr

20170221 Jaehyeon Myeong
School of Computing, KAIST
mjhbest@kaist.ac.kr

20170516 Jinwon Lee
School of Computing, KAIST
arsenal@kaist.ac.kr

20170798 Seungho Kim
School of Computing, KAIST
skim17@kaist.ac.kr

1 Introduction

Nowadays, with the rapid growth of portable technology and the internet, people are connected to each other more than ever. Along with the internet, countless platforms providing various contents from movies or music to social networking services have emerged, bringing interactions across time and space together.

At the same time, however, with the help of anonymity, it has become much easier to not only criticize someone but also denounce and attack them with words. From celebrities and politicians we have learned the negative results anonymity yields. Therefore, our work attempts to contribute to creating a 'better' cyberspace, adapting NLTK to filter malicious comments.

This work is organized as follows: In *Section 2*, more details on the problem we try to solve is elaborated. In *Section 3*, different techniques and algorithms are examined closely. Lastly, intermediate results and discussion will be held in *Section 4*.

2 Problem Statement

Our goal is to distinguish comments that give useful feedback, even those that criticize, from comments that simply reproach others. To start with, we built a comment discriminator assessing how positive and negative the reviewers are about the content. Next, more emphasis will be given on examining tone, or nuances.

For the interim report, we will be concentrating on first task, with extensive use of online comments with ratings((Ni et al., 2019), (Benlahbib, 2019), (Wang et al., 2011), (Pang and Lee, 2005), (Blitzer et al., 2007)). Finally, its robustness will be tested through unseen data without score rated by user such as Youtube comments.

3 Technical Approach and Models

This piece adopts two different approaches. First, it gathers individual scores from words or phrases composing reviews and comments. The frequency of words, appearances of proverbs and memes can fall into this category:

To begin, the text was tokenized into sentences, then into words. Each word is passed through the function that assesses the sentiment value of the word. As a baseline, SentiWordnet (Esuli and Sebastiani, 2006) from the NLTK corpus is used. Then, additional rules handling words that cannot be evaluated in SentiWordnet or contractions (such as don't) have been added manually.

Next, the average of each positivity score and negativity score is calculated. The overall score is summed up for the whole text and divided into the number of words (Word Average Method). This method is also implemented in unit of sentences(Sentence Average Method). The scoring function runs for each sentences and make average of all sentences. In the latter method, the adjectives and adverbs play significant roles on defining the sentiment of the sentence as a whole, so extra weight is given to the scores of adjectives and adverbs.

This first approach is strong in defining the "positivity" and "negativity" of individual words; however, the relationship between words and how they

form a text must be dealt by an another approach. For this approach, we will be focusing on relationships between words and sentences. Distinctive techniques using part-of-speech tags, the semantics of conjunctions, and giving extra emphasis depending on circumstances would be examined:

First of all, negative phrases containing positive words(Negative Phrase Ver.). If the negative word is followed by positive word, we judged that it is an usage of negative expression, so we reversed the positivity with negativity of the following word. This can be applied on the phrases such as "not like" or "disappointed perfectly".

4 Intermediate/Preliminary Experiments & Results

In order to get a glimpse of our interim results, we used the scatter plot. The x-axis is the original review score and the y-axis is the score calculated by our algorithm. The closer the shape of the graph is to the upper right, the more accurate the calculated score.

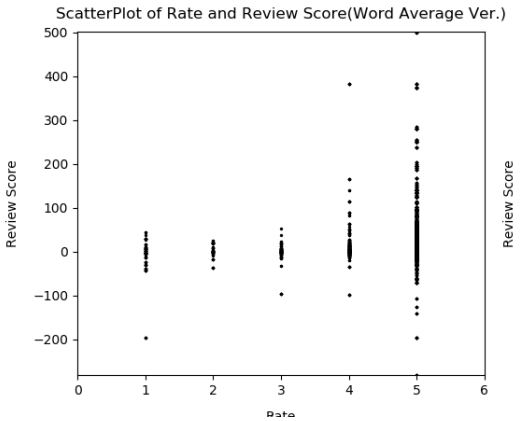

Figure 1: Result of the Naive Algorithm

The first graph shows the calculated result by the word average algorithm, which just uses the average positivity and negativity score of each word. There are some points where the higher the original score, the higher the calculated result. However, there are many points calculated as zero or near zero regardless of the original score, which results in inaccurate results. We found out that these points are mainly resulted by the words which clearly show the positivity or negativity, but SentiWordNet does not give out the sentimental score. Also, there are some words such as meme or proverb that do not even exist on SentiWordNet. Therefore, we should add some additional sentiment handler which we can add some words manually, or we can make our own SentiWordNet dictionary which is specialized in review analysis.

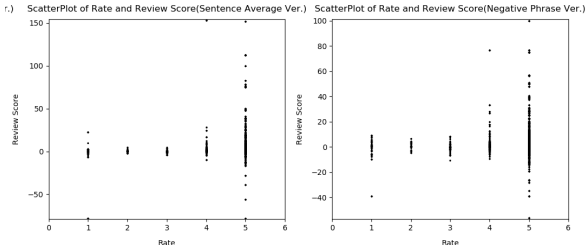

Figure 2: Result of Advanced Algorithm

This second and third graph shows the calculated result by advanced algorithms. The left graph is the result of the algorithm which first calculate the sentence's score and uses the average these sentences score. The right graph is the result of the algorithm with some handling of negative phrases. The overall shape is remained similarly with the first graph because of the limitation of the SentiWordNet dictionary, but we can see some points are improved in their scoring. For example, the number of points with a negative calculated score decreased, corresponding to 3 or 4 points in the x-axis. Also, for the sentence analyzing algorithm, there are fewer points with 0 scores compared to the first graph. Therefore, if we mix these techniques with the naive algorithm, we can improve our scoring algorithm.

From our interim results, it can be concluded that there can be further developments to be made: From the reviews, we found many exceptions from slang words to memes, idioms or proverbs. Additional exception routines will be applied. In addition, the current algorithm determines the preference by sentence. We can broaden the unit from each single sentence to the whole text. Furthermore, more detailed analysis on the part-of-speech(pos) tag of each word or negative phrases would be applied to word to word relationship. Also, checking conjunctions would helpful in determining sentence to sentence relationship. Lastly, algorithm to find out appropriate threshold to divide the result scores into a 0 5 or 0 10 discrete scale rating (as the data set rating) will be included.

References

Abdessamad Benlahbib. 2019. 1000 movie reviews (review + attached rating + sentiment polarity) for reputation generation.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic. Association for Computational Linguistics.

Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*, volume 6, pages 417–422. Citeseer.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.

Hongning Wang, Chi Wang, ChengXiang Zhai, and Jiawei Han. 2011. Learning online discussion structures by conditional random fields. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 435–444.