ECE467 Final Project Report

Susung Choi
Sangyeon Son
Eui Han

I. Abstract

In this project, we present a sentiment analysis which divides documents into two mutually exclusive groups: positive and negative. We use a classifying method based on positive and negative tags, along with some other supplementary tags. In an experiment with 2000 train documents (1000 positive, 1000 negative) and 1000 test documents, we obtained approximately 80% accuracy.

II. Introduction

Originally, the goal of this project was to predict a star rate of yelp reviews by only looking at the review text. However, due to time constraint and the complexity of the problem itself, we failed to achieve such high resolution in figuring out the document polarity.

Instead, in this project, we aim to do more classical sentiment analysis which divides documents into two mutually exclusive groups: positive and negative.

III. Algorithm

Our first attemt was to use Naive Bayes classifier with Bag of Words approach. Research has shown that using Naive Bayes classifier with unigrams or bigrams can be surprisingly effective in sentiment analysis[1]. Infact, by only adding some stopwords and modifying a little from the first classification project, we have achieved higher than 70% accuracy with Naive Bayes Classifier and Unigram approach.

However, we wanted to try something different for this project, and here is what we did:

First, we trained the machine with traditional word frequency. We first parsed the input with nltk regular expression tokenizer and stemmed it with nltk Porter Stemmer. Then, we generated both unigram and trigram from this stemmed set of words.

Next, we counted how many times each unigram/trigram has appeared in a given categroy (positive or negative). But since unigrams tend to have much higher frequency than trigrams do, we gave 50 times more weight to trigrams, except for the first occurence. For example, if a same trigram appears twice in one document, it will have 1 + 50 = 51 frequency whereas a unigram appearing twice will have 1 + 1 = 2 frequency.

Then, we sorted the set of ngrams in the order of higher frequency and took only top 1000 distinct ngrams from each category and discarded everything else. These 1000 ngrams become our positive / negative features. We saved the result in a separate file and ended the training phase.

In the testing phase, we first parsed the document same way as described above: parse, stem, and generate unigram/trigram. Then, we iterated through each ngram obtained from the testing document and checked if each word is in top 1000 features we had obtained in the training phase. If we find the ngram in top 1000 positive features, the document gets positive scores and if we find the ngram in negative features, the document gets negative scores. At the same time, we also check what the previous word of that ngram was. If the previous word of the ngram was ”inverting words” (not, no, not quite), we multiply -1 to the score. If the previous word of the ngram was ”increment words” (very, much, way, too) we multiply the score by a constant number. If the previous word was ”decrement words” (a little, a few) we divide the score by a constant number. Then, if the final score is positive, we conclude that the document has positive sentiment, and if not, we conclude that the document has negative sentiment.

IV. Evaluation

We evaluated our algorithm with 2000 training document and 2 training sets, each consisting of 1000 documents. The original Yelp Review Database has 2 million reviews, and we randomly picked out 1000 positive reviews (star 4, 5) and 1000 negative reviews (star 1, 2) as our training set. For test set 1, we randomly chose 1000 reviews out of 2 million reviews, not allowing any duplicate. Because there are more positive reviews than there are negative reviews, test set 1 consists 770 positive reviews and 230 negative reviews. For test set 2, we picked out 500 positive reviews and 500 negative reviews from 2 million reviews without allowing any duplicate.

If a review has 4 or 5 star and if the predicted tag was ’positive’ then we counted it as a success. Similarly, if a review has 1 or 2 star and if the predicted tag was ’negative’ we counted it as a success. We calculated a total success rate as a fraction of number of success over total number of testcases.

V. Result

We obtained accuracy around 80% for both datasets. Considering that state of art results show mid 80% accuracy, performance of our classifier is proven to be competitive.

VI. Future Works

Parameters of our classifier (for example giving 50 times more weight to trigrams compared to unigrams) are determined empirically. And thus there is a room for improvement in accuracy, if we could optimize the parameters either mathematically or by more rigorous empirical means.

VII. Reference

1 https://www.cs.cornell.edu/home/llee/papers/sentiment.pdf

2 https://www.yelp.com/html/pdf/YelpDatasetChallengeWinner_ImprovingRestaurants.pdf