Filtering and Summarizing Reviews

Nikhil Tej Lingutla Venkata
EECS Department
Oregon State University
Corvallis, OR 97331
lingutln@onid.orst.edu

Arwa Z. Hamid
EECS Department
Oregon State University
Corvallis, OR 97331
hamida@onid.orst.edu

Abstract

Online reviews over a product have become an important source of information, not only for customers to find opinions about products easily and share their reviews with peers, but also for producers to get feedback on their products. As the number of product reviews grows, it becomes difficult for users to search and utilize these resources in an efficient way. Therefore, the solution for this problem is to provide an abstract view of a review to the user and filter the review as supportive or non-supportive. We have approached this problem using different variants of Naive Bayes classification algorithms, sentiment lexicon and Synthetic words for sentiment analysis. These algorithms are Bernoulli Naive Bayes, Multinomial Naive Bayes, Binarized Multinomial Naive Bayes, Log of frequency counts, Sentiment Lexicon and Synthetic Words. After performing the sentiment analysis on the reviews, we have used the most recognized approach of Term Frequency (TF) and Inverse Document Frequency (IDF) for extracting the signature words from the reviews. In this way, applying the TF*IDF after performing the sentiment Analysis results in signature words which provide the features of the product and the opinion user has expressed on the product based on a review.

1 Introduction

Nowadays, e-commerce has become one of the major activities conducted over the Internet. In order to enhance customer satisfaction and shopping experience, it has become a common practice for online merchants to enable their customers to review or to express their opinions on the products that they have purchased [1]. By writing reviews, customers can evaluate how good the product is, moreover, manufacturers can gather customers' feedback about the product. Therefore, millions of reviews on products and services are being published online every day. However with this massive amount of reviews, it becomes very hard to figure out good reviews from bad reviews and the reviews that are informative. As a result, filtering and providing an abstract view of online reviews is an interesting Machine learning problem. Text summarization can be defined as reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. The summarization could be produced by summarizing multiple reviews, or single review. Moreover, there are many different approaches for summarizing the reviews which are outlined as the following [2]:

1- Extraction: copying the important information from the review.

2- Abstraction: Extracting the signature words. (For our project, we have used this technique)

3- Fusion: combines extracted parts coherently.

4- Compression: delete unimportant sections of the text.

In this project, we have implemented some sentiment Analysis algorithms in order to predict the sentiment (whether a review is positive or negative) of the reviews. The different variants for Sentiment Analysis are as the following:

1- Naive Bayes with Bernoulli distribution

2- Naive Bayes with Multinomial Distribution

3- Binarized Multinomial Naive Bayes.

4- Using the log of frequency as suggested by J Rennie et. al [3].

5- Using pre annotated polarity words, Sentiment Lexicon.

6- Using Synthetic words for bernoulli naive bayes

7- Using Synthetic words for multinomial naive bayes

We have selected Large Movie Review Dataset[4] as it is a benchmark for sentiment analysis and care was taken such that no more than 30 reviews are allowed per given movie. As a preprocessing step, some python scripts are written in order to convert the review data which are in XML format into sparse representation (similar to 20 Newsgroups dataset). We have also written python scripts to eliminate stop words and symbols[5] present in the reviews.

2 Sentiment Analysis Algorithms

Since the review could express a positive or negative sentiment, there are different algorithms that can be used in order to classify whether the review contains positive sentiment or negative sentiment. The following is a brief explanation of different sentiment Analysis algorithms that we have used.

2.1 Naive Bayes Models

Naive Bayes is one of the successful methods for text classification. It assumes that all attributes of the examples are independent of each other given the context of the class. Four variants of Naive Bayes have been used for this project: Naive Bayes with Bernoulli distribution, with Multinomial distribution, Binarized Multinomial Naive Bayes, and the Log of frequency count.

2.1.1 Naive Bayes with Bernoulli Distribution

Bernoulli Naive Bayes is a supervised learning method, and probabilistic model. It specifies that a review is represented by a vector of binary attributes indicating which words appear in the review or not. This model doesn't care of how many times a word appears in the review.

$$P(word_i / Class_j) = \frac{(\# \, of \, documents \, that \, contain \, word_i + 1)}{(Total \, \# \, of \, documents \, in \, class_j + (1 * 2))}$$

The above formula also shows the laplacian or add one smoothing

2.1.2 Naive Bayes with Multinomial Distribution

Multinomial Naive Bayes is a supervised learning method, and probabilistic learning method. It specifies that a document is represented by the set of word occurrences from the document. So, it cares about the number of occurrences of each word in the document.

$$P(word_i / Class_j) = \frac{(\# \, of \, word_i \, in \, class_j + k)}{(Total \, \# \, of \, words \, in \, class_j + (K * V))}$$

where K is laplacian constant and V is size of Vocabulary.

2.1.3 Binarized Multinomial Naive Bayes

Sometimes word occurrence may matter more than word frequency. For example, the occurrence of the word (fantastic) would shows that the review is good, but the fact that it occurs 5 times may not tell us much more. In fact, Binarized Multinomial Naive Bayes is slightly different variant of Multinomial Naive Bayes and bernoulli naive bayes. Instead of taking the number of times a word occurred in a document, we just take into account whether a word is occurred or not over all the distinct words in the document. We can achieve this just by removing the duplicates in each document before performing Multinomial Naive Bayes.

2.1.3.1 Difference between binarized Multinomial Naive Bayes and Bernoulli Naive Bayes

In binarized multinomial naive bayes, while calculating the conditional probability we use the total number of distinct words in documents belonging to a particular class while in Bernoulli naive bayes we take into account the number of documents that comes under a specific class.

2.1.4 Using the log of frequency count

While in Naive Bayes with Multinomial Distribution all the occurrences of a word are taking in to consideration, and in Binarized Multinomial Naive Bayes only the presence of a word is taking in to account. Instead of taking 1 or total frequency count, JDRennie et al [3] suggested some number in between these two like the Log of frequency count is another variant of Naive Bayes

$$P(word_i / Class_j) = \frac{(log(\# \, of \, word_i \, in \, class_j) + k)}{(log(Total \, \# \, of \, words \, in \, class_j) + (K * V))}$$

where K is laplacian constant and V is size of Vocabulary

2.2 Using pre-annotated polarity words

Another method of sentiment analysis called Sentiment Lexicon. This method uses the preannotated polarity words as found in [6]. By using these polarity words, we can extract the count of how many positive and negative polarity words are presented in a review. Then, assign the maximum polarity as the sentiment of the review. In fact, this approach does not need any training data and hence can be used in the initial stages of Sentiment analysis of a Machine Learning problem where the training data is zero or less. This approach is not very robust and can be fooled easily, hence we have thought of another approach where we have combined the ideas from Naive Bayes and the pre-annotated polarity words which resulted in Synthetic words approach.

2.3 Synthetic words Approach

We have used “positive_word” and “negative_word” as the two synthetic words. We parse through the document and whenever we encounter a positive polarity word, we append “positive_word” to the document and whenever we encounter a negative polarity word, we append “negative_word” to the document. As a result we have added “positive_word”, 'x' number of times to the document where x is the number of positive polarity words in the document. We have added “negative_word”, 'y' number of times to the document where 'y' is the number of negative polarity words in the document. This way we are building synthetic features into the document which is directly proportional to the count of positive and negative polarity words respectively.

3 Extracting the signature words

After performing the sentiment analysis on the reviews, signature words should be extracted from the reviews. To do this, TF*IDF weights are used to calculate how important a word is to a document. TF calculate the number of times a word appears in a document and can be calculated as following

$$tf = \frac{\#\ of\ a\ wordi\ appear\ in\ a\ document}{total\ \#\ of\ words\ in\ a\ document}$$

IDF measure of whether the term is common or rare across all documents.

$$idf(t, D) = \log \frac{total\ \#\ of\ documents\ |D|}{\#\ of\ document\ where\ the\ word\ t\ \ appears}$$

Then the TF * IDF is calculated as tf*idf [7]-[8]. The descending order of the TF_IDF wights gives us the words in decreasing importance order.

4 Results and discussion

We have tested our classifier accuracies for all the above 7 approaches and we found out that Naive Bayes with Bernoulli distribution provided the best accuracy followed by synthetic words approach comparing to the other methods.

4.1 Empirical results

We have evaluated our Sentiment Analysis classifier on the Large Movie Review Dataset [4] and we were able to achieve the following accuracy.

Table 1: The accuracy achieved by various sentiment analysis algorithms

Algorithm	Accuracy
Naive Bayes using Bernoulli	85.52041%
Naive Bayes with Multinomial	84.66221%

Binarized multinomial Naive Bayes	50.33430%
Using log of frequency counts	58.94621%
Sentiment Lexicon	73.62539%
Synthetic words with Bernoulli Naive Bayes	85.11127%
Synthetic words with Multinomial Naive Bayes	85.10129%

4.2 Discussion

It is usually claimed that the multinomial model gives higher classification accuracy than the binary independence model on text documents because it models word occurrence frequencies [9]. Contrary to this belief, we showed that word frequency hurts more than it helps, and that ignoring word frequency information can improve performance which is in coincidence with [10]

We have also empirically showed that the performance of multinomial naive bayes can be improved further considering the pre annotated polarity words that are freely available on the web [6] We have improved accuracy of multinomial naive bayes by 0.6% using polarity words.

4.3 Sample Review

Following is one of the negative sentiment sample review taken from Large Movie Review Dataset [4]

"This movie was a complete waste of time. The soundtrack was a bad story, was lame and predictable and the acting was terrible. One of the worst movies I have ever seen. After the first ten minutes the rest of the film was completely obvious"

The above review is classified as negative sentiment

The first five words that are extracted using tf*idf approach are the following:

- Lame
- Soundtrack
- Predictable
- Terrible
- waste

5 Conclusion

The objective of this project is to provide an abstract view of a large number of customer reviews of a product. We have performed the sentiment analysis on the reviews using the algorithms like Family of Naive Bayes, Sentiment Lexicon and Synthetic words. Our experimental results indicate that the Bernoulli Naive Bayes has provided the best accuracy which is 85.5% followed by the Synthetic words approach. Moreover, after performing the sentiment analysis on the reviews, the Term Frequency (TF) and Inverse Document Frequency (IDF) have been applied for extracting the signature words from the reviews,

6 References

[1] Hu, M., Liu, B.(2004) Mining and Summarizing Customer Reviews. *In Proceedings of the ACM SIGKDD Intl*. Conf. on Knowledge Discovery and Data Mining (KDD04), pp. 168-177.

[2] Das, D., Martins, A. (2007) A survey on automatic text summarization.

[3] JDRennie, L shih, J Teevan ICML 2003 "Tackling the poor assumptions of Naive Bayes text classifiers"

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

[5] http://www.lextek.com/manuals/onix/stopwords1.html

[6] http://www.wjh.harvard.edu/~inquirer/homecat.htm

[7] Salton, G and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management* 24 (5): 513–523.

[8] Sparck, J., Karen (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28 (1): 11–21.

[9] McCallum, A., Nigam, K.: A comparison of event models for Naive Bayes text classification. In: Learning for Text Categorization: Papers from the AAAI Workshop, AAAI Press (1998)

[10] Karl-Michael Schneider et al, Techniques for improving the performance of naive bayes text classification.