International Journal of Electrical and Electronics Research ISSN 2348-6988 (online)
Vol. 3, Issue 3, pp: (1-5), Month: July - September 2015, Available at: www.researchpublish.com

Bigram Extraction and Sentiment Classification on Unstructured Movie Data

Shubham Tripathi

Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur 302017, India

Abstract: **Classifying text with respect to sentiment is being used very often in diverse fields these days. Number of movies releasing worldwide is increasing day by day. As the movie market is becoming more and more popular, movie reviews help in providing an insight to the movies. With number of reviews increasing up to thousands or even lakhs, it is imperative to classify them in the best possible manner to provide apt ratings as per the scale. Conventionally, sentiment prediction systems just look at words in isolation and provide positive and negative scores to the sentences. In this research, we aim to employ a statistical technique (Naïve Bayes classification) to the movie review data and find the overall sentiment of the document. We compare the conventional prediction system with a new technique to exploit features. The entire research is done in parts. Mining features of reviews that have been commented on the websites, Identifying relevant corpus to apply algorithms on after mining the corpus and finally concluding the results. We apply methods using *R Studio* software. I conclude by examining factors and devising ways for feature selection in the above mentioned technique.**

Keywords: **sentiment classification, review, text mining, feature extraction.**

1. INTRODUCTION

Today, very large amounts of information are available in on-line documents. As part of the effort to better organize this information for users, researchers have been actively investigating the problem of automatic text categorization.

A quite large segment of that work focuses on bunching up the data according as the subject matter (Politics, Drama, Entertainment, Sports). However, in the recent years, online discussion Forums, blogs, review sites etcetera are providing the liberty to individuals to speak their mind with no constraints. Much of that text data is derived by the sentiments of the person which provides a new way to categorise those chunks of data. Tagging those text data based on sentiments provides a better, more clearer and a strong insight to the reader. Sentiment classification also plays a significant role in classifying and tagging important news and information from the more casual and of course the spam content.

In this paper, we examine the effectiveness of applying a particular machine learning technique to the sentiment classification problem. Much of the sentiment is conveyed through prosody and not word choice. Here, we apply a mix of both of these methods to get dependable results. For example, this statement "This movie is neither funny nor good looking!" has no negative words in it. In fact, many algorithms may give it a positive sentiment which we clearly see is not. Hence, sentiment analysis requires more understanding than a normal topic based classification.

Easy tools and methods have been used that give a mathematical output to the entire corpus for better filtering along with conventional Text mining methods and functions. Documents have been classified as positive or negative according to their net polarity score. Documents with extreme polarity scores were kept.

The algorithm which has been used to assign value to polarity of each document first utilizes the sentiment dictionary (*Hu and Liu, 2004*) to tag polarized words, i.e. the words that have been marked as important. Hence, the nature of the score of each document totally depends on the polarity dictionary.

2. MOVIE-REVIEWS DATASET

The main reason to choose movie reviews to work on is its flexibility. They have various words which depict different and unambiguous sentiments. Their domain is also easily available.

The data source used is the *Internet Movie Database (IMDb)* archive of ***rec. arts. movie. reviews***. The dataset available were HTML files which were expressed with some numerical values. The dataset **polarity_html** *(HTML files)* and the sorted out files, all can also be found at the website *http://ai.stanford.edu/~amaas/data/sentiment/* . The entire reviews by authors have been subdivided into positive, negative and neutral data sets. I have classified them as positive or negative in order to null out the neutral text.

The link provided is the distribution webpage for the processed movie-review data. It is to be mentioned that the technique being discussed can be applied to any form of text data as long as the size of training data is apt.

Document Classification:

In this section, we discuss the way in which documents have been classified. The dataset has around 100,000 review documents. Entire corpus of documents were loaded and polarity score of each document was calculated. Polarity scores were not constrained to get values within [-1,1] and hence documents having score >1 or < -0.5 were used. A total of 25,000 documents were selected. Rest 75,000 documents were rejected. The training and testing set was in 3:1 ratio.

The documents are read in as corpus and the entire corpus is then split to give 17,500 training documents and 7,500 testing documents. Punctuation marks such as "?", ".", ";", "|" are taken into account while splitting the documents. The documents have been used as their originals, neither are they *stemmed* nor the English *stopwords* have been removed from the documents. Stopwords do make a lot in terms of quantity in documents. But, they help in providing better insights to what a sentence mean, and hence their polarity score. Sentiments generally depend upon the prosody of the sentences. But, the percentage presence of statements depending only on the manner of speaking is very less. Hence, it is quite certain to believe that people generally use common words to express strong sentiments. Although it is highly unproductive to use these words alone to train and predict a model, hence, if used with n-grams, they yield good results.

For this, I asked two other fellow mates (independently) to give me a list of words, for both positive and negative sentiments, which they would use to express sentiments when writing a movie review. Words selected have been listed out in Figure *1*. Similar words provided by both these fellows have been kept in either of the word list and not both.

Machine Learning Techniques:

Our aim in this work was to examine whether it suffices to treat sentiment classification simply as a special case of topic-based categorization (with the two "topics" being positive sentiment and negative sentiment), or whether special sentiment-categorization methods need to be developed. I experimented with two standard algorithms: Naive Bayes classification and maximum entropy classification. The philosophies behind these two algorithms are quite different. *Maximum Entropy method gave highly unproductive results and so was left out of the project.*

To test these methods, I have implemented the machine learning methods with unigrams and bigrams. Unigrams were used with a direct approach and bigrams, in a more elaborate manner. A mixture of words from both the fellows (*Fellow 1 and Fellow 2*) were taken. For example, I included the word "*funny*" and extracted a bigram *"funny enough*" to test the algorithms.

Naive Bayes Classification:

Abstractly, naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some *n* features (dependent variables), it assigns to this instance probabilities

$$p(C_k|x_1, \ldots, x_n)$$

for each of *k* possible outcomes or *classes*.

Fellow	Suggested Positive Words	Suggested Negative Words
Fellow 1	*good, amazing, great, mesmerizing, charming, wonderful, superb, touching, artistic, brilliant*	*nightmare, horrible, lame, ditch, boring, sucks, nasty, disaster, waste*
Fellow 2	*spectacular, fascinating, awesome, terrific, elegant, perfect, refreshing*	*awful, evil, worst, bad, poor, unwatchable, stereotype, slow, aimless, wrong*

Figure 1: Suggested words for expressing strong sentiments. A prediction accuracy of 58.1% was observed while using these words alone as features.

The problem with the above formulation is that if the number of features *n* is large or if a feature can take on a large number of values, then basing such a model on probability tables is infeasible. The problem is therefore reformulated to make it more tractable. Using Bayes' theorem, the conditional probability can be decomposed as

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

Where P(x) plays no role in selecting C_k. to estimate the term P(x | C_k), Naïve Bayes decomposes it by assuming the number of times a particular feature (positive, negative, suggested words) appears in the pre-processed sentences are conditionally independent.

Despite its simplicity and the fact that its conditional independence assumption clearly does not hold in real-world situations, Naive Bayes-based text categorization still tends to perform surprisingly well; indeed, papers by Domingos and Pazzani (1997) show that Naive Bayes is optimal for certain problem classes with highly dependent features. On the other hand, a much sophisticated method, i.e., maximum entropy algorithm yield better results.

Maximum Entropy Classifier*:

The Max Entropy does not assume that the features are conditionally independent of each other. The MaxEnt is based on the Principle of Maximum Entropy and from all the models that fit our training data, selects the one which has the largest entropy.

The probability distribution equals

$P(x,y) \equiv \frac{1}{N} \times$ *number of times (x,y) occurs in the sample*

Its estimate of P(C_k | x) takes the following exponential form

$P(C_k \mid x) = \frac{1}{Z(d)} \exp(\sum_i \lambda F(d,c))$

Where Z(d) is the normalization function. F is the feature function for each sentence and is defined as follows

$$F(d,c) = \begin{cases} 1 & n(d) > 0 \\ 0 & Otherwise \end{cases}$$

Here, *n(d)* is the number of times the feature (Suggested words) word appears. For instance, a particular feature/class function might work if and only if the bigram "*really amazing*" appears and the document's sentiment is hypothesized to be positive. Importantly, unlike Naive Bayes, MaxEnt makes no assumptions about the relationships between features, and so might potentially perform better when conditional independence assumptions are not met.

The λ are feature-weight parameters; inspection of the definition of PME (Probability for maximum distribution) shows that a large λ means that features are considered a strong indicator for a particular class of document.

**Results not included.*

Algorithm and bigrams extraction:

The idea of my algorithm is to simply find suitable document for sentiment classification out of all the documents. For this, we pre-process the corpus to remove unwanted discrepancies. We then find the polarity score of each document and classify them as positive, negative and test documents. To find the polarity score, we take use of the sentiment dictionary and extra words supplied. For bigrams, we make use of the unigrams selected earlier for the model and select words

appearing adjacent to it in the training set. We then further refine the words seeking unigrams (as per the sentiment dictionary) in the adjacent selected words. It is so required as majority of the unigrams are followed or preceded by stopwords. Using stopwords in bigrams is highly unproductive. Number of bigrams used were approximately *20%* of the unigrams used (132 bigram features, 577 unigram features). We then use the positive and negative classified documents to predict our test documents using Naïve Bayes classification.

1. Corpus <- Entire Corpus
2. U <- set of unigrams
3. B<- set of bigrams *#NULL at beginning*
4. For each document in training set{
Pre-process document by removing numbers, changing to lowercase.
For each pair of adjacent words (w1, w2,w3) where w2 is an unigram, If (w1 is in U or w2 is in U), add bigram "(w1/w3)+w2" to B.
}
5. For each bigram b in B{
If(Frequency of b>20 in all documents), keep the bigram. Else, remove b from B.
}
6. To find documents {
If(polarity score>1 in corpus), keep as positive document. If(polarity score< -0.5 in corpus), keep negative document.
}
7. In test corpus, If(polarity score < -0.5 || polarity score >1), keep test document.
8. Apply ML technique.

3. EXPERIMENT

Finding the Accuracies:

To calculate the accuracies of the above discussed classifiers, the entire corpus was brought down to certain set of documents using appropriate text mining.

A quantitative approach was chosen to select relatively polar documents out of the bag of data. To assign the polarity of each documents, we first utilize the sentiment dictionary *(Hu and Liu, 2004)* to tag the important words along with the additional suggested positive and negative words. A context cluster of words is pulled from around this polarized word (By default four words before and two words after) to be considered as valence shifters. The words in this context cluster are tagged as neutral, positive and negative. Neutral words hold no value in the equation but do affect word count in the document. Each polarized word is then weighted based on the value provided and the *polarity** is measured.

**For details on polarity calculation, refer to Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. National Conference on Artificial Intelligence.*

The set of algorithms are run independently on negative and positive reviews and a set of documents are selected based on polarity score < 0 and > 0 respectively. Hence, we get our data set ready with 25,000 documents.

To calculate accuracies, we apply three sets of features to Naïve Bayes classifier. Our baseline result for human selection of words yielded 58.01% accuracy in classifying documents. Other results are mentioned in the Figure 2 below.

4. RESULTS

Initial Unigram result:

Naïve Bayes classifier classifies with an accuracy of 90%. The features were selected using sentiment dictionary and a minimum frequency of words was set as a cut-off. The one with maximum accuracy among the lot was selected. This accuracy easily crosses the 0.5801 mark and gives us a fair estimation for predicting sentiments. This provides suggestive evidence that sentiment

Features	Accuracy
Human Words	0.5801
Unigrams only	0.9065
Bigrams only	0.502
Unigrams + Bigrams	0.99

Figure 2: Results obtained on 5 fold CV

categorization is more difficult than topic classification, which achieves accuracy of 90% for particular categories *(Joachims, 1998; Nigam et al., 1999).* We could not perform the natural experiment of attempting topic-based categorization on our data because the only obvious topics would be the film being reviewed. Unfortunately, in our data, the maximum number of reviews per movie is too small for meaningful results.

Bigram Results:

In addition with unigrams, we studied the effect of bigrams with unigrams to predict sentiments. We extracted bigrams with technique that yielded extremely good results. Bigrams and unigrams applied together yielded 99% accuracy with Naïve Bayes method. Bigrams, when extracted naively, i.e. when are formed by adding intensifiers to unigrams perform poorly on trained model. But, when bigrams are extracted with the technique mentioned above in the paper and used with unigrams, they give acceptable results.

Results may vary with increase in size of the data.

Less frequent words were removed.

5. CONCLUSION

The results produced via machine learning technique are quite good in comparison to the human generated baselines. Bigram extraction turns out to be useful. Relatively, it is a known fact that Naïve Bayes is one of the most fundamental statistical model available to us. Still, work can be steered in the direction to improve the features/ explanatory variables. It is evident that bigrams alone are not that useful but when used with unigrams, can provide important results. Bigrams alone performed the worst and unigrams along with bigrams yielded almost perfect results. Work can be extended with using trigrams and n-grams as features for sentiment classification. For selecting n-grams as features, certain stopwords can be used in extracting features and I look forward to it in future work.

REFERENCES

[1] Bo Pang and Lillian Lee. 2002. Thumbs up? Sentiment classification using machine learning techniques. Proceedings of EMNLP 2002.

[2] Duda, R. O., & Hart, P. E. 1973. Pattern classification and scene analysis. Wiley and Sons, Inc.

[3] Tackling the poor assumptions of Naïve Bayes Text Classifiers. 2003. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC.

[4] Bo Pang and Lillian Lee. 2008. Opinion Mining and sentiment analysis. Foundations and Trends in Information Retrieval

[5] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis.

[6] C. O. Alm, D. Roth, and R. Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction. In Proceedings of HLT/EMNLP

[7] F. Li, M. Huang, and X. Zhu. 2010. Sentiment analysis with global topics and local dependency. In Proceedings of AAAI.