Sentiment Analysis on Movie Review Data [*]

Manish Sharma
Dept of CSE
IIT Guwahati
Guwahati, India
manish.sharma@iitg.ernet.in

Gaurav Madhuker
Dept of CSE
IIT Guwahati
Guwahati, India
m.gaurav@iitg.ernet.in

Monika Srivastava
Dept of CSE
IIT Guwahati
Guwahati, India
monika@iitg.ernet.in

Ravindra Singh Gurjar
Dept of CSE
IIT Guwahati
Guwahati, India
ravindra@iitg.ernet.in

ABSTRACT

This report provides the basic concept of what we plan to do in the term project on *Sentiment Analysis on Movie Review System*. We discuss the various approaches that we take for solving this problem and also the data set that we plan to perform those experiments on. We also discuss the work distribution and the current progress in the project.

Keywords

Sentiment Analysis, Natural Language Processing

1. INTRODUCTION

The problem of Sentiment Analysis has been studied for a long time. Both supervised and unsupervised learning have been used to solve this problem. Bo-Pang[4] used Imdb[3] dataset and applied various supoervised Machine Learning techniques such as Naive baye's, SVM, Entropy Maximisation to conclude if a Document was positive or negative. Turney[6] used unsupervised techniques to solve the problem of sentiment analysis on various domain. They used *adjectives* and *adverbs* as possible indicators of the subjectivity of the sentence and calculated PMI-IR for predicting the sentiment of the sentence. A recent work by Stanford researchers[5], applied recursive neural network on this problem on the parse tree created using the Imdb data set. This method outperformed all the existing techniques and has cleared ways for new avenues for the future work.

There are many difficulties in determining the polarity of a sentence/document. It is also very different from topic classification of a document because topics can be identified using *keywords* alone but sentiments can be expressed in more subtle manner. For example, "How can you see this?" contains no negative word in its own but conveys a negative sentiment. Also measuring the degree of polarity, i.e how *positive*,*negative* or neutral a sentence/document is another challenging task. *Subjectivity* of a sentence also affect to the degree of classification. For example, "X did a great work in the movie." does not say anything about the movie itself or how the reviewer has rated the movie. Finding these objective sentences from the review data is also a difficult task. Also using phrases can improve the quality of sentiment analysis. We also need to take care of the negative words/sentiments such as "bad","awful","not" etc. before making our decision.

In this report, we discuss the following things. Section 2 discusses the methods that we are planning to apply for performing sentiment analysis on the moview review data. We also discuss about the data set that we will be using to perform these experiments. In section 3 we discuss about the results that we obtained in our experiments and in section 4 we conclude our report.

2. METHOD

In this section, we discuss about the methods that we use to implement the sentiment analysis and the data set that we will use to perform these experiments. Also we perform experiments using the following techniques:

1. Naive Baye's Algorithm
2. Entropy Maximisation
3. SVM
4. Recursive Neural Networks
5. PMI-IR
6. Random Forests

We will discuss about each in the following subsections:

2.1 Data set

We used the Cornell data set[2] and Standford dataset[1] for performing all our experiments. The Cornell data is a movie review collection from *IMDB* and *www.rottentomatoes.org*. The Cornell website conatins three types of dataset:

- Dataset for positive and negative reviews

The first is a dataset of 1000 positive and 1000 negative processed reviews. The Stanford data set,*Sentiment Treebank* is a fine grained sentiment labels of 215,154 phrases in the parse tree of 11,855 sentences. These sentences have been taken from the Cornell data set and have been processed from them. This tree bank captures the essence of phrasal queries and has been shown to give good results after training on them.

[*] This template is adapted from http://www.acm.org/publications/article-templates/SIG%20Proceedings%20Template-May2015%20Zip.zip

2.2 Naive Baye's Method

We use a set of positve and negative words and use *bag of words* model to find the probability of a document to belong to a particular class c (positive or negative). Hence the probability is

$$P(c/d) = \frac{P(c) * P(d/c)}{P(d)} \tag{1}$$

We assign the class $c^* = \arg max_c$ P(c/d) to be the actual sentiment of the document. We can train this classifier on the data set and calculate the P(c/d) = $\prod_{i=1}^{m} P(f_i/c)^{n_i(d)}$, where f_i is the frequency of the i^{th} word and $n_i(d)$ is the number of times the word occurs in various documents. We also perform smoothing for estimation.

2.3 Entropy Maximisation Method

The Maximum Entropy (MaxEnt) classifier is closely related to a Naive Bayes classifier, except that, rather than allowing each feature to be independent, the model uses search-based optimization to find weights for the features that maximize the likelihood of the training data. The idea behind MaxEnt classifiers is that we should prefer the most uniform models that satisfy any given constraint. MaxEnt models are feature based models. In this method we estimate

$$P_{ME}(c/d) = \frac{1}{Z(d)} e^{\sum_i \lambda_{i,c} F_{i,c}(d,c)} \tag{2}$$

where $P_{ME}(c/d)$ is the probability that a class c occurs for a given document d and Z(d) is a normalisation function and $F_{i,c}$ is a feature-class function. This method does not make any assumptions about the relationships between the features unlike NB algo. $\lambda_{i,c}$ is the feature-weight parameter indicating the weight for a feature f_i in class c. The advantages of using this technique is accuracy, consistency, ability to handle huge amount of data and high flexibility.

2.4 Support Vector Machine Method

Support Vector Machines(SVM) have been used in many classification problems and have shown to perform high accuracies in most of the cases esp in document classification problems. In two category case, they basically find a maximum-margin hyperplane. This corresponds to a constrained optimization problem whose solution can be written as

$$w = \sum_j \alpha_j c_j d_j, \alpha_j \geq 0 \tag{3}$$

Here α_i are obtained by solving the dual maximisation problem and d_j are the *support vectors*. The drawback with this method is the training time as it takes lot of time to train due to its high complexity.

2.5 Recursive Neural Network Method

This is a recent work based on the advancements of Deep learning and neural networks models. This uses the Sentiment Treebank and applies recursive neural network models to that to determine the sentiment of the sentence/document. Without going into the mathematical details, this method represents each word in a d-dimensional vector space by random sampling and applies *softmax* function to them for finding the most apporpriate class. For more details please see [5].

2.6 PMI-IR Method

This is an unsuoervised learning technique which uses the POS tags of words and finds PMI between them for finding the similarity to a positive reference word(”good”) or to a negative word(”not”). Phrases containing *Adjectives* and *Adverbs* are extracted using POS tag. The PMI can be calculated as

$$PMI(word_1, word_2) = \log \frac{p(word_1, word_2)}{p(word_1) * p(word_2)} \tag{4}$$

The semantic orientation of phrase can be calculated as

$$SO(phrase) = PMI(phrase, ”Excellent”) - PMI(phrase, ”poor”) \tag{5}$$

2.7 Random Forest

It is an ensemble technique by constructing a multitude of decision trees at training time and outputting the mode of the classes of indicidual trees. It corrects the habit of overfitting the data by decision trees in the training set. Generally *bootstrap aggregation* is used for training random forests. Given a training set X = $x_1, x_2, x_3, x_4 ... x_n$ with the output as Y = $y_1, y_2, y_3, y_4 ... y_n$, bagging repeatedly selects B times a random sample with replacement from the training data set and fits trees to this sample. After training, predictions for the test data can be made by averaging the output from the individual trees constructed during the training time.

$$f_o ut = 1/B \sum_{i=1}^{B} f_i() \tag{6}$$

In random forest„ random subset of features are taken each time during the bootstraping models. This process called *feature bagging* helps in taking the best features to construct the decision tree.

As these are experiments that we have performed, we will explain more about them later in the final report.

3. OBSERVATIONS

3.1 Constructing the feature vectors

One important part of almost all our techniques was to construct the feature vectors. Extracting good features considerably improve the accuracy of the classifier. In our case of documents, we used the *top-k* most occuring words as the features of our training matrix and also removed stop words and lemmatized the data. Also, we found that very large number of features(the training matrix was sparse) poses problems in classifiers such as svm[7] and also due to the huge size of training data, the time taken to train classifiers such as svm was very large. We veried the number of features for different classifiers and their accuracies are presented in the table above.

3.2 What we Observed!

- In MaxEntropy, the use of occurence for words gave better accuracy as compared to the frequency usage.

- Of all the classifiers, SVM gave the best results. Also random forest and naive baye's gave almost similar accuracy.

Results using *nltk*

Type of classifier	Accuracy	Precision	Recall
Naive Bayes(Objective)	0.838	0.868995	0.796
Naive Bayes(Subjective)	-	0.811808	0.88
SVM(kernel = rbf)	-	0.82	0.81
SVM(kernel = linear)	-	0.88	0.88
Max. Entropy	0.953(Itrn2)	-	-

Results using *CoreNLP RNN*

Type of Heuristic for Document Classification	Precision Positive	Recall Positive	F1 Score Positive	Precision Negative	Recall Negative	F1 Score Negative
Full Average	0.72	0.88	0.40	0.75	0.66	0.35
Average of Top 5 and Bottom 10	0.67	0.87	0.38	0.79	0.58	0.34
Average of Top 10 and Bottom 10	0.669	0.87	0.37	0.95	0.57	0.36
Average of Bottom 10	0.6382	0.90	0.37	0.83	0.49	0.31

Results using *SVM*

Group Of files	Precision Positive	Recall Positive	Precision Negative	Recall Negative
800-1000	0.81	0.75	0.73	0.83
600-800	0.82	0.80	0.81	0.82
400-600	0.76	0.75	0.73	0.76
200-400	0.82	0.74	0.76	0.84
0-200	0.81	0.75	0.73	0.83

Results using *Random Forest*

Group Of files	Precision Positive	Recall Positive	Precision Negative	Recall Negative
800-1000	0.78	0.76	0.78	0.79
600-800	0.75	0.78	0.77	0.77
400-600	0.77	0.75	0.76	0.79
200-400	0.75	0.66	0.70	0.78
0-200	0.79	0.81	0.80	0.79

Results using *Maximum Entropy*

Type of Model	Precision Positive	Recall Positive	Precision Negative	Recall Negative
0-1	0.66	0.98	0.96	0.48
Frequency	0.65	0.97	0.96	0.47

- The RNN gave not upto the expected result. The reason may be due to the heuristic used to calculate the overall semantic orientation of the file. We used various methods to calculate the final positiveness.

- Changing the features considerably improved the accuracy of the models. Hence parameter tuning is also an important part.

4. CONCLUSIONS

Our aim in this project is to compare the sentiment analysis techniques and if time remains, try to come up with variations in them. Hence, we have first aimed to perform experiments on various existing methods using famous works of many researchers. We will also try to look into the aspect based sentiment analysis if time remains. We thank Dr. Ashish Anand and entire CS-565 team for giving us the opportunity of working in this project.

5. REFERENCES

[1] *Deeply Moving: Deep Learning for Sentiment Analysis.* http://nlp.stanford.edu/sentiment/.

[2] *Movie Review Data.* https://www.cs.cornell.edu/people/pabo/movie-review-data/.

[3] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity. In *Proceedings of ACL*, pages 271–278, 2004.

[4] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86, 2002.

[5] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D.

Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October 2013. Association for Computational Linguistics.

[6] P. D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[7] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In *NIPS*, volume 12, pages 668–674. Citeseer, 2000.