**Novel Sentiment Analysis with RST Features**

Bharadwaj Tanikella |  | Rohit Pathak

1. **Introduction**

For the individual part of the project, we decided to experiment with RST tree features for sentiment analysis on movie reviews. Our implementation involved modifications of the Bag of Words model (e.g. using only the words in the top-most nucleus of the tree) as well as experimenting with features on top of the Bag of Words model. In this paper we look at how these different approaches affect sentiment analysis.

1. **Dataset**

We used a random sample of an existing dataset of about 50k imdb movie reviews available from the Stanford Artificial Intelligence Lab.

Dataset Type # of POS # of NEG

Train 2000 2000

Stanford Large Movie Data Set Test 400 400

**Table 1.** Statistics of the Dataset used (Stanford AI Lab).

We randomly selected 4000 reviews to train our model and tested on a random set of 800 reviews [400 ‘POS’, 400 ‘NEG’].

**2.1 Document Preprocessing**

Several preprocessing steps were conducted on the dataset to comply with the RST Parser. The ‘.txt’ files were changed to ‘.edus’ files to allow the RST parser to parse through the documents. We considered each sentence to be one edu. Further dependency parsing and POS tagging of the document were used in the implementation of our model; these documents were produced through an existing library (Stanford-Core-NLP), which is available at http://nlp.stanford.edu/software/corenlp.shtml. When the files are processed by Stanford-Core-NLP library a set of ‘.out’ files are created, which were processed again by a script (whatever3.py) to get a set of ‘.headwords’ and ‘.pos’ files for the parser to read.

1. **Classifiers**

We used nltk’s Naïve Bayes classifier and scikit-learn’s Linear Support Vector Classifier (SVC) to classify our documents. The main reason behind this was to study how the classifiers perform on different sizes of the feature space.

1. **Feature Set**
   1. **Baseline Features**

The baseline features we have chosen are a bag of words representation of the document and the length of the document. Through these features we tried to establish a standard by which we wanted to compare and contrast the additional features. The Bag of Words model and the length of the document provide the simplest strategies to obtain the sentiment of a document.

* 1. **Nucleus Bag of Words**

Once the RST parse tree is applied to the document, the most important edu (the nucleus at the topmost level) is selected. If there are multiple nodes in the parsed tree the node with the property as the nucleus below the root is selected. We then decompose the text of this node into BOW features. The intuition behind this approach was that the edus in this nucleus contain all the words relevant to the sentiment of the entire document, so with just these words, our BOW features will be more accurate.

* 1. **Relation Bag of Words**

This feature is essentially an extension of the Bag of Words model. By augmenting the Bag of Words model with the relation of the node that it is in, it can be more positive or negative. If it is consistently inside one relation several times, it can reflect polarity better than just the word count in isolation. For instance the word ‘insanity’ with a relation to ‘elaboration’, ’the characters are a piece of insanity’ have a different connotation to ‘insanity’ with a relation to ‘antithesis’, ‘Serenity now; insanity later’. Therefore we included this particular feature in hopes of providing an understanding of the word in a better sense.

* 1. **Nucleus Polarity**

The same node selected from the Nucleus Bag of Words is applied. An external resource with a dictionary of positive and negative words is compiled and the nucleus node is iterated through and the weight of the polarity is calculated. Through this we assume that the nucleus is representative of the entire document and can help classifying the document with the calculated polarity of the node.

<nucleus\_polarity> : [polarity]

where polarity = len(poswords in nucleus) – len(negwords in nucleus)

* 1. **Level Centric Polarity Relations**

The polarity to the relations and the depth of the tree is tracked in these features. The intuition behind this feature is to keep track of the polarity of the relations at each level. For example if an antithesis relation is more and more negative towards the top of the tree, the document as a whole must be positive. Or if the evidence relation is consistently positive, the document will be positive. The algorithm to decide the polarity of the node keeps track of the depth the node is in and the relation it maintains. Also by specifying the polarity of the relation, it can be understood that the learning model will keep track of the probability of the polarity given a relation and the depth of the node.

* 1. **Relation Parts of Speech**

Relational Parts of Speech counter keeps track of the number of times a particular parts of speech tag has appeared in a particular relation. Through this feature we try to incorporate the importance of POS tag in a particular relation. This feature is used to make sure certain sentiment portraying POS tags have different weights and the addition of certain POS tags such as an adjective, adverb or an interjection help classify a document. However our null-hypothesis of having the POS count for relation was not successful as the feature is very naïve and the lack of structure in the training data made this feature weak for classification.

* 1. **Experimental Features**

A number of multiple experimental features were included to observe the pattern for classification. Some of the features tested are: number of relations in a document, weight of the relations in a document, word-polarity BOW and synset sentiment score for relations. These features do not produce a positive effect on the classification as they have sparse information with respect to Sentiment Analysis.

Num of Relations: , Weight of Relations:

Word-Polarity BOW: & Sentiment Score:

1. **Evaluation**
   1. **Evaluation Metrics**

Accuracy (the percentage of the number of documents correctly classified) is calculated with addition to the Precision, recall and F-measure. Precision and recall are calculated by utilizing the False Positive, False Negative, True Positive and True Negative. (small font cells show feature performance in isolation without the BOW)

* 1. **Evaluation Results**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Feature | Naïve Bayes | | | | Support Vector Classifier | | | |
|  | Acc | F-1 | P | R | Acc | F-1 | P | R |
| Baseline | 0.821 | 0.815 | 0.84 | 0.73 | 0.813 | 0.816 | 0.81 | 0.84 |
| Nucleus BOW | 0.59 | 0.600 | 0.55 | 0.60 | 0.563 | 0.598 | 0.55 | 0.65 |
| Baseline+NBOW. | 0.823 | 0.821 | 0.83 | 0.84 | 0.830 | 0.831 | 0.81 | 0.84 |
| **Relation BOW** | **0.800** | **0.789** | **0.83** | **0.74** | **0.802** | **0.811** | **0.77** | **0.85** |
| **Baseline+RBOW.** | **0.815** | **0.841** | **0.82** | **0.85** | **0.821** | **0.824** | **0.80** | **0.84** |
| Depth BOW | 0.813 | 0.805 | 0.84 | 0.77 | 0.813 | 0.817 | 0.80 | 0.83 |
| Baseline+DBOW. | 0.812 | 0.806 | 0.83 | 0.78 | 0.809 | 0.813 | 0.79 | 0.83 |
| Nucleus Polarity | 0.557 | 0.445 | 0.59 | 0.35 | 0.572 | 0.501 | 0.60 | 0.43 |
| Baseline+NucPol. | 0.818 | 0.812 | 0.84 | 0.78 | 0.817 | 0.818 | 0.81 | 0.82 |
| Level Rel. Polarity | 0.645 | 0.634 | 0.65 | 0.61 | 0.657 | 0.680 | 0.63 | 0.73 |
| Baseline+LevRPol.. | 0.820 | 0.812 | 0.84 | 0.78 | 0.813 | 0.814 | 0.81 | 0.82 |
| RelationPOS | 0.542 | 0.420 | 0.57 | 0.33 | 0.556 | 0.353 | 0.65 | 0.24 |
| Baseline+RelPOS.. | 0.817 | 0.813 | 0.83 | 0.79 | 0.812 | 0.813 | 0.81 | 0.81 |
| All RST Features | 0.798 | 0.791 | 0.82 | 0.76 | 0.816 | 0.818 | 0.81 | 0.83 |

**Table 2.** Accuracy and F-measure scores for features and combinations, excluding experiment features.

The results provided show that there is not a big variation in the scores for both the classifiers. The table above is organized in a format where the single feature score is recorded by using both the classifiers and then scores are recorded with the combination of Baseline features. Overall the Relation Bag of Words performs very well individually and combined with the Baseline features with an f-score of 0.841 for NB and 0.824 for SVC respectively.

1. **Conclusion**

We have presented structural and lexical RST Tree features to improve the popular Bag of Word model while classifying sentiment of a document. These features range from adaptation of BOW model to structural information of the RST Parse Tree. We proposed multiple features over the Baseline Bag of Words model to improve the accuracies of classification. We had success with a few of the structural features however the best features were modifications of BOW.

Furthermore, using sentences as edus was a limitation. Had we used a proper edu generator on our documents, we would have had a better tree to work with and all our tree features would have been more meaningful.

The slight improvements in some cases, however, suggest that structural features indeed can improve sentiment analysis.

**Works Cited:**

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). [Learning Word Vectors for Sentiment Analysis.](http://ai.stanford.edu/~amaas/papers/wvSent_acl2011.pdf) *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*

Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. [The Stanford CoreNLP Natural Language Processing Toolkit](http://nlp.stanford.edu/pubs/StanfordCoreNlp2014.pdf). In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

http://aclweb.org/anthology/I/I11/I11-1038.pdf