# Classifying News for Topic and Sentiment

Thomas Atwood, Miguel de Lascurain, Jack Smith

Final Paper

CS 229

Dr. Andrew Ng

December 13, 2013

# Classifying News for Topic and Sentiment

Around the globe, there are more than 6,500 daily newspapers selling close to 400 million copies every day. Additionally, there are blogs, micro blogs, periodicals, magazines, fanzines, etc. How can we make sense of all this information? How can we classify it and aggregate it so that we can perform quantitative analysis?

This project explores one possible answer to these questions: automating the classification of news articles by sentiment and topic. Our vision is to create the capability to track how sentiment on a topic has evolved over time, how different news outlets cover the same topic, and, in the limit, to be able to predict future behavior through sentiment trends.

## Data

For our project, we have considered two databases: an Economic News database for training and the New York Times 1987-2007 dataset. Rebecca Weiss, Ph.D. candidate in Communication at Stanford, provided us with this data.

**Economic News database:** Our initial dataset is comprised of 10,237 sentences selected by Rebecca Weiss and Richard Socher, Ph.D. candidate in Computer Science at Stanford, from economic/financial news in American newspapers. Weiss and Socher submitted the data to Amazon's Mechanical Turk, where workers labeled each word, phrase, and sentence for sentiment on a 25-point scale from 0 (very bad) to 1 (very good). At least three workers labeled each token, and the dataset stored the average of the three workers.

**New York Times 1987-2007 dataset:** The second dataset consists of every article published by the New York Times between 1987 and 2007 – approximately 700,000 articles. The dataset is well structured – each article was hand-labeled for classification by topic. However, the data does not have labels for sentiment.

## Approach

We approached the problem using incrementally powerful algorithms. This allowed us to compare the performance as well as to comment on the advantages and disadvantages of each algorithm.

**Naïve Bayes:** We applied Naïve Bayes by creating a word vocabulary using the median words in our training set. That is, we chose an upper and a lower bound for the number of appearances of a word in our training set and created the vocabulary with the words between those bounds. We optimized the word bounds by choosing the bounds that minimized the testing error.

Additionally, we assumed that the sentiment could be divided into two classifications: good sentiment and bad sentiment. For this, we considered all the articles in our News Article database that had a sentiment score (determined by the Mechanical Turk workers) less than or equal to 0.5 to have a bad sentiment and the rest to have a good sentiment.

Given this vocabulary and possible categories, we applied a straightforward multinomial event model under the Naïve Bayes assumption.

**Bag of Words Classifier:** We recognized that by creating the vocabulary the way we were doing for Naïve Bayes, we were considering words that didn't have any emotional charge *a priori*. For example, in our vocabulary we included the word "business" which is a neutral word by itself. It takes its emotional charge from adjectives and context.

We tried a Bag of Words Classifier for our second approach. We selected the words for our new vocabulary by identifying the words with the highest emotional charge in our Economic News database. Once this vocabulary was created, we applied a multinomial event model under the Naïve Bayes assumption. Additionally, we scaled the probabilities to reflect the magnitudes of the sentiment of each word. For example, the word "rat-infested" had a higher weight than the word "inappropriate".

**Support Vector Machine (SVM) with Principal Component Analysis (PCA) for context:** As mentioned in Socher, et al. (2013), classifiers that only take single words into consideration are generally bounded in terms of prediction accuracy. With this in mind, we sought a method to represent words mathematically in order to access other machine learning algorithms for classification. We chose to use windowing, which builds upon the theory that a word is given meaning by its context, i.e. the words that surround it. To accomplish this, we ran a parser on the New York Times database that counted the number of appearances of each word in our 8,000-word vocabulary in a given word's "window". For the most part, each window was comprised of the five words on either side of the given word. We added padding to windows that fell at the beginning or end of sentences.

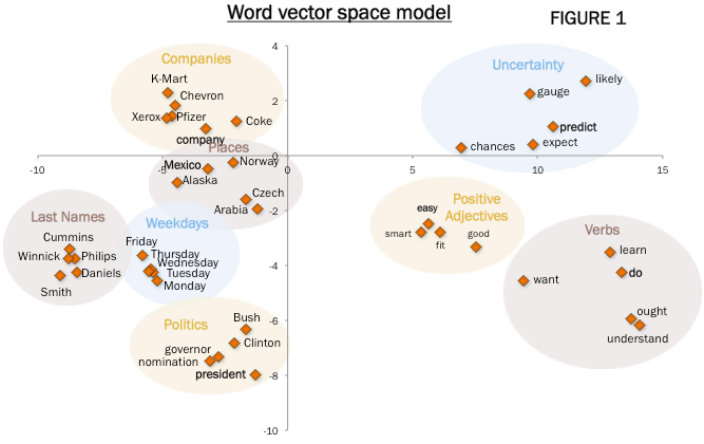


FIGURE 1

We now had an 8,000 by 8,000 matrix containing word counts, and sought to reduce the dimensions in order to use a SVM. To accomplish this, we ran PCA on our matrix, and found that using the eight principal components, we captured 80% of the variance in our matrix. By analyzing a few clusters of words, we found that PCA was very

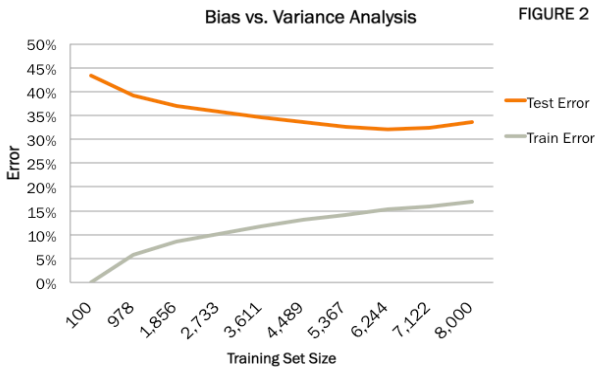
successful at locating synonyms and closely related words. For instance, the closest words in terms of Euclidean distance to "Friday" were "Monday", "Tuesday", "Wednesday", and "Thursday". Figure 1 illustrates other representative clusters.

Finally, we trained an SVM on our labeled dataset by representing each article as an array of words, where words were represented by their four principal components.

## Results

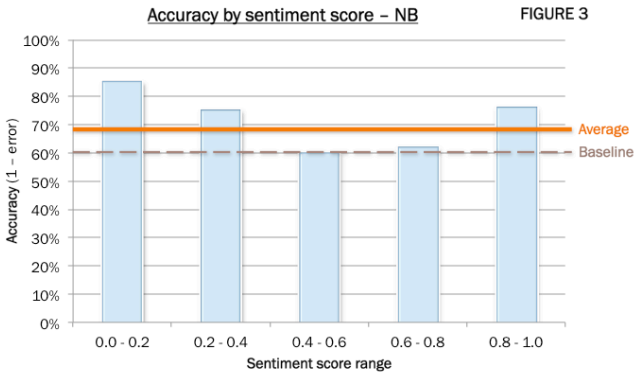
By running a Naïve Bayes algorithm on the Economic News database, we achieved 67.9% prediction accuracy (+7.9% above baseline). By examining how the test error related to the training error using different training set sizes, we found that there was a bias issue since both errors were above our desired error threshold of 15% (Figure 2).

FIGURE 2



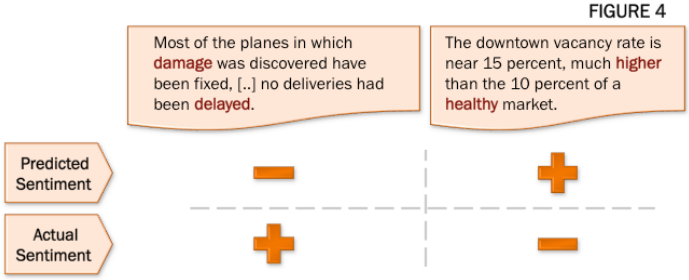

We also looked at accuracy by sentiment score (Figure 3). As we expected, we found that using Naïve Bayes, we are very accurate at predicting "very bad" (scored 0 to 0.2) and "very good" (scored 0.8 to 1) news articles but our predictions are not accurate for "neutral" articles. This result also holds true for Bag of Words.

FIGURE 3




Paradoxically, as we used more powerful models, our accuracy decreased. We achieved an accuracy of 67% (+7% above baseline) using the Bag of Words approach. We found that this algorithm, despite being in theory more defensible, performs worse than Naïve Bayes. We found two reasons that could explain this performance. The first is that the words with higher sentiment charge do not appear as frequently in our training set as we would have liked. Hence, some of the predictions were based on very few data points. This contrasts greatly with our Naïve Bayes approach since we had a lower bound constraint on the number of appearances that a word should have for it to be considered. The second reason why our algorithm performed

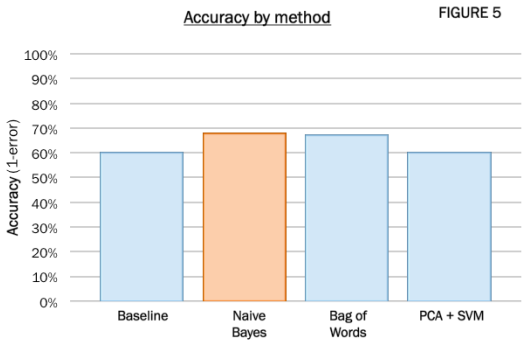
FIGURE 4



badly is that in order to determine the sentiment of a sentence, it is not enough to consider the sentiment of the words. As exemplified in Figure 4, the context in which the words are used can greatly affect the sentiment of the sentence as a whole.

Finally, we achieved an accuracy of 60% (equal to baseline) using the SVM with PCA approach. By using PCA, we were very successful at giving a measure to the words and thus determining which words were similar to other words in terms of their context. However, by running these "contextualized" words through our SVM, we did not achieve a high degree of accuracy because knowing the context in which each word appears is not the same as knowing how one word changes the context of the rest within a sentence. In other words, we were successful at encoding words but we were unsuccessful at encoding sentences. Figure 5 shows a summary of the results.

FIGURE 5




## Conclusions

In summary, we approached a very well known problem with known high difficulty without prior knowledge of the more sophisticated tools that have been developed to deal with problems in this space, and were unsuccessful at predicting sentiment on topics in news articles on a large scale. However, we were successful at measuring words in an abstract space in such a way that we could determine which words are clustered together. Additionally, we achieved an 8% increase in accuracy of predicting the sentiment of a news article. Finally, through research, brainstorming, trial-and-error, and exploration of natural language processing (NLP), we learned an enormous amount both about NLP and about Machine Learning.

## References

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Stanford, CA.