LING 406 Term Project : Sentiment Classifier

Vismayak Mohanarajan

Introduction

People regularly voice their thoughts and opinions on the internet. These very thoughts and opinions are what drive the entirety of the consumer market; they are worth millions to companies. This data could be harvested for a variety of purposes unique to each market niche; the challenge lies in understanding and categorizing these vast swaths of data. The amount of text available to companies is simply too vast to be analyzed manually; therefore, we must train computers to comprehensively read these texts for the purposes of sentiment analysis and the like.

Language itself is a very subtle thing to interpret. It takes years even for humans to begin to understand all the subtilties and nuances, before we can use it practically and on a daily basis for thorough communication. In order for a computer to understand text, we need to give it elementary building blocks for it to begin.

One of the simplest things for a human to do is to understand the tone of a given text through the connotation of the words used. The challenge is to incorporate this sentiment extraction with a computational platform. Once perfected, sentiment classification can prove to be a very useful tool in a variety of fields ranging from assisting speech in artificial intelligence to extracting the feelings associated with certain types of data, such as reviews.

In this report, we observe an algorithm which is first used on movie reviews to classify them as positive or negative. The algorithm is also later used on Yelp restaurant reviews. The paper will detail the steps taken to extract the data and the features that were added to the algorithm and its impact on the accuracy. We also observe a Naïve Bayes classifier on both the datasets.
The dataset containing the movie reviews is found at http://www.cs.cornell.edu/people/pabo/movie- review-data/, Pang et al (2002). The Yelp reviews were created by John Hall, in LING 406 Spring 2016.

Problem Definition

Sentiment analysis in terms of computational linguistics is the process by which a computer can identify and categorize feelings expressed in the form of text to perceive the overall attitude of the writer to that particular topic. In our case, we try to identify if the movie review is good or bad or if a particular review of a restaurant would have a rating above 3.5 stars or not.

To do the classification we need to use both natural language processing and machine learning . We need to use NLP to parse through the text and remove “noise” which could hamper the training of the classifier. We also use a lexicon containing words that are positive and negative. This Opinion Lexicon is available at thttps://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon. This lexicon helps remove words that are not useful in training the data such as “film” or “restaurant” but keep words that matter like “good” and “bad”.

We then created a model trained on the data left and this model was tested on other reviews and it's accuracy was measured. Throughout the projects, various features were added or removed with the goal of increasing the accuracy.

Previous Work

K. Yessenov and S. Misailovic [1] analyzed movie reviews from the social networking platform Digg by extracting text using a bag-of-words model, WordNet synonyms, and negation handling, then proceeded to analyze the extracted text. The bag-of-words model contains only adjectives and adverbs in order to isolate the words that would best portray the sentiments. WordNet is a lexical dictionary in Python, and negation handling allows the computer to differentiate between "good" and "not good", for example. Their effects on accuracy of four machine learning methods, namely, Naive Bayes, Decision Trees, Maximum-Entropy, and K-Means clustering, were measured. Their results prove that a simple bag-of-words model is capable of performing well. However, classification does not prove to be promising when the corpus tested on varies very differently from the model it was trained on.

Pang, Lee, and Vaithyanathan [2] semantically analyzed movie reviews using Naïve Bayes, Maximum Entropy, and Support Vector Machines, in addition to other feature selection methods such as stop-word removal and Part-of-Speech tagging. With regards to their work, Naïve Bayes performed the worst and SVMs performed the best, but only marginally. Part-of-Speech tagging did not prove to be a very useful feature selection method, especially in comparison to unigrams.

Approach

Baseline System: Bag-of-Words

A bag-of-words is a model that extracts words from text which are the features in the model. It also contains the number of occurrences of each word. This is a rough equivalent of a unigram model. Two bags-of-words were utilized, one trained on positive reviews and the other trained on negative reviews. The algorithm used to classify the testing data extracted the words from the review and compared the count of words located in either bag-of-words. In order to create an advanced baseline system, the following features were tweaked to improve the accuracy:

1) Punctuation Removal:
   We parsed through the text and removed all punctuation. We believed this could potentially increase the accuracy of the classifier, as we did not want our bag-of-words model to treat words followed by punctuation to be treated differently from words not followed by punctuation.

2) Uniform Capitalization:
   For the Yelp reviews, we did not want the bags-of-words to treat the same words with different capitalization as different words. Therefore, we standardized all the text to be lowercase.

3) Stopword Removal:
   Stopwords are words that do not hold any significance in deciding the sentiment of the text. They should be treated as noise and removed from the text. Since this is a standard NLP procedure, there exists a standard list of stopwords in the NLTK library.

4) Ratio of Training versus Testing Data:
   It is a standard procedure to use a 70-30 ratio of training versus testing data while creating a model. This ratio was used while classifying the movie reviews. However, the ratio was tweaked while classifying the Yelp reviews to have a higher percentage of training data, as there was not an equal distribution of positive and negative reviews.

5) Lexicons:
   We used the opinion lexicon mentioned previously in the hope that it would remove words which do not reflect the sentiment of the text. However, as we observed later, this feature did not aid in the classification of the movie reviews and severely decreased the accuracy of the restaurant review.

Analysis

Baseline System

We will use a leave-one-out approach to measure the importance of each feature.

But before that let us measure the performance of the basic baseline system for the movie and Yelp reviews:

Data being Classified	Accuracy	Precision	Recall
Movie Reviews	65.67	64.24	70.67
Yelp Reviews	69.33	75.34	87.70

We will now measure the performance of the Improved Baseline System for the movie and Yelp reviews:

Data being Classified	Accuracy	Precision	Recall
Movie Reviews	70.16	68.97	73.33
Yelp Reviews	76.25	80.68	89.21

We will now observe the impact of removing each feature on the Accuracy, Precision and Recall of the Movie Review:

Feature	Accuracy	Precision	Recall
Punctuation Removal	69.67	68.44	73.0
Stopwords removal	70.16	68.97	73.33
Lexicon Usage	68.66	64.89	81.33

It can be observed from the above data that the feature which has the maximum weightage on both the accuracy and precision is the lexicon. This is probably because it helps on eliminating a large amount of noise.

Now let us do the same for the Yelp reviews:

Feature	Accuracy	Precision	Recall
Punctuation Removal	49.02	59.76	100
Stopwords removal	76.25	80.68	89.21
Uniform Case	78.74	83.3	89.85

As mention previously, the lexicon usage actually lowered the accuracy greatly and wasn't used in the final baseline system, but these are the results when **added**:

Feature	Accuracy	Precision	Recall
Lexicon Usage	61.55	69.35	98.38

It is observed for the Yelp reviews that the punctuation removal has the most weight; failure to add this feature decreased the accuracy of the advanced baseline by nearly 30%. However, it must be kept in mind that this huge decrease in accuracy is also because of the influence of the other features.

Also, though the uniform case helps increase the accuracy, it would prove imprudent in the long run not to have this feature, as this is a very important and standard NLP procedure.

Naïve Bayes Classifier:

Using the Naïve Bayes Classifier on the data with the advanced bag-of-words model, we get the following results:

Data being Classified	Accuracy	Precision	Recall
Movie Reviews	71.16	71.97	69.33
Yelp Reviews	78.053	83.69	99.13

We perform a similar leave-one approach to get the significance of each feature:

Feature	Accuracy	Precision	Recall
Punctuation Removal	70.83	71.31	69.67
Stopwords removal	67.67	81.54	45.67
Lexicon Usage	70.67	63.59	99.67

We observe for the Naïve Bayes the stopwords removal have the maximum weightage.
This is probably because the elimination of large useless information keeps the probability of each word more uniform.

Doing the leave-one approach on the Yelp reviews once again, we get the following results:

Feature	Accuracy	Precision	Recall
Punctuation Removal	49.02	59.76	100
Stopwords removal	72.76	78.70	99.67
Uniform Case	78.05	83.69	99.13

Punctuation removal is once again the most important for Yelp reviews. It is to be observed in both the classifiers most of the features have nearly the same significance.

Discussion and Conclusion

It is necessary to preprocess text before analyzing it, as failing to do so significantly reduces the accuracy of the results. The features utilized for preprocessing, as discussed in the section titled "Approach", had varying levels of effectiveness. They are as follows:

1) Punctuation Removal:
   This was a critical step in increasing the accuracy of our classifier. It had a minimal impact in the classification of the movie reviews, but was much more impactful in the classification of the Yelp reviews. Seeing as this is a standard NLP procedure, we had no doubts as to the importance of this feature.

2) Uniform Capitalization:
   In the case of the Yelp reviews, utilization of this feature seemed to hinder the classification accuracy. However, in a bag-of-words model, it is important to standardize the capitalization of the letters

3) Stopword Removal:
   The removal of stopwords do not have a significant impact on the classification accuracy; this is due to the fact that the stopwords will be classified as both positive and negative in the bag-of-words model. However, it should be kept in mind that the removal of stopwords would decrease the size of the vocabulary for the bag-of-words and would thus speed up the computation.

4) Ratio of Training versus Testing Data:
   This was not an important parameter to be tweaked for the classification of the movie reviews. However, it had a significant impact in the classification of Yelp reviews, as the Yelp reviews were not evenly distributed between positive and negative reviews. This lead to a discrepancy while splitting the training and testing data. If this project were to be repeated, a better approach would involve reorganizing the data rather than altering the ratio of training versus testing data.

5) Lexicons:
   Though the usage of lexicons is important in decreasing the volume of words for the bag-of-words model, it is not particularly useful in the classification of the Yelp reviews, as the reviews could have words with positive or negative connotations specific to the type of review. For example, a general lexicon might not include the word "delicious", which is specific to restaurant reviews.

Further improvements to the bag-of-words model would include stem removal (removal of suffixes such as -ing), utilizing a lexicon specific to the dataset, and utilization of a convolutional neural network.

References

1. Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

2. Yessenov, Kuat, and Sasa Misailovic. "Sentiment Analysis of Movie Review Comments" Massachusetts Institute of Technology, Spring 2009. http://people.csail.mit.edu/kuat/courses/6.863/report.pdf

3. Browniee, Jason. "How to Develop a Deep Learning Bag-of-Words Model for Predicting Movie Review Sentiment." *Machine Learning Mastery*, 14 Feb. 2018, machinelearningmastery.com/deep-learning-bag-of-words-model-sentiment-analysis/