Sentiment Analysis using Word Vectors

ATJ and NSB

University of Massachussets, Amherst

Abstract

For the task of sentiment analysis, a document can be represented in multiple forms - as a bag of words of unigram frequencies, as a bag of words of unigram absence/presence indicators, as a bag of words of bigram frequencies, or as word vectors generated from Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), word2vec or doc2vec. The selection of a particular document representation is crucial to yield good accuracies in sentiment analysis. In this project we evaluate these different document representations over three widely available datasets - Review Polarity v2.0 dataset and the Subjectivity dataset provided by [Pang and Lee, 2004], and the Large Movie Review Dataset v1.0 made available by [Maas et al., 2011] using a multitude of classifiers - Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Trees, Random Forests, Support Vector Machines and Ensemble classifiers. The results we obtained align with those obtained by [Maas et al., 2011]. We find that word vector representations using word2vec and doc2vec are the best document representations for sentiment analysis as they capture both semantic and sentiment information.

1 Introduction

Sentiment Analysis is the task of extracting the favorable or unfavorable inclination of a person towards a subject or topic. For example, we could analyze twitter feeds to predict or assess the stock market trends, the favorability of a presidential nominee during elections, or how good a movie is based on the volume of the tweets and their content. It can be used to get user opinions on the products the user purchased based on the star ratings and reviews submitted through online portals like amazon.com which can be used by product manufacturers to identify how well their product is received by the majority of users. This kind of feedback is invaluable and very crucial in many cases. *How can we represent a document so that we achieve high accuracies in sentiment analyis? What information would this representation capture?*

For the purpose of sentiment analysis, one could view a document as simply a “bag of words” in which the position of words does not matter. In addition, we could assume that the words are conditionally independent given the class to which they belong. This is implicitly errant because it ignores the possibility of negated reviews like negated positive reviews which become negative and negated negative which might become positive. In addition, it ignores the semantics of the words. For example, it cannot capture the relationship between “powerful”, “strong” and “Paris”. Vector based models are able to capture the relational structure of the lexicon. Vector based models can represent words as distance or angle between word vectors in a high-dimensional space. This gives us ability to evaluate word similarities. Different word vectors capture different information about the documents. Some word vectors capture semantic information of the words, while other representations capture both semantic and sentiment information. The particular document representation selected for sentiment analysis has a huge impact on the accuracy of the classifier.

In this project, we evaluate some of these different document representations - bag of words using unigram frequencies, bag of words using unigram absenece/presence indicators, bag of words using bigrams, word vectors obtained from Latent Dirichlet Allocation (LDA) [Blei et al., 2003], Latent Semantic Allocation (LSA), word2vec [Le and Mikolov, 2014] and doc2vec using a multitude of classifiers such as Naive Bayes, Logistic Regression, K-Nearest Neighbors, Decision Trees, Random Forests, Support Vector Machines and Ensemble classifiers. We evaluate these different document representations by training and testing the classifiers over three widely available datasets - Review Polarity v2.0 dataset and the Subjectivity dataset provided by [Pang and Lee, 2004], and the Large Movie Review Dataset v1.0 made available by [Maas et al., 2011]. The accuracies that [Maas et al., 2011] and others [Sadeghian and Sharafat,] reported on the three datasets are listed in Table 1. Our experiments similarly reveal that we can achieve high accuracies in sentiment analysis using word vector representations obtained from word2vec and doc2vec. This is possible because these word vector representations capture both semantic as well as sentiment information.

Features	**PL04**	**IMDB Dataset**	**Subjectivity**
Bag of Words	85.45	87.80	87.77
LDA	66.70	67.42	66.65
LSA	84.55	83.96	82.82
[Maas et al., 2011]'s Semantic Only	87.10	87.30	86.65
[Maas et al., 2011]'s Full	84.65	87.44	86.19
[Maas et al., 2011]'s Full + Bag of Words (bnc)	87.85	88.33	88.45
Bag of Words SVM [Pang and Lee, 2004]	87.15	N/A	90.00

Table 1: Classification accuracies reported by [Maas et al., 2011] and others for different word representations.

2 Related Work

explain what other approaches have been to the problem. Cite specific instances of previous work. You must cite at least 10 relevant research papers, and describe them and how they relate to your work. It may be convenient to structure this as a related work or literature review section.

3 Data

We employ the Review Polarity v2.0 dataset and the Subjectivity dataset provided by ([Pang and Lee, 2004]) which are available online at http://www.cs.cornell.edu/People/pabo/movie-review-data/. We also use the Large Movie Review Dataset v1.0 made available by [Maas et al., 2011].

The Review polarity dataset v2.0 consists of 2,000 movie reviews that were obtained from IMDb archives and processed. The folders "pos" or "neg" under which these movie review files exist determine the true sentiment label. The subjectivity dataset contains two files, one containing 5000 objective sentences and the other containing 5000 subjective sentences. The subjective sentences were obtained by processing movie reviews from Rotten Tomatoes and the objective sentences were obtained by processing plot summaries for movies from IMDb.

The Large Movie Review Dataset v1.0 contains 50,000 movie reviews split evenly into 25k train and 25k test sets. The movie review files are under two folders, "test" and "train", each containing a "pos" and a "neg" folder having reviews of the corresponding label. The overall distribution of labels is balanced (25k pos and 25k neg). The train and test sets contain a disjoint set of movies, so no significant performance is obtained by memorizing movie-unique terms and their associated labels.

4 Method

We built different pipelines with stages of preprocessing, feature extraction, classifier training, hyper parameter selection and evaluation using python as the programming language and code libraries from sci-kit learn, nltk, gensim.

4.1 Preprocessing

We built three tokenizers - Simple Tokenizer, Advanced Tokenizer and Bigram Tokenizer. The Simple Tokenizer simply splits documents on spaces and down cases the tokens. The Advanced Tokenizer is built using nltk's *TreebankWordTokenizer* which tokenizes text as in the Penn Treebank. The *TreebankWordTokenizer* splits standard contractions such as "don't" to "do n't" and "they'll" to "they 'll", treats most punctuation characters as separate tokens, splits off commas and single quotes when followed by whitespace and separates periods that appear at the end of line. We also remove stop words using nltk's *stopwords* corpus for the English language. Finally, we down case the tokens. The Bigram Tokenizer uses ntlk's *TreebankWordTokenizer* to tokenize the text. The tokens are then grouped into bigrams using ntlk's *bigram* package.

4.2 Feature Extraction

5 Results

Describe the experiments you ran and identify your baseline method(s). Include the results you achieved with the various methods you are comparing making. This section will probably also include some figures that succinctly summarize your results. Analyze your results (including your models). If you did exploratory analysis or a significant amount of feature engineering, your analysis may merit its own section. After reading this section (and your dataset and methods), an interested reader should be able to duplicate your experiments and results.

6 Discussion and Future Work

discuss any implications of your analysis for the problem as a whole, and what are the next steps for future work. Any other concluding remarks should go here.

References

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[Le and Mikolov, 2014] Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

[Maas et al., 2011] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

[Pang and Lee, 2004] Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics.

[Sadeghian and Sharafat,] Sadeghian, A. and Sharafat, A. R. Bag of words meets bags of popcorn.