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 will be employing the sentiment 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 will also use the Large Movie Review Dataset v1.0 made available by [Maas et al., 2011]. The polarity dataset v2.0 ([Pang and Lee, 2004]) consists of 2,000 positive and negative labelled movie reviews, and the subjectivity dataset ([Pang and Lee, 2004]) contains 5000 objective and 5000 subjective sentences. The Large Movie Dataset v1.0 [Maas et al., 2011]contains 50,000 reviews split evenly into 25k train and 25k test sets.

4 Method

Describe your approach to handling the problem. This should should include any models you used and any modeling assumptions you made. If youve developed new models for this project, you may even want to split a description/analysis of your models into its own section.

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.

Sentiment Analysis is a text categorization problem in Natural Language Processing (NLP) that is often addressed by using a multinomial naive Bayes classifier or a Support Vector Machine (SVM) classifier. An SVM classifier tries to minimize a hinge loss function and combined with a regularization parameter, for a binary classification, it learns the equation of a separating line the classes. All the data points falling on one side of the line are classified with that label.

It learns the weights or coefficients of the line through training. Other classifiers which share a similar decision boundary are Gaussian naive Bayes , Logistic Regression etc. KNN is a non-parametric classifier that stores every training sample in memory, gets better with increase in the number of data samples and suffers from the famous curse of dimensionality. Decision trees and random forests which use multitudes of them averaged build a binary tree with each node as a feature value decision point. Neural networks provide good improvement in the accuracies of NLP classifiers, most of which is through the usage of word vectors as features over the regular one hot encoding or bag of words. LDA and LSA are two most commonly used techniques to generate word vectors.

Latent Dirichlet Allocation(LDA; [Blei et al., 2003]) is a probabilistic document model that assumes each document is a mixture of latent topics. This models the topics directly than the words and is less efficient. The result is a wordtopic matrix in which the rows are taken to represent word meanings or word vectors. Latent Semantic Analysis (LSA) is a vector space model which explicitly learns semantic word vectors by applying singular value decomposition (SVD) to factor a termdocument co-occurrence matrix. While LSA captures the semantic relationships between words, it doesnt capture the sentimental proximities or differences. For example, “wonderful”, “delightful” and “awful”, “ghastly” all seem to have similar semantic proximity or representation.

In this project, we hope to evaluate different word and document vector representations for sentiment and subjectivity analysis. Sentiment determines polarity of a document while subjectivity says if the author is subjective or objective in his opinion. We plan to compare document vectors such as bag of words (unigram) count frequency, bag of words(unigram) presence/absence representation, bigram word dependencies, and word vectors learnt from LDA, LSA, word2vec, neural network , assign strong_pos/weak_pos / strong_neg / weak_neg / neutral tag to each word in a document and using them with classifiers such as Naive Bayes, KNN, Decision Trees, SVM, Ensemble/Boosting classifiers over the three data sets we mention below. We hope to build different pipelines with stages of preprocessing, feature extraction, classifier training, hyper parameter selection and come up with the best that performs across all the data sets. We plan to reach the existing accuracy levels of classification using the respective vector representations and try to improve if possible.

Several preprocessing steps for documents like turning all words to lower case, removing punctuation marks, removal of stop words, removal of high frequency words and stemming will be tried. All the code will be implemented in python itself. Using code libraries like sci-kit learn, nltk, regex, gensims word2vec where necessary.

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.