Sentiment Analysis for Movie Reviews

Jiachen Ning Linhong Li Shiqi Xiao Xiangting Chen
Carnegie Mellon University
Pittsburgh, PA 15213

1 Introduction

Sentiment Analysis is the task of classifying text documents by their *sentiment*, or the overall opinion towards the subject matter (6). In practice, sentiment analysis is useful for companies and policymakers to monitor large scale public opinions on specific products and/or policies, and thus extended applications of sentiment analysis can be found in security, medical, finance, and entertainment domains (5).

The goal of this project is to build an interpretable sentiment classifier based not only on text content information, but also on lexical polarity information and unobserved topic information that could be derived from the text, using one of the most analyzed dataset for sentiment analysis – the IMDb Dataset of movie reviews. The dataset contains a collection of 50,000 reviews posted on the Internet Movie Database. Standard preprocessing ensured that the dataset is balanced, containing the same number of positive and negative reviews, where positive reviews had a score greater than or equal to 7 and negative reviews had a score less than or equal to 4 (4).

Our primary task is to output a binary classification of an audience's attitude towards a movie based on his or her online review. To do that, we will experiment with both content-based and topic-based input representations of the movie reviews on a variety of classification models. The performance of our model will be evaluated based on classification accuracy.

2 Literature Review

Literature review suggested that successful sentiment analysis requires finding (1) the most representative text embeddings and (2) the best performing classification scheme.

For the first requirement, it is a standard practice to tokenize the text into sequences of n words, called n-gram representations, with bigram and trigram representations being the most widely used and showing superior results than unigram tokens (8). To vectorize a tokenized text, common methods include counting the token occurrences (i.e., Bag-of-Word, or BoW model) and assigning a term frequency (TF) or term frequency-inverse document frequency (TF-IDF) score to each token in the corpus dictionary, both of which reflect how important a token is to a document. However, the BoW and TF-IDF models are based solely on the text content. The joint sentiment/topic model proposed by Lin and He suggested that unsupervised topic modeling techniques such as LDA could potentially lead to more informative feature text representation for sentiment analysis (3). Besides topic information, it was shown that weighting words by their lexical polarity and part of speech improves classification accuracy.

In terms of the second requirement for a successful sentiment analysis, several common choices of classifiers presented in previous works include Naïve Bayes (NB), Support Vector Machine(SVM), and (Deep) Neural Networks (NN) (2; 8) . For the IMDB dataset, the current best classifier is ULMFiT, which is a LSTM-based model proposed by Howard and Ruder in 2018 (2); SVM with NB features (NBSVM) proposed by Wang and Mannings showed superior performance among non-NN classifiers (8).

Preprint. Work in progress.

3 Methods

Currently, most existing works mainly focused on testing various classifiers on content-based feature vectors. Thus, we explored different combinations of feature vectorization (BoW, TF, and TF-IDF) and the two most effective classifiers used in sentiment analysis, NB and SVM. For the SVM model, we used both linear and RBF kernels. Note that we fixed the tokenization to be unigram and bigram since it tends to give better performance (6). All baseline models are constructed, trained, and tested using the machine learning package, scikit-learn (7).

4 Preliminary Results

The combination of different vectorization methods and classifiers, as well as their respective testing prediction accuracies are given in Table 1 and Figure 1.

Tokenization	Vectorization	NB	SVM-Linear	SVM-RBF
Unigram	BoW	0.81968	0.83212	-
Uni + Bi	BoW	0.84272	0.87504	0.63148
Uni + Bi + Tri	BoW	0.84476	0.83212	-
Unigram	TF	0.84308	0.87560	-
Uni + Bi	TF	0.85372	0.88060	0.67400
Uni + Bi + Tri	TF	0.85336	0.88092	-
Unigram	TF-IDF	0.83128	0.87240	-
Uni + Bi	TF-IDF	0.85476	0.88584	0.65448
Uni + Bi + Tri	TF-IDF	0.85724	0.88532	-

Table 1: Baseline Models and Test Accuracy

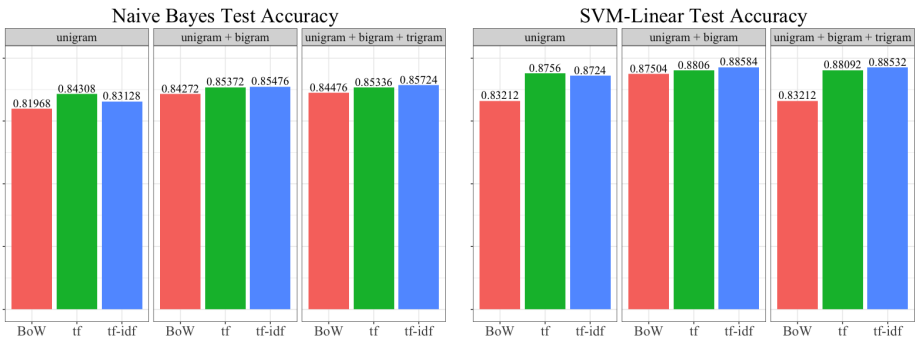

Figure 1: Baseline Models and Test Accuracy

5 Evaluation of Preliminary Work

In terms of classifiers, we can see that when using a BoW model, the best test accuracy was given by NB; when using either TF or TF-IDF as feature vectors, the best test accuracy was achieved by linear SVM. The reason why SVM that using RBF kernel performed poorly is because we didn't perform cross-validation during training to select the optimal gamma parameter and used the default value given by the package instead.

In terms of feature vectorization methods, we observed that both TF and TF-IDF give similar results and outperformed BoW model.

6 Future Work

In our future work, we hope to first explore feature vectors with richer information such as lexical polarity and part of speech tags. Furthermore, we are interested in finding out how topic modelling could be incorporated into and potentially augment the task of sentiment analysis. Through topic modelling, each document could be learned as a mixture of latent topics in an unsupervised fashion. Inspired by the technique of boosting, random decision forest, and previous works done by Ficamos and Liu (1), we think it is reasonable to propose that training classifiers that are specialized in analyzing subsets of reviews discussing similar topics and making a final prediction based on voting would be more effective than training a universal classifier that generalize across the entire dataset. Other than the potential improvement in classification accuracy, since a latent topic is simply a collection of words that could be interpreted and further analyzed, our proposed model also has the merit of being more interpretable in comparison to end-to-end deep learning based models that often learn convoluted feature representations.

7 Teammates and Work Division

We expect projects done in a group to be more substantial than projects done individually. You should outline what everybody in your group will do and by when each task should be complete.

March 11th to March 24th (Feature Vector Improvement)

• More research on ways to improve feature vectors, with a focus on lexical polarity and part of speech tag (Xiangting Chen, Shiqi Xiao)

• Test the new feature vectors on the classifier (Linhong Li, Jiachen Ning)

• Based on the results from the new feature vectors, make modifications accordingly (All)

March 25th to April 7th (Topic Modelling)

• Explore topic modelling to identify latent topics (Linhong Li, Jiachen Ning)

• Train the classifier and find explanations on why or why not the topic modelling works (Xiangting Chen, Shiqi Xiao)

• Start writing final report (All)

April 8th to April 16th (Poster Preparation)

• Prepare content for the poster (Xiangting Chen, Linhong Li)

• Draw result graphs for the poster (Jiachen Ning, Shiqi Xiao)
• Create the poster and identify the most important finding from our research project. (All)
• Prepare the information we need to convey during the talk that will complement our poster. (All)
April 17th to May 1st (Final Report)
• Summarize previous results and write the final report (All) • Final modification of final report based on the feedbacks from the poster session. (All)

References

[1] Ficamos, P., Liu, Y. (2016). A topic based approach for sentiment analysis on Twitter data. *International Journal of Advanced Computer Science and Applications, 7(12)*, 201-205.

[2] Howard, J., Ruder, S. (2018). Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146.

[3] Lin, C., He, Y. (2009, November). Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management* (pp. 375-384). ACM.

[4] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., Potts, C. (2011, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 142-150). Association for Computational Linguistics.

[5] Mäntylä, M. V., Graziotin, D., Kuutila, M. (2018). The evolution of sentiment analysis: A review of research topics, venues, and top cited papers. *Computer Science Review, 27*, 16-32.

[6] Pang, B., Lee, L., Vaithyanathan, S. (2002, July). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10* (pp. 79-86). Association for Computational Linguistics.

[7] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research 12* (pp. 2825-2830).

[8] Wang, S., Manning, C. D. (2012, July). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2* (pp. 90-94). Association for Computational Linguistics.