IMDB Movie Review Sentiment Classification

Eric Li
Department of Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213
eli1@andrew.cmu.edu

Varun Gudibanda
Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213
vgudiban@andrew.cmu.edu

1 Introduction and Problem Setup

We chose the Text Classification for IMDB Movie Review Dataset[4] with the goal of classifying a given review as positive or negative. The inputs will be singular movie reviews as strings and the outputs will be either 0 (negative review) or 1 (positive). To do this, we will be investigating various natural language processing techniques to transform our inputs to be vectors of counts as opposed to strings. We plan on analyzing our results using model accuracy on a held out test set of movie reviews.

2 Data

We are using the IMDB Movie Review Dataset constructed by Andrew Mass et. al. for their paper "Learning Word Vectors for Sentiment Analysis." The dataset is comprised of 50,000 movie reviews along with rating labels on a 1 to 10 scale. A review is labeled if it has a score of ≤ 4 and positive if it has a score ≥ 7. In the dataset, exactly half the reviews are positive and half are negative and no neutral reviews are present (i.e. those with a score of 5 or 6). No more than 30 reviews are allowed for any given movie and the train and test sets contain a disjoint set of movies. Although the dataset contains legitimate movie reviews, it may be biased due to the fact that they are all from IMDB.

This dataset could also be useful for the problem of auto-generating movie reviews using a GAN. In addition, the dataset also contains 50,000 unlabeled movie reviews that can be used for unsupervised learning.

3 Background/Literature

A large amount of research has already been done on sentiment analysis of this dataset. A paper from UCSD explores five different basic classifiers along with eight different feature models [3]. The authors concluded that a logistic regression applied on unigram/bigram mixed modelling leads to the lowest test classification error rate. Furthermore, logistic regression appears to be the best classifier in general, beating out other classifiers by at least 3-5 percent.

Beyond the baseline methods explored by the UCSD paper, various other researchers used this data set as a method of exploring new language modelling techniques. Below are a few examples of the research related to our work:

- Doc2Vec and Word2Vec feature extraction in conjunction with a logistic regression yields strong results. Combining these methods with TF-IDF leads to even better results. [5]
- Learning word vectors for semantic term-document information and sentiment [1] uses both unsupervised and supervised learning methods for feature extraction, leading to a more rich description of the review.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

- Multi-class movie review classification using a low-rank recursive neural tensor network (RNTN) has success similar to that of a high-rank RNTN. [6]
- XLNet is the current most state of the art unsupervised language representation learning method that can be used to pretrain neural networks. It combines autoregressive and autoencoding language modeling techniques to surpass the performance of either one individually. [2]

Given the extent of exploration that has already been performed, we would like to explore how combinations of these methods affect results.

4 Methods/Model

Our baseline model consists of a bag of words feature extraction along with a logistic regression classifier. We compare lemmatized and non-lemmatized movie reviews using this model.

Bag of words construction involved counting the number of occurrences of each word in our data set and sorting the n most frequent words. For each movie review, we created a feature vector as the number of occurrences of the first n most frequent words in our bag of words. Thus, each movie review was ingested as an n-length vector consisting of counts for each word. These vectors were then fed into a logistic regression along with the labels – 0 for a negative review, 1 for a positive review.

Beyond this baseline model, we constructed a unigram-bigram mixed model of the movie reviews. To do so, we constructed a bag of words in the same manner as before and considered all the 9000 most frequent unigrams and bigrams that occured in fewer than 85% of the documents.

5 Preliminary Results

Using grid search to explore various possible values for the regularization term for logistic regression, we settled with an inverse regularization strength of 0.05. We then considered the affects of bag size on accuracy due to the possibility of overfitting.

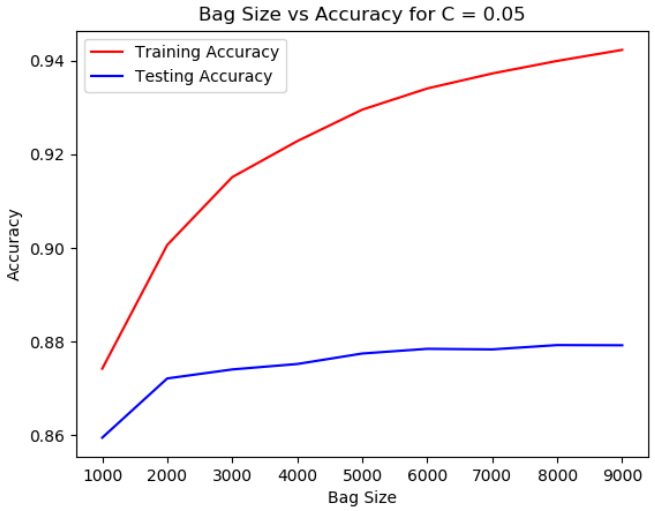

Figure 1: Accuracy versus bag size for C = 0.05 logistic regression

We then explored how lemmatization affects accuracy. This was done without performing part of speech tagging beforehand, so it was not a perfect representation of the accuracy of lemmatized bag of words logistic regression.

Finally, we explored the use of bigram/unigram mixed modelling. This was done using SKLearn and its *CountVectorizer()* function. We compared how feature capping affects accuracy.

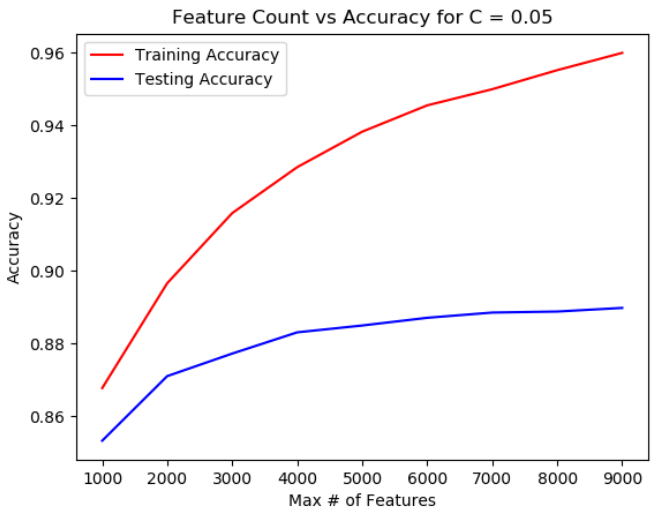

Figure 2: Accuracy versus max number of features for C = 0.05 logistic regression

6 Evaluation of Preliminary Work

Our preliminary results show that bag of words with logistic regression is quite accurate and performs similarly well with or without lemmatization. Due to the computational cost of lemmatization, we opt to ignore this preprocessing step for future work.

Unigram and bigram modelling showed increased accuracy. The maximum testing accuracy measured was 88.976% when using unigram and bigram mixed modelling as compared to a maximum test accuracy of 87.928% for a traditional bag of words model. This confirms the results of Ankit Goyal and Amey Parulekar.

7 Future Work

In order to beat our baseline model, we plan on exploring a combination of XLNet and Doc2Vec. The Doc2Vec feature extraction method offers a rich document-level semantic representation while XLNet provides extremely powerful fine-detail sentiment features. In order to combine these methods, we will use XLNet to extract features by removing the final prediction layer and input these features along with the paragraph id used in Doc2Vec into a logistic regression model.

8 Teammates and Work Division

We hope to complete all necessary coding by March 20th. In doing so, we will have time to train our model and potentially leverage AWS resources. Model evaluation will be once again based on testing accuracy, however we will have far more hyperparameters to explore and optimize.

In order to successfully implement both parts of model, we will divide the task. Eric will implement the XLNet feature extraction method and look into AWS resources if necessary while Varun will implement the Doc2Vec feature extraction method. From there, we will combine our results and perform hyperparameter optimization and model evaluation as a team.

References

[1] Andrew L. Maas et al. Learning word vectors for sentiment analysis. *Stanford University*, 2011.

[2] Zhilin Yang et al. Xlnet: Generalized autoregressive pretraining for language understanding. *Carnegie Mellon University*, 2020.

[3] Ankit Goyal and Amey Parulekar. Sentiment analysis for movie reviews. 2015.

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[5] Alejandro Pelaez, Talal Ahmed, and Mohsen Ghassemi. Sentiment analysis of imdb movie reviews. *Rutgers University*, 2015.

[6] Hadi Pouransari and Saman Ghili. Deep learning for sentiment analysis of movie reviews. *Stanford University*.