Testing the Performances of Bi-LSTM + Attention model on Sentiment Analysis on Short Movie Reviews

Jianzhe Xiao, Emily Qiao

Northwestern University

ABSTRACT

Sentiment analysis has been a critical task of natural language processing (NLP), and in this project, we focus on analyzing sentiments of short movie reviews using different models. We construct Bi-LSTM models, and group it with the Attention model. We also explore two different ways to generate sentence vectors, and compare the performance of our models with other popular NLP models. Through our experiments, we find that the BERT-based Bi-LSTM with Attention model achieves the best overall results.

Index Terms— Sentiment Analysis, NLP, Deeping Learning

1. INTRODUCTION

The prevalence of the internet greatly propagates the amount of user-generated contents. As user-generated contents usually reflect the thinkings of the authors, they contain valuable information that can be helpful to multiple stakeholders. For example, a movie producer might want to know how audiences are reacting to her latest movie by reading through the comments and reviews online. It is reasonable to argue that such an approach is not efficient enough if there are thousands of reviews available, and it is natural to think of whether it is possible to use deep learning techniques to systematically analyze such information on a large scale.

Hence, for this project, we propose to test the performance of various deep learning models on predicting sentiments of movie reviews, and we want to focus on short reviews in particular because they are usually hard to train due to the limited length. There has been a lot of research around classifying sentiments of text. In 2002, Peter Turney proposed a simple unsupervised learning algorithm [5] that can classify opinion words as positive or negative with 74% accuracy. The term sentiment analysis first appeared in a 2003 paper by Tetsuya Nasukawa and Jeonghee Yi, where they defined sentiment analysis as to determine how emotions are expressed in the text and whether these expressions indicate positive or negative opinions on the subject. In 2015, Tai et al. proposed to analyze semantic representations by a serialized LSTM model with added syntactic structure [7]. Their work achieved good results in sentence-level sentiment classification, yet we think there are areas of improvement on their work because the LSTM model only preserves past information. On the other hand, the Bi-LSTM model can preserve both past and future information, and we think this can help to increase accuracy even more. We also want to further explore whether combining models (e.g. Bi-LSTM + Attention) can further improve accuracy of labeling sentiment of short movie review. Therefore, in this project, we construct variations of Bi-LSTM models and compare their performances with other popular NLP models.

2. METHOD

2.1 Dataset and Preprocessing

We train our models on the IMDB movie sentiment dataset provided by [4]. The dataset contains 50,000 binary labeled movie reviews for training and testing, with the reviews equally divided into training and testing sets. 50,000 unlabeled data are also included for unsupervised training.

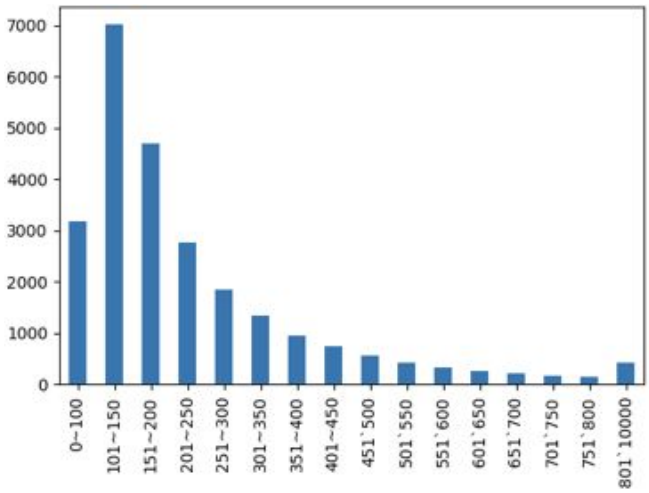

Figure 2.1 Distribution of text length

We analyze the dataset by examining the distribution of text length. As shown in figure 2.1, the majority of reviews are within the length of 250 words. As we focus on short

movie reviews for the project, we decide to use 200 as the maximum text length to filter out long reviews. We then perform data cleansing for feature construction by decapitalizing all the letters and removing punctuations and stopwords. We also remove words with low frequency (< 5) because they may be typos and do not carry significant meaning.

2.1 Experiment Design

After the preprocessing is done, we experiment with two different models to generate word embeddings that are later used as the input of our deep learning models. We first use the Word2vec model. We specify to use the Skip-gram method to train, which predicts the word based on relevant context. We choose Skip-gram over CBOW because it is good at representing rare words and has higher accuracy.

We also use the BERT model to generate sentence embeddings as it is a relatively new model and is good at resolving polysemy, so we wonder if using word embeddings from the BERT model can improve accuracy. We use the BERT base model with 12 layers. Although it is possible to use the output of any layer as the word embeddings for later use, we find that the output of the 11th layer produces the best results. If we use the output from earlier layers, the model may not be sufficiently trained; if we directly use the output from the last layer, it would be too similar to the original text.

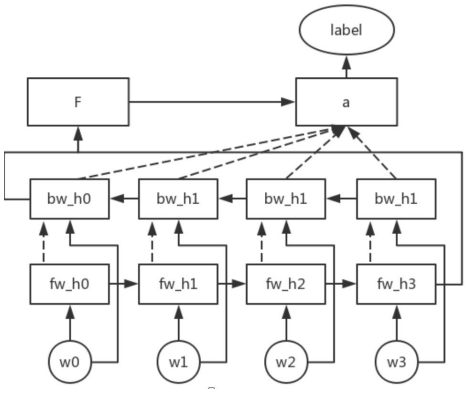

Figure 2.2 Overall model structure diagram

We then construct a Bi-LSTM model with 128 neurons for training, with each neuron defining a forward LSTM structure and a reverse LSTM structure. We then concatenate their outputs, and pass it to the next layer of the Bi-LSTM model.

To examine whether combining the Attention model with Bi-LSTM can improve accuracy, we pass the result of the last layer of the Bi-LSTM model to an Attention model. We then apply the tanh activation function to the output, multiply it with the weight vector, calculate the softmax, and pass the final result to the fully connected layer.

3. RESULT

Because we're performing binary classification, our predicting results belong to one of the following four outcomes: true positive, true negative), false positive, and false negative. Therefore, we measure performance by four metrics: Area under the ROC Curve (AUC), Accuracy (ACC), precision (PRE), and recall (REC). Ideally, we would like to maximize all of the above metrics.

We compare both our Word2Vec-based model and BERT-based model against four other commonly used deep learning NLP models such as textCNN and CharCNN, as well as a single directional LSTM model.

Model	ACC	AUC	PRE	REC
TextCNN	0.8616	0.9332	0.8726	0.8503
CharCNN	0.8379	0.9176	0.8383	0.8381
LSTM	0.8571	0.9201	0.8626	0.8120
Bi-LSTM	0.8451	0.9064	0.8975	0.8029
Bi-LSTM+ Attention	0.8762	0.9381	0.9077	0.8615

Table 3.1 Word2vec-based model training results

As shown in table 4.1, the Bi-LSTM model alone does not perform other common models, yet the Bi-LSTM+Attention model clearly outperforms all the other models in almost every aspect.

Model	ACC	AUC	PRE	REC
Bi-LSTM	0.8841	0.9232	0.9257	0.8592
Bi-LSTM+ Attention	0.9343	0.9506	0.9517	0.9239

Table 3.2 BERT-based model training results

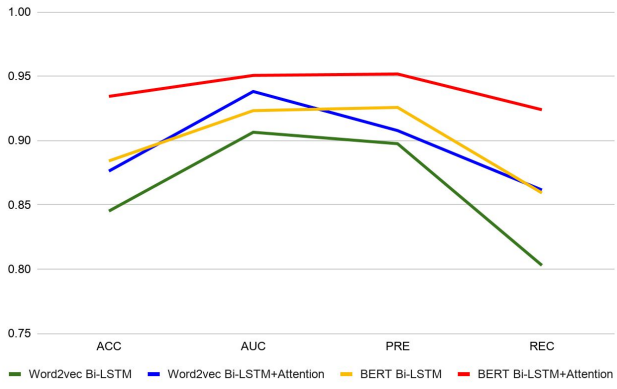

Figure 3.1 Bert-based result line chart

Our BERT-based models get even better results as compared to the Word2vec models, as shown in table 3.2 and figure 3.1.

4. DISCUSSION AND FUTURE WORK

As it is unprecedentedly easy to acquire large amounts of data, deep learning technology develops at a fast pace. Sentiment analysis, as an application of deep learning, is also gaining increasing attention and values, hence motivating us to optimize the algorithm to obtain better results.

Our work shows that the combining Bi-LSTM model with Attention model improves performance, and BERT-based Bi-LSTM model combined with the Attention model has the best overall performance when classifying sentiments on short movie reviews. We think this is because Bi-LSTM can weaken the influence of the text input sequence on results. Combining the Attention model can strengthen the influence of key information, which effectively reduces the importance of historical information to performance. And finally, the BERT model can correctly resolve polysemies in the text, which also contributes to the better performance.

We have proved the functionality and advantages of our models by running them on IMDB short reviews dataset, yet there are also a few limitations of our work. First of all, we limit the maximum text length to the average of the overall dataset, thus unable to capture the information contained in almost half of the dataset. Shall we get access to machines with higher computing power, we could train the models on the complete dataset by not limiting the length. Also, we take a relatively simple approach to pre-process the text data by decapitalization and removing punctuations and HTML tag. If time permits, we can employ more complicated pre-processing methods to get better training results. Another area of future work is to use hierarchical softmax as the output layer instead of softmax, as we choose to use softmax throughout the project for its simplicity.

5. REFERENCE

[1] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'13). Curran Associates Inc., Red Hook, NY, USA, 3111–3119. DOI: https://dl.acm.org/doi/10.5555/2999792.2999959

[2] Kim Y. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. 2014 Aug 25. DOI: https://doi.org/10.3115/v1/D14-1181

[3] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI'15). AAAI Press, 2267–2273. DOI: https://dl.acm.org/doi/10.5555/2886521.2886636.

[4] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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 (HLT '11). Association for Computational Linguistics, USA, 142–150. DOI: https://dl.acm.org/doi/10.5555/2002472.2002491

[5] Turney, P. D. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp 417-424.

[6] Nasukawa, T. and Yi, J. Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the K-CAP-03, 2nd International Conference on Knowledge Capture.

[7] Tai, K. S.; Socher, R.; Manning, C. D. Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics. pp 1556–1566.