Comparing Naive Bayes and BERT models for Classification of Textual Data

Annabelle Dion, Jonas Vinson, Miles Weberman

March, 2023

Abstract

This project involved creating a Naive Bayes model from scratch and implementing a BERT model using the PyTorch pre-trained BERT library to classify IMDB movie reviews as positive or negative. The text data was preprocessed differently for the two models, with the Bayes model using lemmatization and the BERT model using tokenization. The BERT model outperformed the Bayes model in all metrics, indicating it is a more accurate and reliable model for this task. Pre-training on an external corpus, like BERT does, can be beneficial for movie review prediction by allowing the model to learn general language representations that can be fine-tuned on a specific task. Deep learning methods tend to perform better than simpler machine learning methods (such as Naive Bayes) on complex tasks that require feature extraction. While simpler machine learning methods can be effective on easier tasks that require less computational resources. The choice between deep learning and traditional machine learning methods depends on the specific task and available resources.

1 Introduction

The initial step of the project involved creating a Naive Bayes model from scratch that utilizes the appropriate likelihood for the given features. The resulting model comprises four parameters: the word count associated to positive and negative reviews, the priors, the count of each word, and the vocabulary (i.e., all the words encountered during training). The `fit()` function was developed to determine the total number of positive and negative messages, and to set up Bernoulli priors for each (as we are performing binary classification). It also calculates the total count of each word in the training data set. Next, the `predict()` function examines each text to make a prediction. Any unseen words are skipped, and the likelihood for positive and negative reviews is calculated using Laplacian smoothing. The likelihood and prior terms are then added to the total scores, and the higher score between the positive and negative options is returned. Finally, the `evaluate()` function utilizes the count logged instances library from `tables.utils` to determine the accuracy of the model. In contrast, the BERT model was implemented using the PyTorch pre-trained BERT library. The BERT classifier was initialized with a sigmoid activation function and dropout.

In our evaluation, both models demonstrated commendable performance (refer to section 3 for detailed results). However, the pre-trained BERT model outperformed the Naive Bayes model in every evaluation metric that we analyzed. The only edge that the Naive Bayes model holds over the BERT model is its shorter training time. Specifically, the Naive Bayes model can complete training on a training set of size 2,000 in less than a minute, while the BERT model takes approximately 30 minutes to fit the data under the same training conditions.

In the 2011 article that presents the movie review dataset used in our work, the authors propose a model that captures both semantic and sentiment similarities among words. Their model consists of two components: a semantic component and a supervised sentiment component. The semantic component learns word vectors through an unsupervised probabilistic model of documents. The supervised sentiment component leverages the vector representation of words to predict the sentiment annotations of contexts in which the words are used. When applied to various tasks on the IMDB dataset, their model achieves an accuracy ranging between 85 and 90% [1].

In recent years, numerous publications have proposed modifications to BERT-based models, with a focus on simplifying and accelerating the model. Despite being more efficient, these models exhibit performance metrics comparable to the pre-trained BERT model adopted in our work. For instance, a recent publication in 2023 introduces a model named oBERTa, which employs techniques such as pruning and knowledge distillation to extend previous work and produce a faster model [2].

2 Datasets

This project uses the IMDB Reviews data from `http://ai.stanford.edu/œamaas/data/sentiment/`. The training and testing data were loaded and subsampled to prevent memory overflow, with 2000 train samples and 500 test samples selected (we discuss experiments carried out to determine sample size in section 3). The dataset was first presented in 2011 and has since served as a benchmark for developing sentiment classification algorithms. It comprises 25,000 movie reviews with high polarity, evenly split between training and testing sets. This dataset is renowned for its robustness and requires minimal cleaning, with only basic preprocessing steps necessary to prepare the data for the chosen model.

When preparing the data for the BERT model, the text and sentiment were separated into individual lists, and the text was tokenized to match the format of the BERT model training data. This required adding a ”CLS” token at the start and a ”SEP” token at the end of each entry. In addition, since each entry could contain a maximum of 512 characters, the text length was limited to 510. Each token was converted to an integer and indexed using the BERT tokenizer, and attention masks with values of 1 were added to the actual data, while padding was assigned a value of 0.

As for the Naive Bayes method, the textual data was first cleaned and tokenized, with all words converted to lowercase and html tags, numbers, and punctuation removed. The words were then lemmatized to their most basic forms.

3 Results

The winner of the IMDB Reviews classification task based on the performance metrics of precision, recall, and f1-score is BERT (as evidenced in figure 1). BERT outperforms Naive Bayes in all metrics, indicating its superiority as a more accurate and reliable model for this task. BERT achieves an accuracy score of 0.93, whereas Naive Bayes achieves an accuracy score of 0.81. These results demonstrate the effectiveness of BERT in accurately classifying IMDB reviews and its potential for broader applications in sentiment analysis. Note that the Google Colab with our implementation contains a more detailed table with performance metrics for each class.

model	**precision**	**recall**	**f1-score**
Naive Bayes	0.82	0.81	0.81
BERT	0.93	0.93	0.93

Figure 1. Performance of Naive Bayes vs BERT

For both models, recall and precision are similar, it means that the model is predicting both true positives and false positives at comparable rates. In other words, the model is identifying relevant instances correctly (true positives) and also predicting some irrelevant instances as relevant (false positives) at roughly the same rate. This suggests that the model is well-balanced in its classification performance and is not overemphasizing one type of error over the other.

One of the advantages of the Naive Bayes model over the BERT model is its simplicity, which results in significantly faster training times. Under the same training conditions, the Naive Bayes model can complete training more than 60 times faster than BERT, owing to its straightforward implementation and low computational overhead.

The results presented above were obtained using a training set with 2000 instances and a test set with 500 instances. However, conducting further experiments by varying the training set size (with increments of 500) did not result in any significant performance improvements for either model. It is noteworthy that using excessively large training sets may lead to memory overflow errors, and hence, it is crucial to ensure that the available resources can handle the computational demands of the model.

To fine-tune the pre-trained BERT model, we conducted training with various hyperparameter configurations. We discovered that dropout enhances the model's performance. For determining the optimal batch size and number of epochs, we executed a grid search with batch sizes in the range of 2 to 10, incrementing by 2, and epochs ranging from 10 to 50, incrementing by 10. The most efficient and accurate results were achieved with a batch size of 4 and 10 epochs. This is neglecting slight improvements to the model when the hyperparameter settings lead to a computationally heavy training process.

We examined the attention matrix between the words and the class tokens for some of the correctly and incorrectly predicted documents to gain an understanding of the inner workings of a BERT model. The attention matrix indicates the importance of each input token in predicting the output. By visualizing the attention matrix, which we did using a heat map representation as well as extracting some statistics such as mean and maximum values, we can understand how the model focuses on certain words or phrases in making its predictions.

Using the attention matrix data, we can identify the words the model pays more attention to when classifying documents. In the case where the model classifies this gives us valuable information on which words are important for determining the sentiment of a movie review. In the case of an incorrect classification this tells us which words were paid to much attention to.

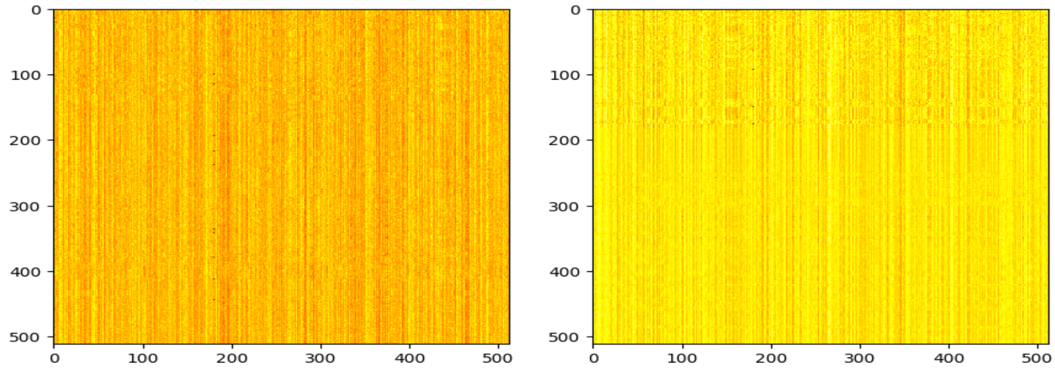

Figure 2. Attention matrix for correctly classified (left) and incorrectly classified (right)

While the available data and analysis are insufficient to reach a definitive conclusion, it appears that reviews that are classified correctly tend to have a higher number of class tokens with higher attention scores. Figure 2 displays the attention matrices as heatmaps for both a correctly and incorrectly classified review.

4 Discussion and Conclusion

Our results demonstrated that the BERT-based model significantly outperformed the traditional Naive Bayes approach in terms of accuracy and ability to generalize.

Pre-training on an external corpus, as BERT does, has proven to be beneficial for the movie review prediction task. The main advantage of pre-training lies in the ability to capture semantic and syntactic relationships present in the text, which are crucial for understanding the sentiment expressed in movie reviews. For example, the model is able to capture the fact that words like wonderful and marvelous are semantically close. Note that the method described in the 2011 publication discusses in section 1 is designed to achieve a similar effect [1]. The pre-training phase enables the model to learn the complexities of natural languages well and is a a strong foundation which we can then fine-tune on the target task, which is sentiment classification of movie reviews in our case.

The performance difference between deep learning and traditional machine learning methods can be attributed to several factors. Deep learning models, such as BERT, have a more expressive capacity due to their architecture, which enables them to capture complex patterns and relationships in the data. This is because deep learning models can automatically learn patterns and features from raw data, eliminating the need for feature engineering to produce more accurate results. Furthermore, deep neural networks have been shown empirically to be able to capture complex patterns in data, and natural languages are certainly complex and contain a high degree of ambiguity. On the other hand, traditional machine learning methods, such as the Naive Bayes classifier, rely on more simplistic assumptions, such as a prior. Therefore, they are unable to capture the complex nature of languages to the same extent as their deep learning counterparts.

Nevertheless, traditional machine learning methods still have their place and can be effective on simpler tasks where feature engineering is sufficient for high accuracy. These methods are also more interpretable and demand fewer computational resources, which can be beneficial in certain settings.

Therefore, the choice between deep learning and traditional machine learning methods depends on the specific task and the available resources. It is also interesting to note that there has been research into hybrid models which salvage the power of both deep learning methods and traditional machine learning methods [3].

In conclusion, our study highlights the power of pretrained deep learning models, such as BERT, in tackling complex natural language processing tasks like movie review sentiment prediction.

References

[1] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., 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 (pp. 142-150). Association for Computational Linguistics.
[2] Campos, D., Marques, A., Kurtz, M., Zhai, C. (2023). OBERTa: Improving Sparse Transfer Learning via improved initialization, distillation, and pruning regimes. arXiv preprint arXiv:2303.17612.
[3] Wang, Y., Bi, W., Liu, Z. (2019). A hybrid framework for text modeling with convolutional RNN. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 430-439).