Mini Project 3: Classification of Textual Data

Leo Chen 260984301 Dijian Guo 260433101 Amani Jammoul 260381641

April 14, 2023

Abstract

In this project, we compared the performance of Naive Bayes model and pretrained BERT model on textual data classification with the IMDB dataset. Their tasks were to predict, given a movie review, whether it is a positive or negative review. We found by comparing their accuracies that the pretrained BERT model outperformed the Naive Bayes model by ∼7%. We then compared the attention matrix between the words and the class tokens for some of the correctly and incorrectly predicted labels. For the correctly labeled predictions, the attention matrix showed the attention weights being higher on words that are meaningful in deciphering the movie review, like ”terrible”. In contrast, the incorrectly labeled predictions shows a more random disparity of weights throughout the attention matrix.

1 Introduction

This project aims to perform textual data classification on the IMDB dataset [1]. We first implemented a Naive Bayes model for this task. While doing so, we tried to boost our model's performance by adjusting the vocabulary size (number of individiaul words it considers) and the number of ”most frequent” words we omit from this list. We found that limiting the vocabulary size and omitting ”most frequent” words improved the model. These lower the variance and prevent overfitting. We tuned these two hyperparameters by running multiple test experiments and plotting the accuracies.

Our goal was to compare Naive Bayes' performance with a pretrained BERT model. After running experiments on both, we found that the BERT model was more accurate than Naive Bayes. One interesting and unexpected finding was that the accuracy for Naive Bayes barely dropped (83.71% → 83.2%) when using the full training and test sets (each containing 25,000 points) versus much smaller dataset sizes. Then, from using a pretrained BERT model, we obtained an accuracy of 90% by training it on 2000 train data and 500 test data, which is better than any of the Naive Bayes experiments.

Furthermore, we examined the attention matrix between the words and the class tokens from some of the correctly and incorrectly labeled documents. By choosing a transformer block and using a specific attention head, we were able to create figure 3 and 4 which give us a weight of the tokens. The first observation is that there is a clear diagonalization, suggesting that the model places importance on the neighbouring words as well as the target word itself. Then, on the left column, we can see the weight with respect to the class tokens from each tokens/words. For that column, we find that incorrectly classified reviews will have a word heat with little to no correlation to classification while correctly classified reviews will have a slight correlation between word heat and their importance to classification.

2 Datasets

All of our experiments are run on the IMDB dataset [1], which is a collection of 50,000 movie reviews containing an even number of positive and negative reviews. It only contains highly polarized reviews, meaning only ones rated $\leq 4/10$ (negative) $\geq 7/10$ (positive). Both training and testing sets contain 25,000 reviews each.

We used the TensorFlow library to acquire the data. By doing so, we are easily able to load the train and tests sets, already preprocessed, as sets where each point (*review*) is encoded as a list of word indices. A word's index reflects how often it occurs in the training set (ranked by most to least frequent).

3 Results

3.1 Evaluate Effect of Vocabulary Size on Naive Bayes

When using TensorFlow's ”load_data” function, it is possible to set the maximum vocabulary size to any value (meaning only a certain number of most frequent words are kept). We predict that removing the least frequent words will result in better performance as it would remove noise and prevent overfitting. The less frequent words are less likely to appear in other contexts so eliminating them removes irrelevant information and allows the model to generalize better.

For this experiment, we wanted to find the best vocabulary size to use. To do this, we loaded the dataset 5 times with various vocabulary sizes ([100, 1000, 10000, 20000, 50000]) and evaluated accuracies on the test set. Figure 1 shows a plot of the results. A low value makes the model too simple and doesn't fit the data well. A value higher than 10,000 starts to reduce the accuracy because of overfitting. A size of 10,000 words results in the best performance, so this value is used for the remained of the experiments.

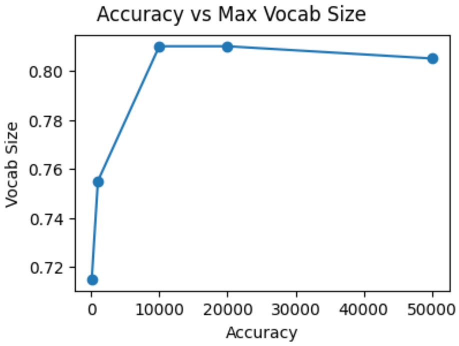

Figure 1: Accuracy vs Maximum Vocabulary Size

3.2 Evaluate Effect of Removing Most Frequent Words on Naive Bayes

Similar to how we can set the number of most frequent words we want to keep in our train and test sets when loading them, it is also possible to set a number of ”most frequent” words that we want to omit. Doing this would also improve performance because we eliminate the common words such as ”the” or ”an” which are not informative.

We performed this experiment by loading the data sets 8 times with various values of top words to skip ([0, 2, 5, 8, 10, 12, 15, 18]) and evaluated the test accuracies. Figure 2 shows a plot of the results. A value that is too low results in overfitting and a value that is too high removes important information. A value of 10 is found to be the best, so it is used for the remained of the experiments.

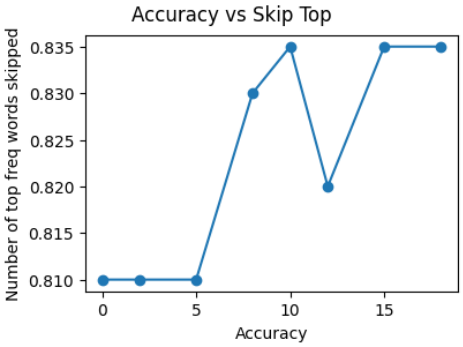

Figure 2: Accuracy vs Number of ”Most Frequent” Words to Skip

3.3 Performance of Naive Bayes vs BERT

	Naive Bayes on full test set	Naive Bayes on partial set	BERT on partial set
Accuracy	83.708%	83.2%	90%

Table 1: Test Accuracies for Naive Bayes vs BERT (partial set: 2000 train, 500 test)

The BERT model was used with 2000 training points and 500 test points. We ran two experiments here using the Naive Bayes model: one using the entire train and test sets, and another using the same sizes as BERT (2000 train and 500 test). The accuracies are shown in Table 1.

In all cases, BERT outperforms Naive Bayes. Surprisingly, Naive Bayes doesn't suffer much between training on the full set of 25000 dataset and training on the partial set of 2000. For the BERT model, we ran into memory issues when trying to train it with all 25000 dataset, hence why we chose a smaller amount which is 2000. By running it with batch size 4 and 10 epochs, we get a 90% accuracy. Since both Naive Bayes results are very similar at 83%, it doesn't seem to matter as much, but on the same amount of training data, BERT is still outperforming Naive Bayes. Similarly, BERT also does better even if Naive Bayes is trained on the entire 25000 dataset.

3.4 BERT Attention Matrix

We created attention matrices by choosing a specific attention layer and attention head. At first, we tried every single layer in order to find the heat map that showed the strongest correlation. Finally, we found that the seventh layer had the strongest correlation since it had a very strong diagonal. See figure 3 and 4 below.

As you can see in the figures above, both correctly and incorrectly labeled reviews produce a very strong diagonal suggesting that the model places high importance on the neighbouring words when generating the classification output.

Additionally, we can see a slight correlation between the heat of each word and their significance to the output when looking at the first column of the heat map for the correctly labeled reviews(figure 3). This column represents the attention the model places on that word in relation to the [CLS] token. You can easily see that the word ”movie” and ”negative have a higher heat because they are important words for the classification. This is not the case for every single correctly labeled review (see more in the ipynb file). However, when examining the first column of heat map for the incorrectly labeled reviews(figure 4), we notice that it does not follow the same pattern.

This disparity between the two figures reveals that incorrectly predicted reviews will have errors and inaccuracies in their heat map while correctly predicted reviews will have a better heat map. This is due to the fact that when the model is correctly predicting a review, the attention matrix is able to effectively highlight the important words from the review. On the other hand, when the prediction is incorrect, it might be due to the fact that the attention matrix was not able to identify the most important words for the classification. Consequently, the resulting heat map will not reflect the true importance of the words.

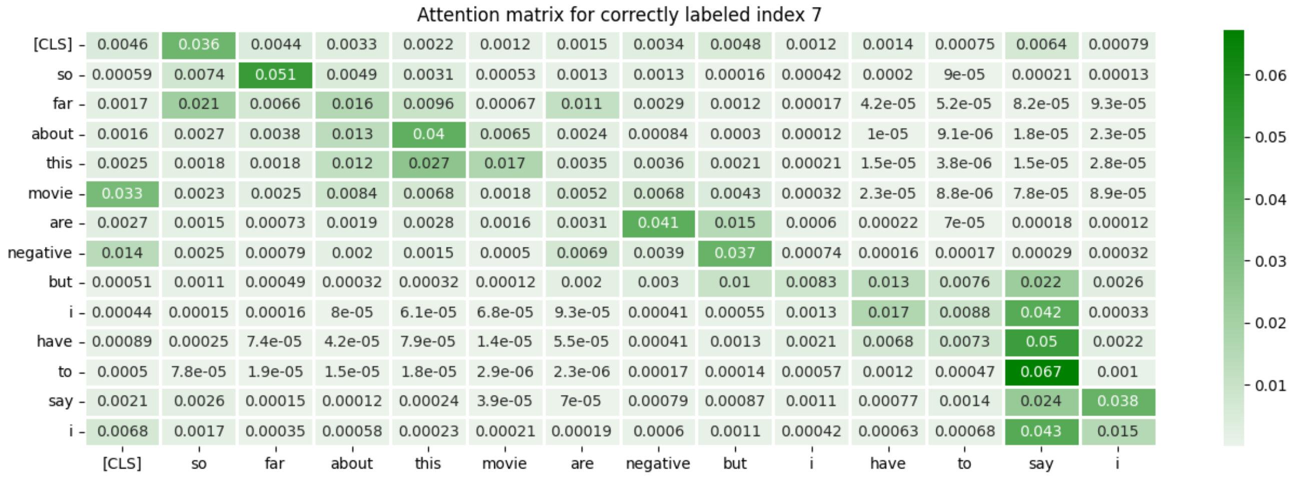

Figure 3: Attention Matrix of a Correctly Labeled Review using the Seventh Layer

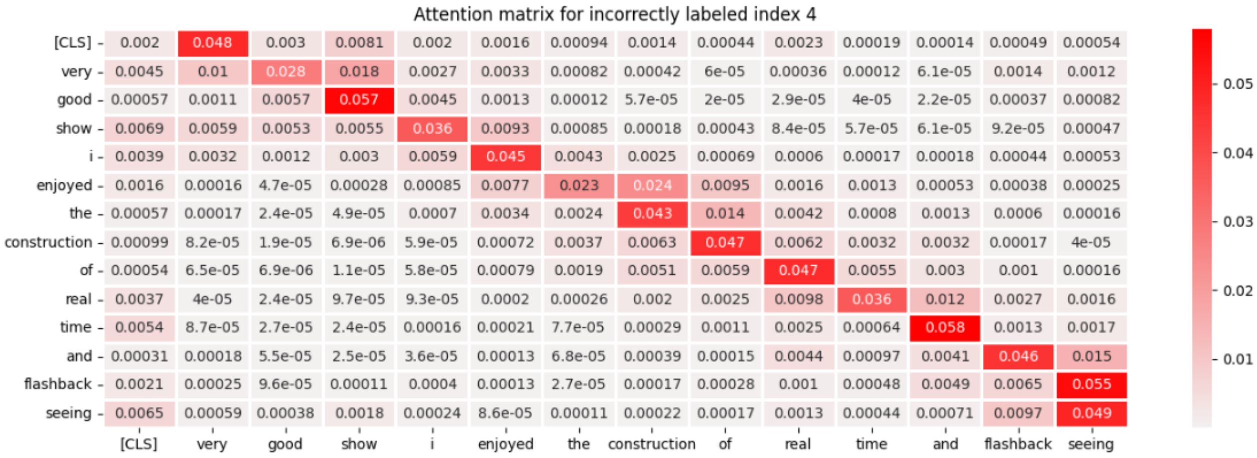

Figure 4: Attention Matrix of an Incorrectly Labeled Review using the Seventh Layer

4 Conclusion

As stated previously, BERT with pretraining outperforms Naive Bayes, even if BERT is only trained on 2000 train data and Naive Bayes is trained on 25000. Simply with 2000 train data, we managed to get a high accuracy of 90% using the pretrained BERT model.

While making our Naive Bayes model, we experimented with limiting the number of words it considers and omitting frequent words which don't add value to prevent overfitting. Our results show that a vocabulary size of 10,000 words and removing the top 10 most frequent words optimizes our model's performance.

Our results seem to show that pretraining increases performance of the BERT model, especially when compared to the Naive Bayes model in the movie review prediction task. Using a pretrained model implies using one which has already been training an a large external corpus of text data. We therefore think that pretraining is especially helpful in this kind of task because the model learns semantic relationships between words, which allows it to capture language nuances. Then, when using the model for a specific task, such as this one (classifying positive and negative movie reviews), the parameters, which are already weighted in the right direction, simply need to be fine-tuned based on the given trained data. The fitting would then be easier and faster, and the predictions would be more accurate.

From our results, we conclude that deep learning is generally much more flexible and versatile than traditional machine learning methods, since it can be used in different scenarios and can be pretrained. Pretrained models are advantageous because they are trained on large amounts of data, allowing them to capture more complex, syntactic relationships between words which traditional models may not. However, traditional machine learning methods can still be very efficient tools and can perform well. As we saw from our experiments, the Naive Bayes had only about a 7% less accuracy than the BERT. Traditional models are also faster to train, which we noticed while running our experiments. In addition, traditional models like Naive Bayes can be trained on small amounts of data and still maintain high performance, while pretrained models require large amounts of data. Indeed, as mentioned, the accuracy of Naive Bayes after fitting on 2,000 data points versus 25,000 data points was less than 0.5%.

For future investigation, we wonder what the results would be like if there was no pretraining with BERT and how would it compare. Additionally, our examination of the heat maps created using the attention matrices demonstrated that the model puts importance on neighboring words when computing classification for both correctly and incorrectly labeled reviews. Moreover, we observed a correlation between the heat of each word and their significance to the classification task in the [CLS] token column for some correctly labeled reviews. However, the heat maps for incorrectly labeled reviews do not follow the pattern. We concluded that this discrepancy was due to the attention matrix not being able to identify the most important words for classification properly in incorrectly labeled reviews. Therefore, we can see that our BERT model cannot perfectly classify the reviews because it does not accurately identify the most significant words for classification.

5 Statement of Contribution

In this project, Amani worked on everything pertaining to the Naives Bayes model and its experiments. Dijian and Leo worked together on the BERT model and its experiments along with everything related to attention matrix comparisons. Finally, all 3 members worked together to complete the written report.

6 Bibliography

[1] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).*