COMP 551: MiniProject 3

Lucas Bennett, Bjørn Christensen, Megan Ng

Abstract

In this project we investigated the performance of two linear classification models on an IMDB movie review classification task. The two models of interest were a Naive Bayes Model and a pretrained BERT model. Both models performed the same classification task in which the input were several movie review texts and the output was a "good" or "bad" label for each of the input texts. We found that deep learning machine learning techniques, such as BERT, outperform traditional machine learning techniques, such as Naive Bayes, with the trade-off of longer training times and higher computational power requirements.

1. Introduction

In this project we implemented Naive-Bayes Model from scratch, loaded a sklearn Logistic Regression Model, and fine-tuned a BERT Model. The Naive Bayes Model implementation was based off of class slides, while the Logistic Regression Model and BERT Model implementation was based off of a kaggle tutorial which specifically demonstrated testing on an IMDB dataset [1]. All models were tested on an IMDB movie review dataset in which the task was to classify a set of movie reviews into either "good" or "bad" sentiment category. Past studies that aimed to compare the performances of traditional vs deep learning machine learning techniques arrived at the same main conclusions from our experiments in the present project [2]. The main conclusion, in agreement with past literature, was that deep learning machine learning techniques, such as BERT, outperform traditional machine learning techniques, such as Naive Bayes, with the trade-off of longer training times and higher computational power requirements. In addition, we also investigated the attention matrices for a few correctly and incorrectly classified movie reviews. We found that for correctly classified texts, high attention weight values were assigned to words that appeared to be relevant to the sentiment of the text. In contrast, for incorrectly classified texts attention weights were either homogeneously distributed or high value weights were wrongly assigned to words irrelevant to the overall sentiment of the text.

2. Datasets

In order to pre-process our data we decided that removing common stop words and any incorrectly tokenized words would provide the best results when training our model. The incorrectly tokenized words would appear when a larger word was broken up into multiple parts. They were noticeable in the data because they all shared a common form, `##` followed by a sub string of the initial word. For example the word banality could be tokenized in two parts, ban and `##`ality. In addition to the stop words and tokenizing errors, there were also instances of line breaks being inserted which we did not deem necessary for our model to learn. These appeared with the form <, br, /, >. For all instances where these errors or undesirable tokens occurred we removed them along with a number of stop words derived using the python library NLTK such as "is" or "this".

For the BERT model we used the IMDB dataset which gave us equal parts positive and negative reviews, 12,500 of each. When running experiments, especially with the fine-tuning method, we would need to take partitions of the data in order to manage time complexity, and as such we worked with two main train-test splits, 2500-500 and 6000-1500. We randomly sampled from the entire sets leading to an uneven mix of positive and negative reviews in our model training. When working with the Naive-Bayes model we accessed the files directly which not only allowed us to compare positive and negative sentiments, but the exact review grade as well. For our model, because the testing data is only sampled from reviews with scores ranging

from 1-4 and 7-10, we ignored any non-polarizing reviews with a score of 5 or 6 in our training data. We also noticed that the reviews in the training data were biased towards scores of 1 or 10, with each of these two categories having much higher frequencies than any other. The test scores followed a similar trend and so we deemed it unnecessary to even out the class distribution of our input data for training the model.

Score	1	2	3	4	7	8	9	10
Occurrence	5100	2284	2420	2696	2496	3009	2263	4732

Table 1: Table comparing frequency of review categories in training data

3. Results

3.1. Task 1

We compared the performance of the Naive Bayes Model, Logistic Regression, and fine-tuned BERT Models, on the IMDB Reviews classification task. We fine-tuned to find our best training/testing splits, and observed that, with limited processing power, a 2500-500 split provided the strongest results in the shortest time. Testing data was used to estimate all reported accuracies. As seen in the results table below, the fine-tuned BERT model performed the best.

	Naive Bayes Model	Logistic Regression Model	Fine-Tuned BERT Model
Accuracy	85%	86%	93%

Table 2: Table comparing performance of Naive Bayes Model and BERT Models

To access the generalizability of the Naive Bayes model, we performed cross-validation testing. Model accuracy was evaluated on five separate 80:20 training-validation splits of the given training data set. The validation sets shared no common observations between one another. The average accuracy of the model across all splits was found to be 84.8%, with the standard deviation in prediction accuracy reported at 1%. The accuracy of the model, along with its limited variance implies a small generalized error.

We were interested in what effect the strength of the prior has on model performance. All accounts of the Naive Bayes so far have incorporated Laplace smoothing. To find the relationship between the hyper-parameter and model performance, we accessed the model under identical conditions at eight values of alphas. With a purely Bernoulli prior ($\alpha = 0$), the model was severely hindered, demonstrating an accuracy of 60%. The introduction of a weak prior immediately increased model performance, though it remained non-optimized. Model accuracy increased sigmoidally with increasing values of alphas, until plummeting beyond an optimal value. It was found Laplace smoothing ($\alpha = 1$) provided optimal model performance.

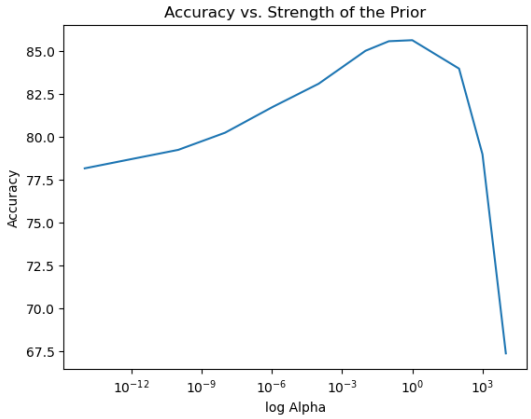

Figure 1: Relationship between the strength of the prior and model performance.

3.2. Task 2

We struggled to view the attention matrix given our choice of BERT model implementation. However, based on the definition of the attention matrix and meaning of the individual weight values, we were able to make well justified predictions of what the attention matrix would look like for correctly and incorrectly classified texts. When comparing the attention matrices of correctly vs incorrectly classified texts, we predicted to observe differences in terms of which words, or columns and rows, had high attention weight values. For correctly classified texts, we expect that high attention weight values would be assigned to words that appeared to be relevant to the sentiment of the text. This would mean that the model correctly captured which words were important in determining the text's overall sentiment. In contrast, for incorrectly classified texts, we predict that low attention weight values would be assigned to sentiment-relevant words. This would mean that the model failed to recognize these words to be important in classifying the text's overall sentiment and wrongly attended to irrelevant words. See figure below for a simple visualization of the attention weight pattern just described.

Correctly Classified						Incorrectly Classified					
	This	**movie**	**was**	**great**	**!**		**This**	**movie**	**was**	**not**	**good**
This	0.3	0.2	0.1	0.4	0.0	**This**	0.2	0.1	0.2	0.2	0.3
movie	0.2	0.2	0.2	0.4	0.0	**movie**	0.1	0.1	0.3	0.3	0.2
was	0.1	0.2	0.4	0.2	0.1	**was**	0.3	0.2	0.1	0.1	0.3
great	0.3	0.3	0.1	0.3	0.0	**not**	0.3	0.3	0.2	0.1	0.1
!	0.0	0.0	0.2	0.2	0.6	**good**	0.2	0.3	0.2	0.2	0.1

Table 3: Attention matrices for a simple movie review to demonstrate observed weight value patterns. The attention weights in the correctly classified review reveal that the model is focusing on the important words "great" and "!" while ignoring the less important words "this", "movie", and "was". In contrast, the attention weights in the incorrectly classified review show that the model is not attending to the negation "not" and is focusing more on "movie" and "was" than on "good".

3.3. Question 1

Q: Is pretraining on an external corpus (like BERT does) good for the movie review prediction task? What do you think pretraining does that might help with this task in particular?

A: From the accuracy results of our experiments, pretraining on an external corpus, in our case pytorch pretrained BERT Model, performs well on the movie review prediction task. We are able to make a few well supported hypotheses as to how pretraining may help with increasing accuracy in the IMDB movie review task. First, the pretraining allows the BERT model to gain representation learning in which it is able to capture the meaning of words in different contexts. In contrast, the Naive Bayes Model uses a bag-of-words representation of the input data that treats each word as independent and does not take into account the context of its appearance. In addition, with a pretraining that occurs on a large corpus of text, the BERT model is able to grasp general patterns of language which can then be fine-tuned to more specific tasks such as the present IMDB movie review classification task. Whereas, the Naive Bayes Model has to be trained from scratch for each task, is not able to execute such fine-tuning, and must rely on the features that were present in the training data.

3.4. Question 2

Q: What conclusions can you make about the performance difference between deep learning and traditional machine learning methods?

A: After observing the training process and the results of our experiments, we can make a few conclusions regarding the performance difference between deep learning (BERT) and traditional (Naive Bayes) machine learning methods. First, deep learning machine learning methods out perform traditional machine learning methods in terms of accuracy. In the context of the IMDB movie review classification task, this greater accuracy is likely achieved due to the BERT Model's capability to gain an understanding of each of the feature words in their respective context via the pretraining phase and to fine-tune its weights for the specific task at hand. On the other hand, traditional machine learning methods such as the Naive Bayes Model have must faster training times as deep learning models are more complex and have many more parameters to learn.

Pretraining reduces the total amount of training time such that training time only involves the fine-tuning phase to capture task specific feature patterns. In conclusion, deep learning machine learning methods out perform traditional machine learning methods in terms of accuracy, with the trade-off of training time.

4. Discussion and Conclusion

In conclusion, our project involved implementing and testing Naive Bayes and BERT models on an IMDB movie review dataset, and our findings were consistent with prior research that deep learning techniques, specifically BERT, outperform traditional techniques like Naive Bayes in terms of accuracy, despite longer training times and higher computational requirements. Furthermore, our analysis of attention matrices revealed that correctly classified texts showed high attention weight values assigned to words relevant to the sentiment of the text, while incorrectly classified texts had either homogeneously distributed attention weights or high value weights assigned to words irrelevant to the overall sentiment of the text. These insights offer a deeper understanding of how attention mechanisms can contribute to the success or failure of deep learning models in sentiment analysis tasks.

In the future, we would like to gain access to devices with more computational power in order the test the full performance capabilities of the BERT Model. For the Naive Bayes Model, we were able to use the full dataset for training. While for the BERT Model, we were only able to use a small portion the data samples to train the model. Increasing the size of the training data caused our program to crash. And so, to conduct a more accurate comparison where both models are training on the same amount of data, a computer with sufficient RAM and GPU power would be needed. More computing power would also allow for tuning the epoch hyperparamter. Nonetheless, it is expected that the BERT Model would still perform better than the Naive Bayes Model because increasing training data leads to increase in performance accuracy. In addition, it would be interesting to investigate other variations of the Naive Bayes Model such as the Gaussian Naive Bayes Model and other traditional machine learning models to discover which traditional technique proves to be the best and examine what specific features of such techniques allows for performance improvements. Lastly, as we did not extensively preprocess the input text, it would be valuable to add further preprocessing steps, such as the removal of sentiment irrelevant words, and observe BERT Model performance changes as well as test how much more data our computers can take with the reduced number of parameters.

5. Statement of Contributions

Lucas Bennett created and analysed the Naive Bayes classifier. Bjørn Christensen worked with the BERT model and data pre-processing. Megan Ng compared and contrasted the performance of the Naive Bayes and BERT model.

References

[1] Atulanandjha, Bert testing on imdb dataset : Extensive tutorial (Feb 2020).
URL `https://www.kaggle.com/code/atulanandjha/bert-testing-on-imdb-dataset-extensive-tutorial/notebook`

[2] B. Magnini, A. Lavelli, S. Magnolini, Comparing machine learning and deep learning approaches on nlp tasks for the italian language, in: Proceedings of the 12th language resources and evaluation conference, 2020, pp. 2110–2119.