Abstract

This study implemented and compared the performance of a classical Naïve Bayesian model, coded from scratch, with that of a BERT neural network model on the Internet Movie Database (IMDb) benchmarking dataset. The IMDB dataset comprises 50000 polarized positive and negative movie reviews and is divided into two equal training and test sets, each of which contains equal ratios of positive and negative instances. Inputs for the NB model were preprocessed with the removal of stop words, lemmatization, and vectorization using the CountVectorize function. Additionally, BertTokenizer from the transformers package was used to convert textual data to numerical values for the BERT model. This model was able to outperform the NB model, but it required a significantly longer time for training-prediction compared with the NB model.

Using the pre-trained BERT model, we examined different attention matrices for correctly predicted and incorrectly predicted examples of IMDb movie reviews to see the importance of particular words in sequence classification. This results in a visualization of a model's attention scores, which can suggest correct and incorrect predictions of the model.

As the full NB model required a significant amount of time for training-prediction, we endeavored to observe the effects of altered maximum features and ngrams during the preprocessing stage on the performance and training-prediction time of the NB model as a further exploratory step. It was observed that decreasing the maximum features from the full word set to lower values (e.g., 4000) reduced the amount of training-prediction time, but it also reduced model test set performance. However, NB models with lower maximum feature values still provided decent, though decreased, performance. Changing the ngram values with fixed maximum feature variables (here 4000), however, had minimal impact on model performance though the training-prediction time was increased mildly.

Introduction

In this project, we implemented 2 different language models for semantic classification of textual data, more specifically the Internet Movie Database (IMDb), which is a dataset containing 50000 movie reviews classified as either positive or negative, to obtain a prediction on the result of any given review [1]. The dataset has been previously used for various benchmarking purposes [2, 3]. To achieve this, we implemented a Naïve Bayes (NB) model from scratch. Furthermore, a BERT (Bidirectional Encoder Representations from Transformers) model [4] was also implemented through the use of the transformer and PyTorch libraries. Preprocessing for the NB model was done using the CountVectorize function. Additionally, to improve the performance of NB models and decrease computation burden, removal of stop words and lemmatization were also implemented. For BERT models, BertTokenizer from the transformers package was used for data preprocessing. We compared the two methods using three metrics, including test accuracy, F1 score, and model running time. The utilized BERT model was able to achieve better performance compared with most NB models though it required much more time to train.

For the BERT model, we used BertForSequenceClassification on the IMDb dataset for classification of the sequences of words in the reviews, and we examined the results from different attention matrices. The attention matrices showcase the relative significance of words and contextual relationships between them in a sequence classification task. Specifically, in this project, they illustrate the significance of specific words and their relationship with other words in a movie review to determine if the review is positive or negative. Upon aggregating the multi-dimensional attention matrix and examining it, the results indicate that seemingly-insignificant words have a higher weight than others. Meaning their use can aid in accurate predictions. We also found that using unconventional characters and contradictory words can lead to model error.

As the full NB model required a significant amount of time for training-prediction (5968 s), we endeavored to observe the effects of alternate maximum feature and ngram values from the CountVectorize function on model performance and accuracy. We observed that decreasing the number of tokenized maximum features also decreased training-prediction time at the cost of mild performance deterioration. Different ngram values, however, had little effect upon NB performance, but training-prediction time was mildly increased for more complex ngrams.

Dataset:

The IMDb Movie Reviews dataset is a collection of 50,000 reviews taken from the Internet Movie Database (IMDb) that are categorized as either positive or negative for binary sentiment analysis [1]. To ensure balance, the dataset has an equal number of positive and negative reviews, and it is typically split into 25000 train and test parts. The reviews included are highly polarizing, with only those that score below 4 out of 10 being labeled negative and those with more than 7 out of 10 being considered positive. To avoid potential ram problems and improve the performance of the classical Naïve Bayesian model, we used lemmatization, which is the transformation of all shapes of a word into a single entity for analysis purposes [5]. Furthermore, we also removed stop words such as "the", which are frequently used and do not significantly contribute to a sentence's meaning. Removing these words can help shift the model's focus onto more meaningful words that give a better indication of whether the review is positive or negative. For BERT models, BertTokenizer from the transformers package was used for data preprocessing and conversion to numerical values. The model was also trained over 10 iterations using GPU to help the speed of training from the dataset.

Results

Comparison of BERT and NB models

In this project, The BERT model was able to tune to the dataset with a low number of epochs, only requiring 10 iterations, (Figure 1 BERT loss). The BERT model also outperformed the NB model over the test set (Table 1, below). As seen in Table 1, the BERT model achieved higher accuracy as well as F1 score in most cases, except when the NB model had max features=All. A downside of the BERT model, however, was its relatively long training time where only over 10 iterations, and using GPU, it took approximately twice the full NB training-prediction time (Table 1).

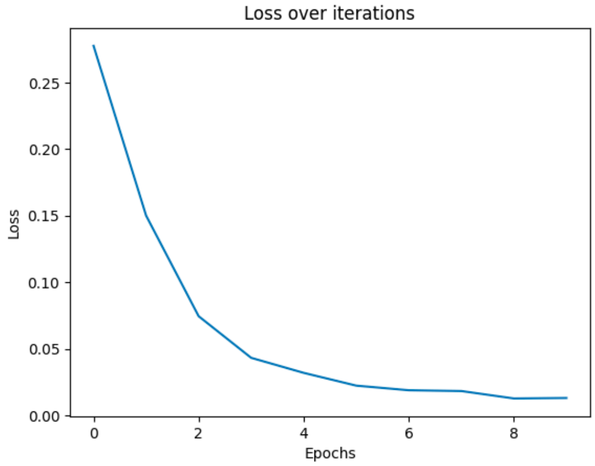
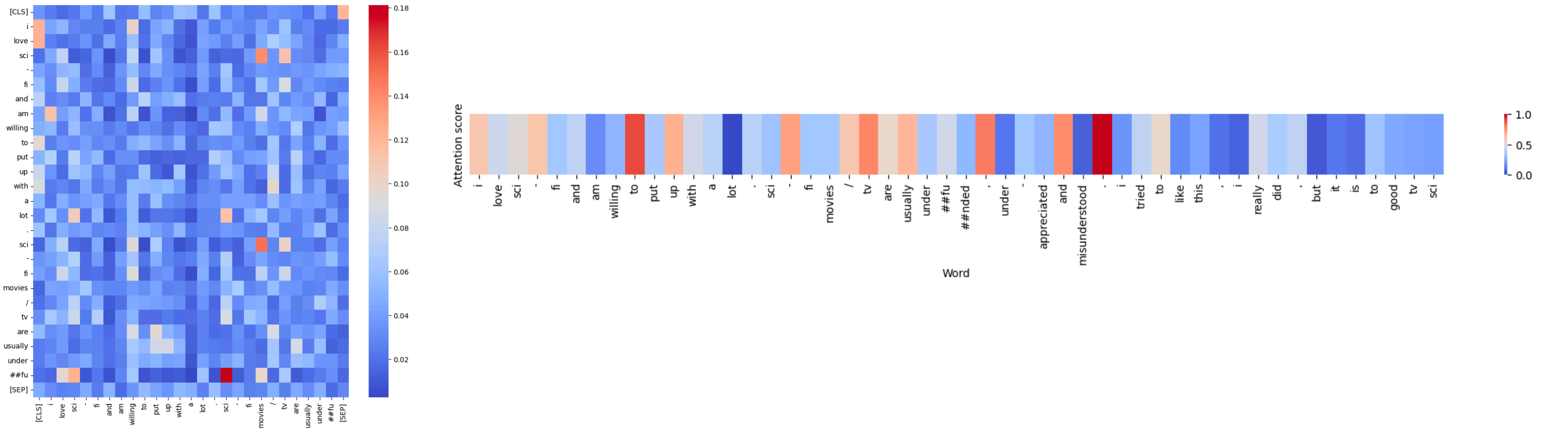


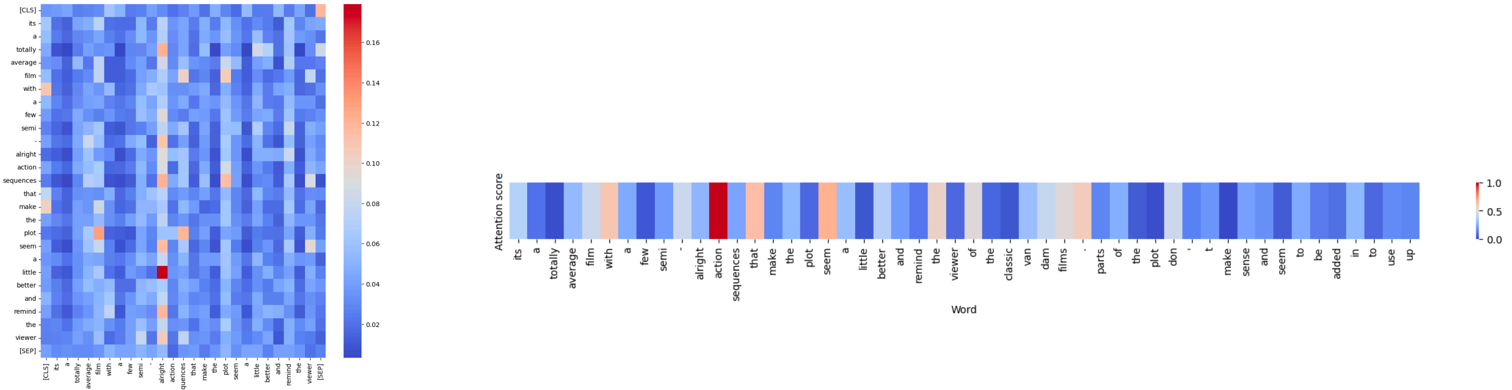
Figure 1. Loss over iterations for the BERT model. Finished with a loss of 0.0138 over 10 epochs.

Investigating attention matrices

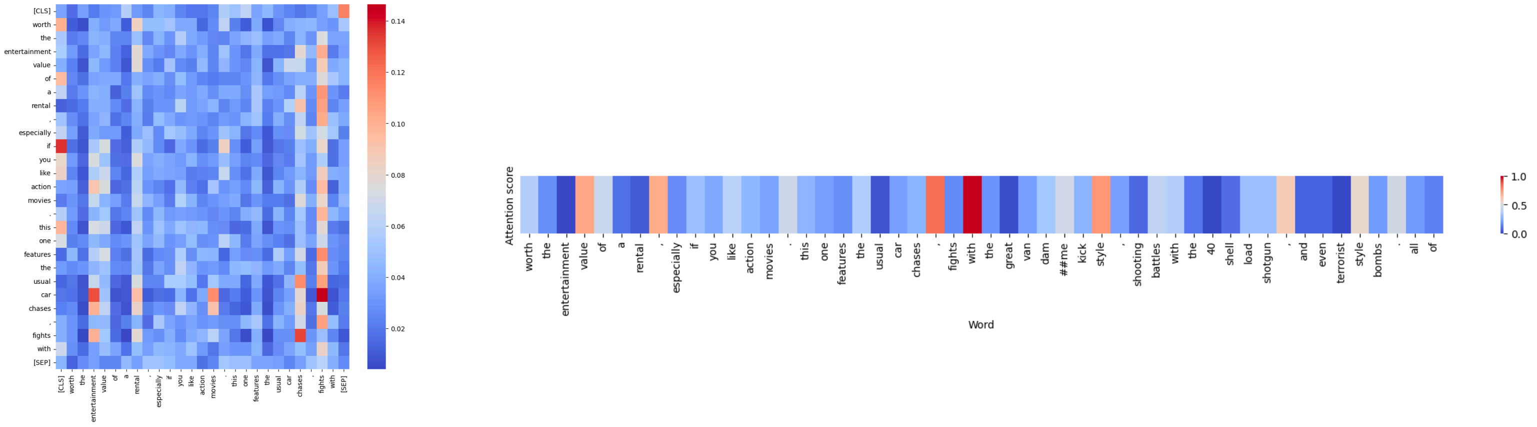
For the BERT model, we can get an inner look of how the model finds associations between different words and their indication of whether the review is positive or negative. These are in the form of attention scores, which can be seen in attention matrices like those below. Upon investigation of the attention matrices, it’s clear that the BERT model often finds accurate associations between the relations of words and their classification, which is beyond human expectation or understanding. Below are some examples of correctly predicted and incorrectly predicted attention matrices, in (N+2)x(N+2) form and 1xN form.



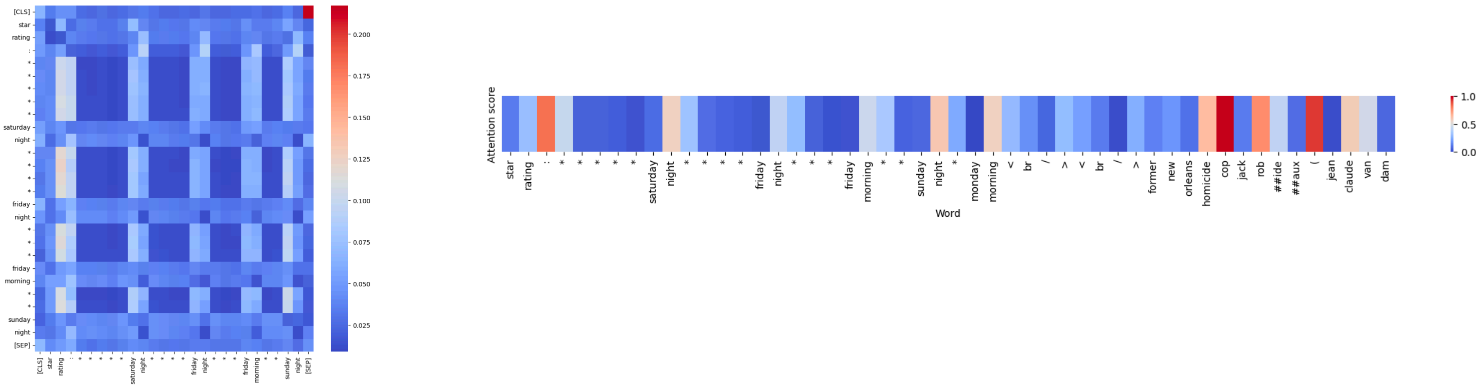
Matrix 1. Correctly-predicted sci-fi review



Matrix 2. Correctly-predicted average review



Matrix 3. Incorrectly-predicted review with many high attention scores



Matrix 4. Incorrectly-predicted unconventional character review

It should be noted that the matrices, although still large, have been shortened for visualization sake and do not represent the full attention scores for every word in the review. It should also be noted that the 1xN matrix has an attention score normalized between 0 and 1, while the (N+2)x(N+2) matrix does not. Upon examination of these matrices, we can see that there are some seemingly insignificant relations that garner high attention scores. For instance, in the first graph, the combination of “##fu” and “sci” results in a dark red box. Similarly, in the second graph, “little” and “alright” has a dark red box as well, indicating that these combinations are important for prediction. For the two incorrectly predicted examples, the first one looks very similar to the two correctly predicted ones, but with some more darker red values. This would suggest that the particular review has more high-attention combinations, and if many are contradictory, it could lead to the BERT model making an error. Finally, in the last matrix, square shapes from the use of asterisks can be seen.

This suggests that using unconventional characters a repeated amount can lead to model error, likely from BERT's inability to make relations between parts of the sentence and predict meaning behind them.

Different maximum feature and Ngram values with NB

As an additional experiment, we opted to experiment with various maximum feature values from the CountVectorizer preprocessing with the NB model. Setting a value for maximum features limits the preprocessing to only consider the specified number of features based on occurrence frequency. For instance, max features=4000 will only choose the 4000 most frequent words from the inputs for preprocessing. In this project, we used 4000, 8000, 12000, and all of the features for training and testing. The results revealed that increasing the number of model features, as expected, increased the NB performance. However, the training-prediction time was markedly increased as the model needed to perform calculations for more features (Table 1, Figure 2). Using different ngrams (e.g., (1,2), (1,3), etc) for a fixed maximum feature value of 4000 did not significantly alter performance although the training-prediction times were mildly increased for more complex ngrams. This is probably due to the nature of the dataset being curated to be more polarized and simpler in some sense.

Table 1. Model performance and training-prediction time of NB and BERT models

Model	Accuracy (%)	F1 score (%)	Training-prediction time (S)
NB (max features=4000)	85.8	86.4	475
NB (max features=8000)	86.7	87.3	786
NB (max features=12000)	87.4	87.9	1264
NB (max features=All)	91.1	91.3	5968
BERT	90.98	91.3	9328

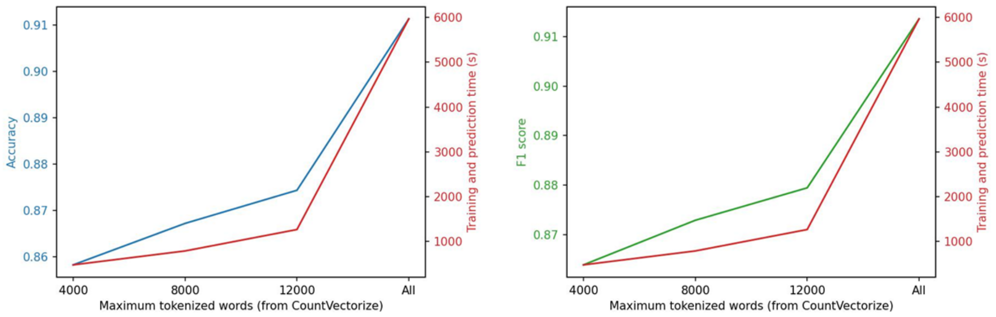


Figure 2. Accuracy and F1 score of NB models with different maximum features along with the required training-prediction time.

Conclusion

The project investigated and compared the performance of a classical machine learning framework, NB, with a pretrained deep neural network, BERT, in a semantic classification framework using the IMDB dataset. We observed that the NB model performed well, but was outperformed by the BERT model in most cases, although the latter required a longer time for training and tuning on the dataset over 10 iterations using GPU. The NB model was able to achieve decent performance in part due to the implemented preprocessing steps, namely the removal of stop words and lemmatization. Interestingly, the pretrained BERT model was able to achieve a decent performance over only 10 iterations, which displays the power of utilizing pretrained models for solving similar tasks to the data on which they have been previously trained. The classical NB model, however, if pretrained on a certain task, will be less of a tool in situations where the new project's question is different than the pretrained framework.

With regard to attention matrices in the BERT model, it's not always completely obvious when a review is going to be positive or negative, or when the review it going to be correctly or incorrectly predicted. Attention matrices can give some insight, showcasing that seemingly insignificant combinations of words can lead to accurate classification of reviews because they have high attention scores. They also suggest that ambiguous reviews with contradictory words or repeated use of unconventional characters can confuse the model and lead to error. Evidently, there are complex mechanisms within BERT that understand these associations and make accurate predictions based upon them.

As a further explorative step, we altered max features and ngram values in the CountVectorize function during the preprocessing phase of the NB model. Lower max feature values resulted in faster model training-prediction but reduced prediction performance. Different ngram values, on the other hand, had little impact on prediction performance over a fixed max feature value. These results could be due to the nature of the dataset as it is both polarized and somewhat simple. Future studies can utilize other language models for comparison and also use models trained on radically different textual frameworks to investigate the amount of re-training required to adjust them to the task at hand.

Statement of Contributions

Mahdi Mahdavi: Coding and running of the NB models. Writing and revision of the report.

Yu Cheng: Coding and running of the BERT models. Writing and revision of the report.

Taylor Fergusson: Coding and running of the BERT models. Writing and revision of the report.

References

1. Maas, A., et al. *Learning word vectors for sentiment analysis*. in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*. 2011.
2. Dai, A.M. and Q.V. Le, *Semi-supervised sequence learning.* Advances in neural information processing systems, 2015. **28**.
3. Johnson, R. and T. Zhang, *Effective use of word order for text categorization with convolutional neural networks.* arXiv preprint arXiv:1412.1058, 2014.
4. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding.* arXiv preprint arXiv:1810.04805, 2018.
5. Müller, T., et al. *Joint lemmatization and morphological tagging with lemming*. in *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2015.