COMP551 A4

Yusen Tang, Keyu Yao, Jaylene Zhang

December 2022

Abstract

In the modern world, social media user data is generated every millisecond. As reviews, feelings, and opinions of people expressed in writing form, data's significance can be noticed or seen. Customer-generated information may relate to occasions, foods, goods, etc.It is obvious that analysing user-generated material must be difficult to handle and time-consuming because the data is in the form of bulk data. For this, we require an intelligent system that aids in classifying the information according to its positivity, negativity, or neutrality.The goal of this work is to apply the BERT model to analyse a dataset of IMDB user-generated movie reviews in order to ascertain the concept of human emotions. The strength and suitability of our experimental technique corresponds to the efficacy of sentiment analysis.We found that BERT achieved worse/better accuracy than the more traditional ML methods and was significantly faster to train.We achieved the test AUROC of 94.30 %

Introduction

In this assignment, we implement one machine learning model, Bidirectional Encoder Representations from Transformers(BRFT) on the IMDB reviews dataset. We also implement sklearn models on this dataset, including LR, SVM, RF, and XGBoost. We will report the binary classification performance in terms of AUROC for BERT, LR, RF, SVM, and XGBoost on the IMDB test data and examine the attention matrix between the words and the class tokens for some of the correctly and incorrectly predicted documents.

The IMDB dataset contains movie reviews along with their associated binary sentiment polarity labels. It is intended to serve as a benchmark for sentiment classification. The sentiment includes positive and negative sentiment and the overall distribution of labels is balanced (25k pos and 25k neg). In the labeled train/test sets, a negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. Thus reviews with neutral ratings are not included in the training/test sets.

Related Work

Till now, several studies have done on text classification and sentiment analysis. In a paper published by Fudan University exhaustive experiments were conducted to investigate different fine-tuning methods of BERT on text classification task and they offered a general solution for BERT fine-tuning. Finally, the proposed solution obtains new state-of-the-art results , with the highest accuracy being 95.79. In addition to training large language models like Bert and GPT-2, researchers have found the polarity by using a fuzzy set theory. According to the Journal of Soft Computing and Decision Support Systems, some experiments were carried out on patient's reviews on several different cholesterol lowering drugs to determine their sentiment polarity. And the most interesting finding is that there exist some type of objective sentences that do not contains any sentiment words whereas they express sentiment and can be regarded as an opinionated sentence. And the novel technique developed by the researchers is believed to be a reliable method for discriminating these kinds of sentences from non-opinionated one and determining their sen-

timent.

Citation

Yazdavar, A.H., Ebrahimi, M. and Salim, N., 2017. Fuzzy based implicit sentiment analysis on quantitative sentences. arXiv preprint arXiv:1701.00798.

Chi Sun, Xipeng Qiu, Yige Xu, Xuanjing Huang, 2020. How to Fine-Tune BERT for Text Classification?, arXiv:1905.05583

Saad Abdul Rauf et al., 2019. Using BERT for Checking the Polarity of Movie Reviews, International Journal of Computer Applications

George Mihaila, Complete tutorial on how to use GPT2 for text classification, Github

ATUL ANAND, 2019, BERT testing on IMDB dataset : Extensive Tutorial, Kaggle

Datasets

To preprocess the IMDB dataset, we load svmlight file using the function loadsvmlightfile() to load all the features' counts as X, and sentiment benchmark as Y. We firstly transfer Y data into a binary numpy array, making positive review(score $\geq$ 7) as 1 and negative review(score $\leq$ 4) as 0. Since X has an array attribute called **indices** consisting of all the word indices in each review file, we then group indices by features and obtain the number of files that each word occurs in. We filter out the words appearing in less than 0.01 of the files and those more than 0.5, preventing us from choosing wrong features in later processing steps. Then we use the standardized X and calculate the z-score of each rows by using the formula:

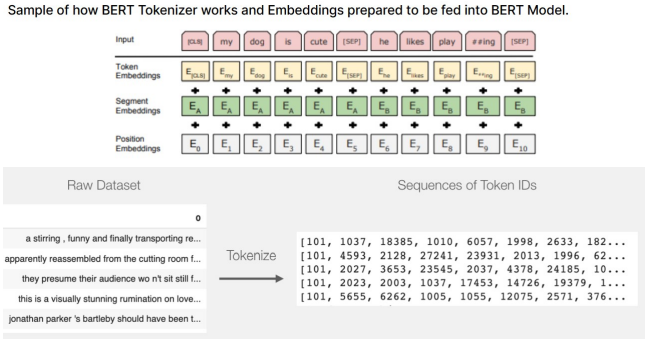
$$z = \frac{X^T Y}{\sqrt{N}} \tag{1}$$

We choose a total of 500 features with the highest z-score since the greater the z-score the more likely we are to reject the null hypothesis. However, we will experiment with varying feature sizes in later steps. For Bert, we simply used the unstructured text imdb data downloaded from datasets.

We tried to implement the models by using full datasets, 25000 samples. But it takes really long time for fitting the models. So we choose 2000 samples for training and 2000 for testing. The choosing training samples are from 11499 to 13499 so we could have balanced positive and negative labels.

Benchmark

For Bert, first we convert our text data into tokens, and then we calculate our token ids for fitting into the BERT model. We import the pre-trained BERT model from pytorch pretrained bert import BertModel.



Then Masking few random IDs from each sentences to remove Biasness from model. BERT extracts patterns or representations from the data or word embeddings by passing it through an encoder. The encoder itself is a transformer architecture that is stacked together. The model has 12 transformer layers and 12 attetion heads. We fine tune the model with our data. For logistic regression, we used the L2 penalty. For SVM, rbf kernal was used. For RF, we used the default setting of 100 trees. For XGBoost, the default booster was set to gbtree.

Results

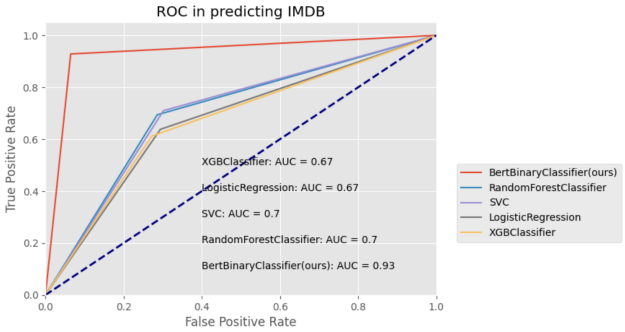
We implement the BERT model with 4 epochs and 4 batches on the full dataset and achieve the accuracy of 94.30 %

	precision	recall	f1-score	support
False	0.93	0.94	0.94	12500
True	0.94	0.93	0.94	12500
accuracy			0.94	25000
macro avg	0.94	0.94	0.94	25000
weighted avg	0.94	0.94	0.94	25000

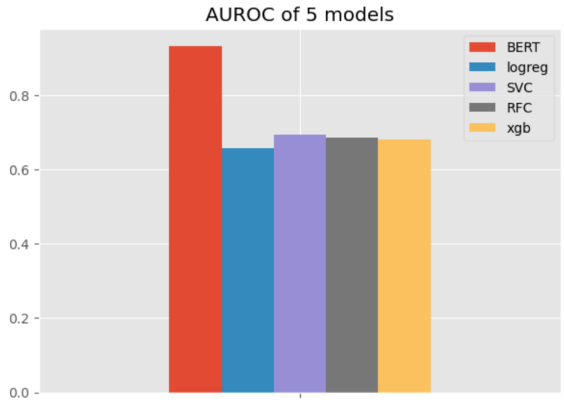
Also, for initial experiment, we tried BERT model with 4 epochs and 4 batches on 1000 train and test samples. We achieve the accuracy of 91.30 %

	precision	recall	f1-score	support
False	0.9041	0.9240	0.9139	500
True	0.9223	0.9020	0.9120	500
accuracy			0.9130	1000
macro avg	0.9132	0.9130	0.9130	1000
weighted avg	0.9132	0.9130	0.9130	1000

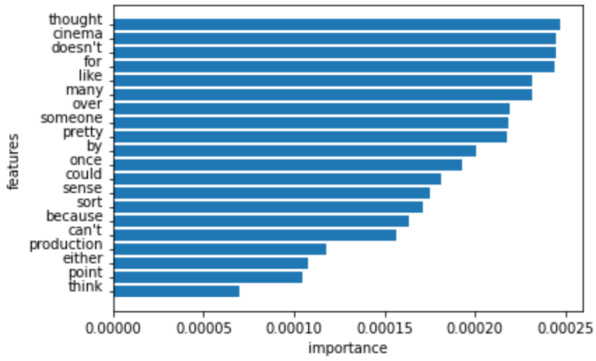
We drew A single plot containing four ROC curves of BERT, LR, SVM, RF and XGBoost on the IMDB test data.



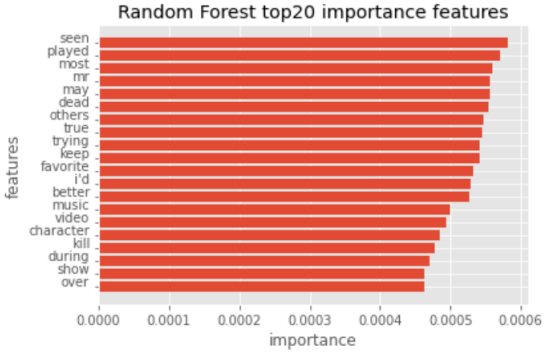
We could see that the BERT model show a folded line and the turning point is between FPR=0 and FPR=0.2.And the ROC value of BERT equals to 0.93. The other four models show the relatively low ROC values and the XGB classifier shows the lowest.



And we draw the correspoding AUROC bar plot. We could see the the AUROÇ value of the BERT model also shows the highest,and the logistic model shows the lowest value.



We drew a horizontal bar plot showing the top 20 important features from RF on the IMDB data with the feature importance scores as the x-axis and the feature names as the y-axis.



The highest important feature(word) is "thought"

and "think" is the 20th important word. However, the 2000 samples may be limited for features selection so we fit the 25000 samples and select the top 20 again. We can see that the highest is "word", and the "over" is the 20th important word.

Comparing with GPT2

GPT-2 is a transformers model pretrained on a very large corpus of English data in a self-supervised fashion. Usually, it is trained to guess the next word in sentences. And since GPT-2 is a decoder transformer, the last token of the input sequence is used to make predictions about the next token that should follow the input. As a result, instead of using first token embedding like what we did in Bert, we used the last token embedding to make prediction with GPT2. In this assignment, we used the GPT-2 from the Hugging Face transformer library.

In order to save memory, our GPT-2 model used batch size of 4 and epoch size of 4 just like what we did in Bert. The following chart shows the precision, recall and F1 score for the model. We can see that the accuracy obtained from the GPT-2 model is 0.83, which was much lower than what we got using the Bert model.

	precision	recall	f1-score	support
neg	0.84	0.83	0.83	12500
pos	0.83	0.84	0.83	12500
accuracy			0.83	25000
macro avg	0.83	0.83	0.83	25000
weighted avg	0.83	0.83	0.83	25000

Discussion and Conclusion

We predicted the polarity of IMDB movie reviews by using the concept of sentiment analysis using Bert Model. The parameters which we used in our model are Eval Accuracy, F1 score, Precision and Recall Where our batch size is 4, for 4 epochs. This model gave us a great result. By comparison of all the other models, Bert gives us the best accuracy. We therefore conclude that Bert is the best model for sentimental text analysis. In future work, we are looking forward to using this model and technique with the stream or online data.

Future Study

For future investigation, we would like to try out another pre-training task that is NSP(next sentence prediction) since it is one of the two training process behind the BERT model. NSP consists of giving BERT two sentences, sentence A and sentence B, and the model will predict whether B is the next sentence followed by A. Also, it is worth mentioning that while MLM is used to understand relationships between words, NSP teaches BERT to understand longer-term dependencies across sentences.

Statement of Contributions

Jaylene Zhang: Data processing and result analysis
Keyu Yao: Data loading and model implement
Yusen Tang: Experiments and Tuning