Performance Comparison of Text Sentiment Classification Models

Tianyao Pu
Dept. of Computer & Info. Science
Fordham University
New York, NY, USA
tpu3@fordham.edu

Jee Hun Kang
Dept. of Computer & Info. Science
Fordham University
New York, NY, USA
jkang78@fordham.edu

***Abstract*— Sentiment analysis has become a popular and common task in Natural Language Processing. There are various classification models available to perform the sentiment analysis. We are interested in investigating whether complex models generally perform better than traditional models on the identical text mining task. By performing the sentiment analysis task with different classification models with varying model complexities on an identical dataset, we can explain the cause of the potential differences in model performance based on the research results. The results should not only include the prediction accuracy of each model, but also other factors that are involved throughout the process, for example, the computational efficiency. The entire process includes but is not limited to data analysis, data preprocessing, model training and tuning, and result comparison.**

Keywords—sentiment analysis, text mining, classification model performance, model creating and training

I. Introduction

To our knowledge, advanced neural network models have been known to produce higher prediction accuracies compared to traditional classification models. Therefore, our intuition for our research is that more complex and advanced models will produce higher prediction accuracies than traditional models when an identical dataset is used.

To investigate whether our intuition is correct, we picked four classification models, which are Logistic Regression, Support Vector Machine, Long Short Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT, pre-trained). We set out to solve our question by implementing these four models to perform the sentiment classification on the IMDB dataset [1]. Our initial thought is that the neural network model LSTM will perform better than traditional machine learning models (e.g., Logistic regression and SVM); the current popular model in Natural language processing - BERT will yield the best performance result among these four models.

This paper will therefore include detailed explanations for each experiment setting steps as well as results and conclusions.

In our search for related work, we were able to find ways to perform parameter tuning for LR and SVM using Scikit Learn [2]. Moreover, we found the introduction to the parameters of vectorizer [3], which is an essential tool for our text mining project. We found the introduction to the parameters of LR and SVM [4]. Lastly, to actualize the experiment with BERT, we found a simplified version of BERT, DistilBERT, and its related introduction of deployment [5].

As for the contribution of work on this project. Tianyao Pu and Jee Hun Kang contributed equally.

II. Experiment Methodology

A. Data set

A single IMDB dataset is used throughout the entire process, and this dataset contains 50,000 highly polarized movie reviews. The dataset consists of two columns and 50,000 rows, and each row accounts for each entry.

The first column is named “text”. This column contains review texts provided by reviewers. Every review has a different text length.

The second column is named “label”, and this column contains classified sentiment results labeled by binary values 0 and 1, where 0 indicates negative review, which has a score less or equal to 4 out of 10, and 1 indicates positive review, which has a score greater or equal to 7 out of 10.

B. Models introduction

a) Logistic regression: Logistic regression is the simplest yet most widely used classification technique amongst our four models. The model uses the logistic function to find the best fitting model to explain the causality between the dependent and independent variables. The model is usually used for predicting binary values, and it requires the vectorization of text data before the training process.

b) SVM (Support Vector Machine): SVM is a supervised machine learning algorithm which can be used when solving both regression and classification problem. The algorithm goes through an extensive transformation which enables the model to understand the relationships between data points and solve complex classification problems. Therefore, it is known to be computationally expensive.

c) LSTM (Long Short-Term Memory): LSTM is part of a neural network family, and more specifically, the particular version of RNN (Recurrent Neural Network). While RNN suffers from handling long-term dependencies, LSTM is capable of long-term memory. The three gates (input, forget and output) in LSTM enable the model to only keep relevant data and produce highly effective results. However, LSTM generally performs better when the dataset is large.

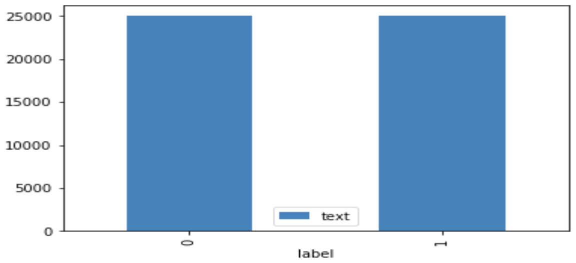
d) BERT (Bidirectional Encoder Representations from Transformer): BERT is a transformer-based technique for NLP (Natural Language Processing), and it was created by Jacob Devlin and his colleges at Google in 2018. Since the technique emerged, Bert has been sensational in the world of NLP as it deeply and better understands the context and nuance of a word, and this is feasible as the model uses a bi-directional approach when looking at a text sequence. In the project, DistilBERT was used, which includes 40% less parameters than the original model yet performs 60% faster while maintaining over 95% performance.

C. *Data preprocessing*

Data preprocessing is beneficial for data mining to ensure only the relevant data can be used for model training. Especially for text classification, text data needs to be converted into vector form for computers to decode the texts. Before diving into data preprocessing, it is crucial to have a deeper understanding of the dataset to filter out unnecessary data and only work with the necessary ones for the project.

Many scientists and researchers struggle with class imbalance in the dataset. This results in skewed class distribution and unequal cost of misclassification errors, which greatly alters the prediction result. Therefore, it was our priority to check whether the dataset was imbalanced or not. As illustrated in table 1, there was no class imbalance found in the dataset.

TABLE I. CLASS IMBALANCE

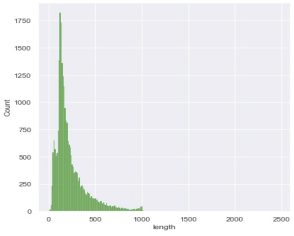


The distribution of text lengths for positive and negative reviews showed a similar pattern. Review lengths were most frequent in the 200-300 range, with their mean and standard deviation values falling into a similar range. Table 2 illustrates this further.

TABLE II. REVIEW LENGTH

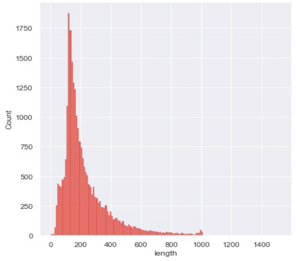
Distribution of text length for positive sentiment reviews.

	length
count	25000.0
mean	232.85
std	177.5
min	10.0
25%	125.0
50%	172.0
75%	284.0
max	2470.0



Distribution of text length for Negative sentiment reviews.

	length
count	25000.0
mean	229.46
std	164.95
min	4.0
25%	128.0
50%	174.0
75%	278.0
max	1522.0



Text alteration was performed to increase the model prediction accuracy and computational efficiency.

First, entire texts were converted into small letters, and irrelevant texts were eliminated. These steps include removing HTML strips, punctuation marks, and stop words. These irrelevant texts do not add weights in determining class labels and occur too frequently throughout the texts. Therefore, eliminating these irrelevant texts improves computational efficiency and prediction performance.

Once irrelevant texts are removed, lemmatization converts words into their original root. For example, the words "stops", "stop", "stopping" carry the same meaning; however, when changed to vector form, computers treat these words differently. As lemmatization generally performs better in understanding contexts and converting words into meaningful roots than stemming, lemmatization was used in our project. After the above preprocessing steps, the first review record changed from "I grew up (b. 1965) watching and loving The" to "i grew up b watching and loving the".

Then, texts were converted into vectors. For logistic regression and SVM, TF-IDF vectorizer from Scikit-Learn is used, whereas Tokenizer from Keras is used for LSTM. For TF-IDF vectorization, by setting *min_df* to 10, we managed to eliminate words that occur less than 10 times throughout the texts. Also, by setting *ngram_range* to (1,2), only unigram and bigram of words were considered for TF-IDF vectorization.

D. *Models training process*

For training and test data splitting, 80% of the data was used for training, and 20% of the dataset was used to test all four models.

The popular machine learning library Scikit-learn was used for Logistic regression and SVM. We trained the models with default settings and acquired the results based on different percentages of dataset sampling from 2% to 100%.

For LSTM, we utilized the machine learning library Keras. We trained the model with a basic setting based on different percentages of dataset sampling from 2% to 100%.

For DistilBERT, we utilized the machine learning framework Pytorch. Due to limited hardware available (out-of-memory issue), we failed to deploy the model based on a 100% dataset. Thus, we only deploy the model based on 2% of the dataset.

E. Models tuning

After training and testing each model, appropriate model tunings for each model were performed to set the proper parameter values to increase the prediction accuracy.

For Logistic Regression and SVM, GridsearchCV was implemented. Doing so, appropriate *C* and *max_iter* values could be found for Logistic Regression, and *C* and *gamma* values were found for SVM.

For LSTM, the research team wasn't able to perform model tuning with limited knowledge in terms of hyperparameter optimization. However, an epoch optimization process was performed to optimize the model's computational efficiency. The validation set was sampled from the training set based on a 10% dataset with a 20% split, and then we drew training and validation loss/accuracy graphs to find the relationship between loss/accuracy and the number of epochs to find the optimal epoch setting.

Model tuning for DistilBERT was not implemented as DistilBERT is a pre-trained model that does not require tuning unless fine-tuning is needed.

F. Evaluation metrics

Only the accuracy metric was used to evaluate the performance of the models on the dataset with binary class distribution without class imbalance. Using the same test dataset, the accuracy of each model was recorded.

III. Result

A. Initial result without model tuning

Since DistilBERT only produced the prediction result when only two percent of the dataset was used, a comparison amongst four models was only fair when only two percent of the data was considered.

However, performance comparison for Logistic Regression, SVM, and LSTM was possible with all fractions of data and, therefore, compared the performances before models were tuned.

Since there was a lack of knowledge to tune LSTM and a significant trade-off between the number of epochs and training time, we only present the final result from LSTM acquired based on a 100% dataset.

Before the models were tuned, when 100 percent of the dataset was used, SVM produced the highest accuracy with 0.895, followed by Logistic Regression and LSTM being 0.891 and 0.874, respectively, as presented in Table 3. Training time for each model was recorded, where Logistic Regression, SVM, and LSTM spent 2 seconds, 2,400 seconds, and 14,000 seconds to train, respectively.

TABLE III. INITIAL PREDICTION ACCURACY WITH DEFAULT SETTING

Models	**LR**	**SVM**	**LSTM**
Accuracy	0.891	0.895	0.874
Training Time	2s	2,400s	14,000s

B. Final result with model tuning

The Final results available based on all fractions of the dataset were recorded as presented in Table 4. When the Logistic Regression and SVM models were tuned, both models' performances were improved, where SVM produced the highest accuracy of 0.907, followed by Logistic Regression with an accuracy of 0.905.

Logistic Regression and SVM performed better than LSTM in all fractions of the data; however, it shows that LSTM's performance improved steeply when the dataset's size gets larger compared to Logistic Regression and SVM.

When two percent of the dataset was used, DistilBERT produced the least accuracy amongst the four models. The Model evaluating time-based on two percent of the dataset is roughly 1,800s.

TABLE IV. PREDICTION ACCURACY AFTER TUNING

Models	**LR**	**SVM**	**LSTM**	**BERT**
2%	0.789	0.785	0.775	0.759
10%	0.857	0.854	0.801	X
50%	0.889	0.885	0.847	X
100%	0.905	0.907	0.874	X

C. LSTM training tuning opimization result

The training and validation loss/accuracy graphs of LSTM were generated as shown in Table 5. The loss of validation set starts to increase after epoch 3, and the training accuracy decreases after epoch 2. To optimize the model's computational efficiency, we set the epoch from 7 to 3 to evaluate the model. The training time and accuracy after the change were recorded in Table 6. We found a great extent of the trade-off between the number of epochs and training time. The training time after the change decreased to 5,040s from 14,000s, whereas the prediction accuracy was slightly reduced by 0.013.

TABLE V. TRAINING AND VALIDATION LOSS/ACCURACY

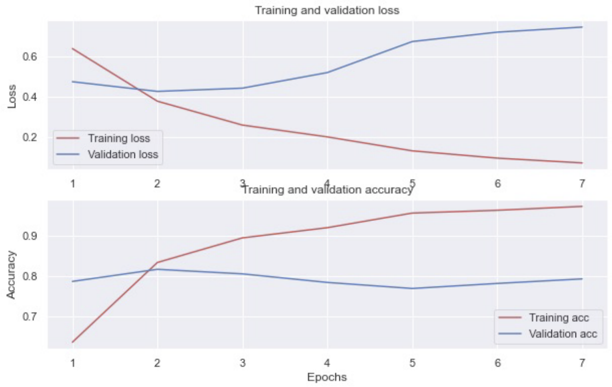


TABLE VI. LSTM EPOCH ACCURACY TRADE-OFF

Records	**Epoch 3**	**Epoch 7**
Training time	5,040s	14,000s
Accuracy	0.861	0.874

IV. Conclusions

A. Model accruacy

Logistic Regression and SVM performed better than LSTM based on all fractions of the dataset. In our only fair comparison that used 2 % of the data, Logistic Regression and SVM performed better than LSTM and DistilBERT. The SVM performed slightly better than Logistic Regression, and it was the model that produced the highest accuracy among the four.

We consider that the LSTM and DistilBERT have great potential to yield better result with larger dataset. However, we consider that the LSTM and DistilBERT were not the appropriate models for the dataset used in this project.

B. Model tuning

We can conclude from the experiment results that the model hyperparameter tuning applied on Logistic Regression and SVM were helpful to improve the models' prediction accuracy. However, the improvement is limited.

The training time of the LSTM model was significantly saved after the epoch setting optimization, yet accuracy remained stable. We can conclude that our LSTM training time optimization was successful.

C. Model computational efficiency

We can conclude that the simpler the models were, the higher computational efficiencies there were. The Logistic Regression is the most computationally efficient model among the four models tested.

D. General conclusions

We can conclude that Logistic Regression is the ideal model among the four models used for our text sentiment classification task. Logistic regression yielded similar prediction accuracy as SVM but took much less time to train.

When we launched this project, our initial intuition was that a more complex and advanced model would yield higher accuracy results than the traditional models. However, our results showed that the traditional models performed better than the advanced models on our dataset.

We learned that selecting appropriate models for different data science tasks is vitally important due to each model's varying characteristics, which can affect prediction accuracy and maximizing efficiency.

We experienced and learned the importance of model deployment and hyperparameter tuning. Furthermore, after trial and error in the LSTM epoch setting, we learned about the trade-off between computational efficiency and prediction accuracy.

V. Future work

Our performance results are solely dependent on using pre-processed data. To address the importance of pre-processing, our future work will include conducting performance comparisons of models with and without the data pre-processing. We assume that training models with raw data will result in less accurate predictions and worse computational efficiency.

Moreover, we would like to study deeper about low-level processes for LSTM and BERT to understand the details of the models. This will enable us to choose appropriate hyperparameters and provide the ability to tune the models to an optimal level. By doing so, we will better understand the trade-offs in choosing different parameter values for our models.

References

[1] Z. Yuan, "IMDB dataset (sentiment analysis) in CSV format," Kaggle, 28-Nov-2019. [Online]. Available: https://www.kaggle.com/columbine/imdb-dataset-sentiment-analysis-in-csv-format. [Accessed: 17-Dec-2021].

[2] "Sklearn.model_selection.GRIDSEARCHCV," *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html. [Accessed: 17-Dec-2021].

[3] Russell, "More NLP with Sklearn's countvectorizer," *Medium*, 03-Aug-2017. [Online]. Available: https://medium.com/@rnbrown/more-nlp-with-sklearns-countvectorizer-add577a0b8c8. [Accessed: 17-Dec-2021].

[4] A. M. Kumar, "C and gamma in SVM," *Medium*, 17-Dec-2018. [Online]. Available: https://medium.com/@myselfaman12345/c-and-gamma-in-svm-e6cee48626be. [Accessed: 17-Dec-2021].

[5] "Distilbert," *DistilBERT*. [Online]. Available: https://huggingface.co/docs/transformers/model_doc/distilbert. [Accessed: 17-Dec-2021].