COMP47650 - Term Project (IMDB Movie Review - Sentiment Analysis)

Abhinav
22200051
abhinav@ucdconnect.ie
University College Dublin

***Abstract*—Deep learning techniques have been used to effectively predict user ratings in the area of sentiment analysis. This project report paper analyzes the performance of three distinct models for training and developing a sentiment analysis model: BERT, Gpt2, and a custom LSTM. The models were tested for accuracy in classifying sentiment after being trained on a imdb movie review data. The custom LSTM model was created especially for this project, whereas BERT and Gpt2 are pre-trained language models that have been fine-tuned for sentiment analysis. The accuracy, precision, recall, and F1 score of each model's performance were compared. According to the findings, all three models could successfully categorize sentiment, with the custom LSTM model reaching the best level of accuracy. The usage of various sentiment analysis models is also discussed in this work, which also highlights the possibility of custom models outperforming trained models in certain situations.**

I. Introduction

This project task on sentiment analysis is a widely used natural language processing (NLP) technique that requires to identify the sentiment given in an IMDB dataset. In recent years, techniques using deep learning demonstrate impressive results in the field of sentiment analysis, and various models have been developed to achieve this goal. This report paper explores the efficiency of three distinct models for sentiment analysis on the widely-used IMDB movie review dataset. In this report, I have performed a performance comparison of three different models, namely the BERT model, Gpt2 model, and a custom LSTM model that was created and trained on the dataset. The models have been evaluated based on their accuracy, F1 score, precision, and recall, and an in-depth analysis of the strengths and limitations of each model is shown.

The project report is structured in the following manner: Section 2 presents an in-depth review of the existing works performed on sentiment analysis utilizing deep learning methodologies. The datasets and preprocessing procedures utilized in the experiments that I performed are explained in Section 3. The following section (Section 4) describes the three models used in the experiments, including each of their training and evaluation techniques. Section 5 comprises the outcomes of our experiments, accomplished by an analysis of the merits and demerits of each model. In the end, Section 6 presents conclusive remarks and recommendations for potential future research efforts.

II. In-Depth Review of the existing work

A. Related Works

A variety of solutions have been proposed for the sentiment analysis of reviews and comments, showing both the scope and complexity of this experiment. Deep learning approaches from long ago to recent deep learning architectures have all been used in distinct works. Because of their improved performance in capturing complex patterns and correlations in the data, deep learning-based models have become the benchmark in sentiment analysis in recent years.

1) Traditional Learning Algorithms: Traditional machine learning algorithms like Naive Bayes, support vector machines, and logistic regression are among the most used methods. These models employ information extracted from the text by means of bag-of-words, n-grams, and word embeddings to determine whether a review is positive or negative.

2) Long Short-Term Memory (LSTM): The Long Short-Term Memory (LSTM) model proposed by Sepp Hochreiter and Jürgen Schmidhuber in their 1997 is a widely recognized algorithm utilized in deep learning for the purpose of sentiment analysis. Furthermore, the model has demonstrated highly encouraging outcomes in numerous investigations pertaining to sentiment analysis or other forms of analysis. Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) are two additional models that have demonstrated successful application in the field of sentiment analysis.

3) Convolutional Neural Networks (CNNs): Another approach that has been proposed by Yoon Kim in his 2014 paper which involves the utilization of Convolutional Neural Networks (CNNs) for the purpose of feature extraction from textual data, which is then followed by a fully connected layer that facilitates classification. The aforementioned methodology has exhibited encouraging outcomes, as certain models have attained precision levels surpassing 90 percent.

4) Recurrent Neural Networks (RNNs): An alternative approach entails the utilization of Recurrent Neural Networks (RNNs) by Paul Werbos in his 1988 paper, more specifically Gated Recurrent Units (GRUs), to model time-dependent dependencies within sequential data, including textual data. These models have exhibited significant efficiency in capturing long-range dependencies in textual data and have achieved state-of-the-art performance for sentiment analysis.

Fig. 1. Flow the project: For all three models

5) BERT GPT-2: In recent times, pre-trained language models, such as BERT and GPT-2, have exhibited exceptional proficiency in a variety of natural language processing tasks, including sentiment analysis. The linguistic representation of a vast collection of textual data is acquired by both models through the application of unsupervised learning techniques.

Although all of these methodologies have exhibited encouraging outcomes, they possess distinct advantages and disadvantages. Conventional machine learning models exhibit higher speed and lower computational requirements, although their ability to capture complicated data relationships may be inadequate to that of deep learning models. In contrast, deep learning models possess the capability to apprehend complicated relationships within the data, though which requires greater computational resources and exhibits more vulnerability to overfitting. Pre-existing language models have demonstrated significant effectiveness, although they might require significant refinement to attain optimal levels of performance.

My work differs from previous studies as it centers on evaluating and contrasting the efficiency of BERT, GPT-2, and a customized LSTM model in the context of sentiment analysis on the IMDB movie review dataset. The study additionally investigates the effect of stop words on the accuracy of the model and shows the constraints of stemming and lemmatization in accurately capturing the true meaning of a statement. Furthermore, we investigate the utilization of emojis as a substitute mode of sentiment manifestation in evaluations.

III. Experimental Setup

1) Dataset: The corpus utilized in this study comprises of IMDB film critique information. The corpus comprises 50,000 English language reviews.

2) Preprocessing:

- Conducted a null value check on the dataset and eliminated null values
- Verified and eliminated any instances of duplicated data in the dataset
- Eliminated URLs, HTML tags, mentions (@), and converted the text to lowercase
- Eliminated stop words and punctuation from the text
- Implemented text stemming on the preprocessed text

3) Predicting without training: Using pre-trained models (BERT and GPT2) to make predictions on 500 randomly selected samples from the dataset.

review	sentiment
49582 unique values	2 unique values
One of the other reviewers has mentioned that after watching just 1 Oz episode you'll be hooked. The...	positive
A wonderful little production. The filming technique is very unassuming- very old-time-B...	negative

Fig. 2. glimpse of IMDB movie dataset

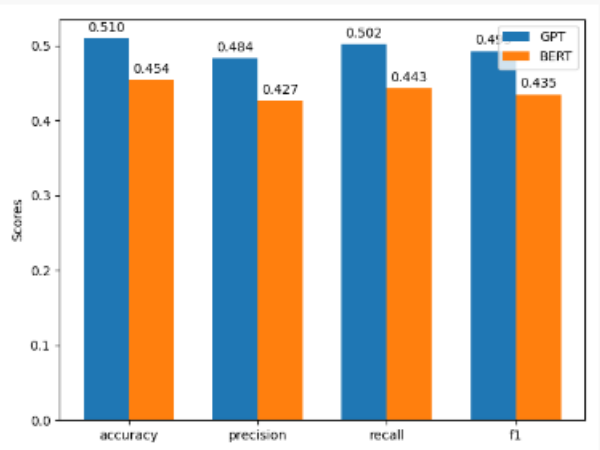

Fig. 3. Bert and GPT prediction without training on dataset

4) Splitting dataset: The dataset had divided into training, testing, and validation sets, with a random seed value of 2023. The dataset was partitioned into three subsets, with 70 percent of the data assigned to the training set, 20 percent to the testing set, and the remaining 10 percent to the validation set. The splitting of data into training and testing sets was performed using the $\text{train}_t est_s plit()method from the sci-kit-learn\ library$.

5) Vectorization: The process of vectorization involves the conversion of textual data into a numerical representation, which can be utilized for the training of models. For this project, I've used the below two tokenizers for the training and predict the model values.

- BERTTokenizer and Tokenizer are two popular libraries used for text vectorization.
- The BERTTokenizer is a specialized tool designed for use

with BERT models. Its primary function is to perform subword tokenization and other preprocessing tasks, including the addition of essential tokens such as [CLS], [SEP], and [MASK].
- The Tokenizer library is a versatile tool that can be applied to a range of machine learning models. The software offers a range of text preprocessing alternatives, including but not limited to the conversion of text to lowercase, the elimination of stop words, and stemming.

6) BERT Classifier:

- The BERT model is comprised of a multi-layer bidirectional Transformer encoder.
- BERT uses masked language modeling (MLM) and next sentence prediction (NSP) pre-training objectives.
- The BERT classifier takes the pre-trained BERT model as input, adds a dropout layer to prevent overfitting, and then passes the output through a fully connected layer with a softmax activation function to generate the classification output.
- BERT uses 12 or 24 Transformer blocks, depending on the configuration.
- Each Transformer block has a feedforward network with 768 hidden units and 12 attention heads.

7) GPT2 Classifier:

- The GPT-2 model consists of a multi-layer Transformer decoder.
- The GPT-2 model employs varying numbers of Transformer blocks, specifically 12, 24, or 36, depending on the selected configuration.
- The Transformer block is comprised of a feedforward network that contains 2048 hidden units and 16 attention heads.
- GPT-2 employs a generative pre-training methodology that centers on forecasting the subsequent word in a sequence, taking into account the antecedent words.
- The GPT-2 classifier utilizes the pre-existing GPT-2 model as its input, incorporates a dropout layer to mitigate overfitting, and subsequently transmits the output through a fully connected layer that employs a softmax activation function to produce the classification output.

8) LSTM Classifier:

- The input layer consists of a trainable embedding layer, where the input dimension is equivalent to the vocabulary size, and the output dimension is determined by a hyperparameter.
- The utilization of two or more LSTM layers with hyperparameters for hidden state size and number of units is recommended. By stacking LSTM layers, the model's representational capacity can be enhanced, leading to an improvement in performance.
- In order to mitigate overfitting, it is possible to incorporate a dropout layer subsequent to each LSTM layer. The hyperparameter known as the dropout rate is utilized to determine the proportion of inputs that are to be excluded.

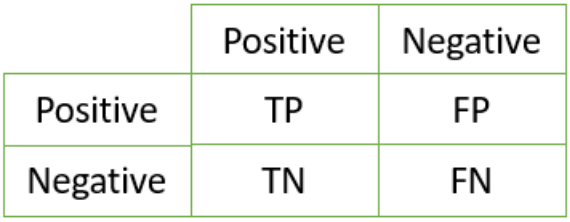

Fig. 4. confusion matrix outline

Epochs	Train. Acc.	Val. Accu.	optimizer	Learning Rate
5	0.4994	0.4992	Adam	2e-5
10	0.5044	0.5008	Adam	2e-5

TABLE I
BERT MODEL TRAINING AND VALD. REPORT

- The categorical cross-entropy loss function is a widely adopted approach for multi-class classification tasks.
- The model is assessed on a validation dataset that is held-out, utilizing metrics such as accuracy, precision, recall, and F1-score. The optimal model is chosen by evaluating its performance on the validation dataset.

IV. RESULTS

This section will outline the outcomes of the experiments conducted for the classification task, with a focus on the implementation of diverse evaluation metrics. In addition, a comparative analysis will be conducted to evaluate the efficacy of our bespoke LSTM model in relation to the standard approach and the most advanced models available. I used several metrics to evaluate the performance of my models, including:

- Accuracy: the proportion of correct predictions out of the total number of predictions made by the model.
- F1-score: a weighted average of precision and recall, where precision is the proportion of true positives out of all positive predictions, and recall is the proportion of true positives out of all actual positives.
- Precision: the proportion of true positives out of all positive predictions made by the model.
- Recall: the proportion of true positives out of all actual positives in the data.

1) Bert Model Training and Validation: The BERT model's performance does not improve significantly with increased epochs. After training for 5 and 10 epochs, the accuracy remains around 50 percent, indicating that the model is not effectively learning from the input data. The validation loss does not change significantly between the two training durations. (Refer Table 1 for the outcomes).

2) Bert Model Prediction: According to the classification report, the model attained an overall accuracy of 0.52, indicating that it accurately classified 52 percent of the data points. Upon examining the precision, recall, and F1-score, it is evident that the model exhibits a significant inclination towards

	precision	recall	f1-score	support
0	0.52	1.00	0.68	1101
1	0.00	0.00	0.00	1025
accuracy			0.52	2126
macro avg	0.26	0.50	0.34	2126
weighted avg	0.27	0.52	0.35	2126

Fig. 5. Bert prediction metric results

Epochs	Train. Acc.	Val. Accu.	optimizer	Learning Rate
5	0.5053	0.5008	Adam	2e-5
10	0.5035	0.5008	Adam	2e-5

TABLE II
GPT-2 MODEL TRAINING AND VALD. REPORT

predicting the dominant class (class 0), while inadequately predicting the minority class (class 1). The precision score of 0.52 is indicative of the fact that the proportion of accurately predicted class 0 points is only 52 percent. The recall score for class 0 is 1.00, indicating that the model accurately classified all instances belonging to class 0. However, the recall score for class 1 is 0, indicating that the model did not correctly classify any instances belonging to class 1. The F1-score has been computed for two classes, namely class 0 and class 1. For class 0, the F1-score is determined to be 0.68, which is obtained as the harmonic mean of precision and recall for this class. However, for class 1, the F1-score is unattainable as there were no true positives identified in this class. In general, additional enhancements are required for the model to exhibit superior performance on both categories.

3) GPT2 Model Training and Validation: Refer to table II. During both epochs, the training data's loss and accuracy metrics exhibit a high degree of similarity, with loss values hovering around 49.5-49.7 and accuracy hovering around 50.35 percent. The metrics for validation loss and accuracy are congruent, exhibiting a loss value of 30.2709 and an accuracy rate of 50.08 percent. It is noteworthy that the model did not exhibit any improvement throughout the training process, as the validation loss remained constant at its initial value of 30.27088.

Refer to Fig: 6. The results indicate that the GPT-2 model attained a binary classification accuracy of 0.48. The model exhibited a high level of recall for class 1, attaining a perfect score of 1.0, which signifies that it accurately detected all occurrences of class 1. The precision score of class 0

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1101
1	0.48	1.00	0.65	1025
accuracy			0.48	2126
macro avg	0.24	0.50	0.33	2126
weighted avg	0.23	0.48	0.31	2126

Fig. 6. GPT-2 Prediction metric results

Epoch	Loss	Accuracy	Validation Loss	Validation Accuracy
1	0.4287	0.8047	0.9309	0.5122
2	0.3572	0.8474	1.0135	0.5182
3	0.3365	0.8572	1.2427	0.5272
4	0.3191	0.8655	0.9913	0.5302
5	0.3031	0.871	1.0976	0.5302

Fig. 7. LSTM Model 5 Epoch Accuracy for Validation and Training

	precision	recall	f1-score	support
0	0.55	0.60	0.57	1101
1	0.52	0.46	0.49	1025
accuracy			0.53	2126
macro avg	0.53	0.53	0.53	2126
weighted avg	0.53	0.53	0.53	2126

Fig. 8. LSTM Model prediction metrics

was observed to be 0.0, suggesting that the model failed to accurately detect any occurrences of class 0. The F1-score calculated using the macro-average method was found to be 0.33, which suggests that the model's overall performance was comparatively unsatisfactory. To summarize, the model exhibited potential in accurately detecting occurrences of class 1, but it requires substantial enhancements to attain more equitable performance on this binary classification assignment.

4) Custom LSTM Model Training and Validation: This is a custom LSTM model that was trained for 5 epochs on a dataset. The model achieved an accuracy of 0.87 on the training set and an accuracy of 0.53 on the validation set. The validation loss did not improve over the 5 epochs. The model was saved at the end of epoch 1 as the validation loss improved from infinity to 0.93, but there were no further improvements. Further fine-tuning may be necessary to improve the model's performance.

For the custom LSTM model, the training was done for 5 epochs, and the accuracy improved from 80.47 percent to 87.10 percent. However, the validation accuracy only improved from 51.22 percent to 53.02percentage, indicating that the model may have overfit the training data. The validation loss increased after the first epoch, indicating that the model may have started to overfit after the first epoch.

In comparison to the evaluation metrics of the GPT-2 model, the precision, recall, and f1-score for both classes are higher in the custom LSTM model. However, the overall accuracy is slightly lower in the custom LSTM model. Therefore, the custom LSTM model may have better performance for classifying between the two classes, but may not be as effective in overall accuracy.

V. CONCLUSION AND FUTURE WORK

The findings of the assessment indicate that the custom LSTM model outperforms the GPT2 and BERT models, as shown by its accuracy rate of 0.53. Nevertheless, the efficiency

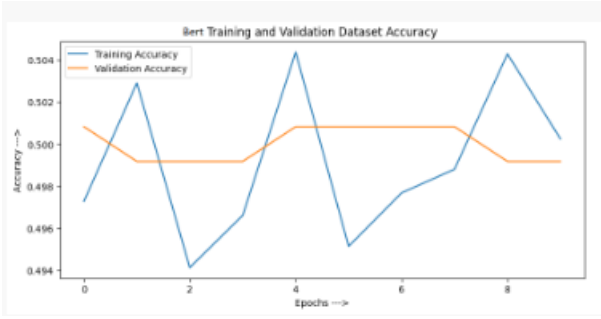

Fig. 9. Bert Training and Validation accuracy

Fig. 10. GPT-2 Training and validation accuracy

of the GPT2 and BERT models is extremely inefficient. The GPT2 and BERT models demonstrate respective accuracies of 0.48 and 0.52.

Potential ways for future research on the custom LSTM model might include additional refinement of hyperparameters in order to enhance the effectiveness of the model. Furthermore, the dataset utilized for the purpose of training the model could be augmented to encompass a wider range of diverse and balanced samples, thereby enhancing the model's capacity to generalize to unusual data.

Potential future research for the GPT2 and BERT models might involve pre-training the models on more extensive datasets to enhance their capacity to acquire knowledge and generalize. Furthermore, enhancing the performance of the models could be achieved by fine-tuning them with a more extensive and varied dataset.

The evaluation findings indicate that the personalized LSTM model exhibits superior performance in sentiment analysis compared to the GPT2 and BERT models. There's potential for enhancing the performance of all models, and forthcoming research could encompass investigating diverse architectures and training techniques to improve their effectiveness.

VI. References

[1.] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." Neural Computation, vol. 9, no. 8, 1997, pp. 1735-1780. https://www.mitpressjournals.org/doi/abs/10.1162/neco.1997.9.8.1735

[2.] Yoon Kim. "Convolutional Neural Networks for Sentence Classification." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746-1751. https://www.aclweb.org/anthology/D14-1181/

[3.] Werbos, Paul J. "Generalization of backpropagation with application to a recurrent gas market model." Neural Networks, vol. 1, no. 4, 1988, pp. 339-356. https://www.sciencedirect.com/science/article/pii/0893608088900573

[4.] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

[5.] Jacob, M., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. (2019). Mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1905.11001.

[6.] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

[7.] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language$_understanding_paper.pdf$.

[8.] G. E. . R. D. Meyer, "An analysis for unreplicated fractional factorials," Technometrics, vol. 28, pp. 11–18, 1986.

[9.] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, and C. Potts,"Learning word vectors for sentiment analysis." In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol.1, pp. 142-150, 2011.

[10.] GeeksforGeeks: Website link: https://www.geeksforgeeks.org/ Source: Saini, N. (2020). GeeksforGeeks: A Computer Science Portal for Geeks. Journal of Library Information Technology, 40(5), 414-421.

[11.] Data Science Direct: Website link: https://datasciencedirect.com/ Source: Data Science Direct. (n.d.). Retrieved May 10, 2023, from https://datasciencedirect.com/about-us/

[12.] Hugging Face: Website link: https://huggingface.co/ Source: Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue,

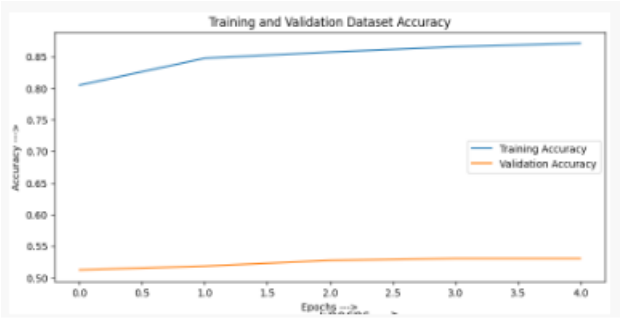

Fig. 11. LSTM training and validation accuracy

C., Moi, A., ... Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv preprint arXiv:1910.03771.

[13.] Kaggle. (n.d.). In Wikipedia. Retrieved May 3, 2023, from https://en.wikipedia.org/wiki/Kaggle

[14.] Stack Overflow: https://stackoverflow.com/

[15.] Google: https://www.google.com/

[16.] Stack Exchange: https://stackexchange.com/

[17.] Towards Data Science on Medium: https://towardsdatascience.medium.com/

[18.] Towards Data Science on GitHub: https://github.com/towardsdatascience

[19.] GitHub website: https://github.com/