Text Classification on IMDB dataset

Baldeep Dhada

Karthiga Sethu Sethuramalingam

Somya Nagar

The University of British Columbia Okanagan

DATA 586 Advanced Machine Learning

Dr. Shan Du

April 26, 2024

Summary

In today's digital landscape, an estimated 200 TB of text data is generated globally on a daily basis, primarily through commenting and reviews on social media platforms. This sheer volume of unstructured text presents a challenge for manual analysis, hindering the extraction of valuable insights. We're delving into text classification techniques to tackle this issue.

Our focus lies on the IMDB dataset sourced from Kaggle (N, n.d.), where we're leveraging state-of-the-art Transformer-based methodologies such as BERT and roBERTa. These advanced techniques, including Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT Pre-training Approach (roBERTa), equip us with the tools to effectively organize and comprehend this vast trove of data.

To establish a benchmark for our investigation, we've scrutinized a research paper from 2016 which employed Recurrent Neural Network (RNN)-based techniques for text classification across diverse datasets, providing us with a foundational reference point.

Our primary objective is to evaluate the performance of Transformer-based models in comparison to RNN-based models, specifically in terms of classification accuracy. Additionally, we're keen on identifying which specific Transformer model yields the most promising results for our specific task. Through this comprehensive analysis, we aim to unlock valuable insights and enhance our understanding of text classification methodologies in the context of vast and varied textual datasets.

Introduction

We began by diving into a research paper titled "Recurrent Neural Network for Text Classification with Multi-Task Learning" by Pengfei Liu, Xipeng Qiu, and Xuanjing Huang (Liu, Qiu, and Huang 2016). This paper explored using Recurrent Neural Network (RNN) techniques on various datasets, including SST-1, SST-2, SUBJ, and IMDB. We chose to focus on the IMDB dataset as it's readily available on Kaggle and easy to grasp.

The goal of the research paper was to apply neural network based methods to natural language processing tasks, and in the case of the IMDB dataset, it was text classification. Their goal was to classify the text data in the “review” column into categories “positive” or “negative”. It provided useful applications like classifying reviews into categories to gauge the audience’s response for a movie.

In this paper, they had implemented multi-task learning to maximize the potential of deep neural network (DNN) models. It is difficult to train such a model because it does not generalize well with limited data. Also, the costs are extremely expensive to build the large scale resources for some NLP tasks. Hence, they proposed the use of multi-task learning to solve that issue.

Multi-task learning is an approach where a model is trained to perform multiple tasks simultaneously, rather than just one. The main idea behind multi-task learning is that by jointly learning multiple tasks, the model can use the shared knowledge and correlations among the tasks to improve performance on each individual task. In this paper, the authors proposed three different models of sharing information with recurrent neural networks (RNN). All the related tasks are integrated into a single system which was then trained jointly.

The method implemented in this paper performed well, as evident in the results section of the paper. They incorporated results from each type of architecture which showed a high accuracy with the IMDB dataset. Our goal was to implement a more recent model on the same dataset to try and achieve better performance and accuracy with the same dataset.

We researched several more recent methods than the one implemented in the research paper. We looked into transformer based models, models like BERT and its extensions. We also looked into

Graph Neural Networks and tried to learn about which model would help us get better performance and accuracy with the IMDB dataset.

After wrangling and splitting the dataset into training, evaluation, and test sets, we embarked on experimenting with both BERT and roBERTa models. Through hyperparameter tuning and utilizing the built-in trainer functionality, we aimed to compare the performance of these advanced Transformer-based techniques with traditional RNN-based methods.

Methodology

Data Inference methodology:

We warangled the data using basic Python methods from libraries such as pandas, numpy, and missingno and then we focused on optimizing our model's performance through hyperparameter tuning. As highlighted by Zheng and Casari (2018), fine-tuning parameters such as learning rates, epochs, and dropout rates are crucial for effectively comparing models. By systematically adjusting these parameters, we aimed to enhance our model's predictive accuracy and generalization capabilities

BERT:

BERT (Bidirectional Encoder Representations from Transformers) is a state-of-the-art natural language processing (NLP) model introduced by Google in 2018. It revolutionized the field by capturing contextual information bidirectionally from text data.

Unlike traditional models that process text sequentially, BERT employs a transformer architecture, enabling it to consider the entire context of a word by looking at both the left and right contexts simultaneously. This bidirectional approach allows BERT to better understand the nuances and dependencies within a sentence. BERT is pre-trained on a large corpora of text data using two unsupervised tasks: masked language modeling (MLM) and next sentence prediction (NSP). During MLM, BERT randomly masks some words in a sentence and learns to predict the masked words based on the surrounding context. In NSP, BERT learns to predict whether two sentences are consecutive or not, helping it understand relationships between sentences.

After pre-training, BERT can be fine-tuned on specific downstream tasks, such as text classification, question answering, or named entity recognition. Fine-tuning involves adjusting BERT's parameters on a task-specific dataset to adapt its learned representations to the nuances of the target task.

One of the key advantages of BERT is its ability to capture deep contextual information, resulting in impressive performance across a wide range of NLP tasks. Its pre-trained representations can be fine-tuned with relatively small amounts of task-specific data, making it highly versatile and applicable to various real-world scenarios.

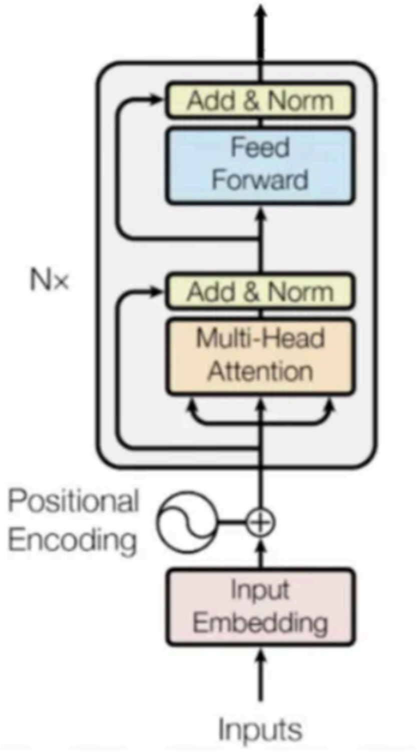

Fig 1. BERT Encode from (Calvo 2018). The structure denotes one encoder’s structure architecture in BERT. Usually BERT and its variation has multiple encoders stacked upon each other.

roBERTa:

It is an extension of the BERT model introduced by Facebook AI in 2019. It builds upon BERT's architecture and training methodology, aiming to improve performance and robustness.

Similar to BERT, roBERTa utilizes a transformer architecture, enabling it to capture bidirectional contextual information from text data. However, roBERTa incorporates several enhancements and optimizations to further enhance its capabilities.

One notable difference is in the pre-training process. roBERTa uses a larger training corpus and trains for longer durations compared to BERT. It also removes the next sentence prediction (NSP) task used in BERT's pre-training, focusing solely on masked language modeling (MLM). This alteration allows roBERTa to better capture deep contextual information and dependencies within sentences.

Additionally, roBERTa employs dynamic masking during MLM, where different masks are applied to each training instance. This dynamic masking strategy helps roBERTa learn more effectively from the training data, improving its ability to generalize to unseen examples.

Furthermore, roBERTa incorporates other optimizations such as larger batch sizes, longer sequences, and training on more diverse data sources. These optimizations contribute to roBERTa's enhanced performance and robustness across various natural language processing tasks.

Overall, roBERTa represents a significant advancement in transformer-based NLP models, offering improved performance and robustness compared to its predecessor, BERT. Its ability to capture deep contextual information and generalize effectively to diverse tasks makes it a powerful tool in the field of natural language understanding and processing.

Model comparison metrics:

After reading the article from IBM website (IBM 2021), we used accuracy, precision, f1 score and recall to compare the models. Accuracy is a metric that quantifies how often a model correctly predicts the outcome. Precision shows how often a model is correct when predicting the target class. Recall is a true positive rate that measures how often predictions for the positive class are correct. On a higher level low recall indicates a need to add more training data. Finally, F1 score is a weighted average of the precision and recall values to strike the balance between precision and recall. All four metrics were used to provide a comprehensive evaluation of a model's performance.

Experiments

Data Inference:

The dataset contains 50,000 rows which contain proportional positive and negative reviews with 25,000 rows each. Since the data is balanced, sampling techniques were not necessary. Positive and negative sentiment labels were converted to 1 and 0. Using the train_test_split function, we split the data into a 70/15/15 (train/test/evaluation) set.

BERT:

We used a pre-trained bert-base-uncased model from AutoModel in the transformers library and 'bert-base-uncased' tokenizer from BertTokenizerFast in the transformer library. Bert-base-uncased refers to a specific pre-trained BERT model that trained on a large corpus of text data using a particular architecture. On the other hand, a generic "base model" could refer to any base model within the BERT family. It comes with pre-trained weights that have been fine-tuned during the pre-training process using a large set of text data. It uses a specific vocabulary that includes lowercase tokens only, meaning all the input text is converted to lowercase before processing.

The layers of our model consist of pre-trained BERT layers followed by dropout regularization to mitigate overfitting. Subsequently, ReLU activation functions are applied. The final layers comprise of two linear layers and a log softmax layer. Without any fine-tuning, our baseline accuracy stands at 66%.

Subsequently, we experimented with three different learning rates: 1e-3, 2e-5, and 0.01. Both 1e-3 and 2e-5 yielded equally satisfactory results, while 0.01 led to significantly poorer performance. Further, when employing a learning rate of 1e-3, two different dropout rates (0.5 and 0.8) were tested, both of which resulted in a notable decrease in accuracy. Moving forward, with a learning rate of 1e-3 and dropout rate of 0, we conducted experiments with four epochs (1, 10, 50, and 100) and increasing the epoch didn't bring significant change in accuracy.

The accuracy increased however the gain didn't increase more than 2-3%. So as a final step we used the Trainer module from transformers. This inbuilt trainer model comprises approximately

111 million parameters, distributed across its various components. It includes 23.8 million parameters in the embeddings (word, position, and token type), 86.4 million parameters in the 12 Transformer encoder layers, 590,592 parameters in the pooler layer, and 1,538 parameters in the classifier layer. This architecture, featuring 12 layers, is tailored for sequence classification tasks and is well-suited for applications like sentiment analysis and text classification. Despite system limitations restricting us to only one epoch, we successfully increased accuracy to around 93%. However, further experimentation beyond a single epoch was not feasible due to system constraints leading to crashes.

roBERTa:

A pretrained model of roBERTa (robertaForSequenceClassification) was imported from the transformers library. This model is pretrained on 2.5 TB of filtered CommonCrawl data. robertaTokenizer is used for the tokenization process for roBERTa and imported from the transformers library. The roBERTa Tokenizer masks tokens differently at each epoch. roBERTa also uses byte pair encoding which splits a word into pairs of characters resulting in a larger vocabulary size compared to BERT This is based on the robust roBERTa architecture enhanced with prefix tuning mechanisms. The model consists of a roBERTa encoder with 12 layers, each comprising self-attention and feed-forward neural network blocks. Additionally, a classification head is attached to the encoder to predict the target label. The model uses a dropout rate of 0.1 for both the classification head and for the base model. Since roBERTa trains longer we were able to train on a single epoch before running out of GPU usage on google colab, our accuracy is 95% without fine tuning. The total number of parameters in this model is 117,441,568.

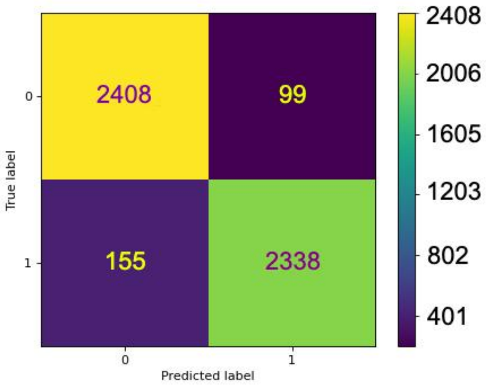

Fig. 3. Confusion Matrix of the roBERTa model. True label represents the sentiment label from the testing set. The predicted label is the predicted sentiment from the model. 0 represents negative sentiment and 1 represents positive sentiment. The bar ranges from 0 (dark purple) - 2408 (yellow) and represents the range of classifications. The boxes are colored according to the range of the classification. 2408 negative labels were correctly predicted as negative, 2338 positive labels were correctly predicted as negative, 99 negative labels were incorrectly classified as positive, 155 positive labels were incorrectly classified as negative.

Issues Faced:

The major issues we encountered during this project were during training these advanced machine learning models due to the limited GPU resources available to us. It required high GPU availability and training time which posed a challenge to complete the tasks outlined in this project. Using a pre-trained model to perform classification on a specific dataset requires a lot of fine tuning of hyper-parameters so that the model learns the intricacies in the dataset and improves its predictions over time. This involves a lot of trial and error with different parameters like “learning rate” and the “number of epochs”. Due to limited computational resources, this process was difficult.

We also tried experimenting with another technique based on BERT called DistilBERT. DistilBERT is generally faster than Bert because it has fewer parameters. We were not able to implement it successfully because of lack of time for the implementation and debugging. We will try to incorporate this model in this project as part of future work.

Conclusion and Discussion

Result table:

Metrics	RNN model from paper	BERT	roBERTa
Accuracy	91.7%	93.62%	95.0%
Precision	-	93.37%	95.0%
Recall	-	98.92%	95.0%
F1 score	-	93.64%	95.0%
Epoch	-	1	1

Table 1. Table representing the computational metrics of RNN, BERT and roBERTa. The first column represents metrics used for evaluation, while the second, third, and fourth columns correspond to scores obtained from RNN, BERT, and roBERTa models. Accuracy measures the proportion of correctly classified instances. Precision measures the proportion of true positive predictions out of all positive predictions. Recall measures the proportion of true positive predictions out of all actual positive instances. The F1 score is the harmonic mean of precision and recall. An epoch refers to a single pass of the entire dataset through the model during the training phase.

Considering Table 1, in terms of accuracy, BERT shows a significant improvement in accuracy compared to the initial RNN network, achieving an accuracy of 93.62%. roBERTa further improves accuracy to 95.0%, surpassing both the RNN and BERT models. Both BERT and roBERTa demonstrate high precision. However, roBERTa performed comparatively better. Next, delving into recall, BERT achieves a very high recall rate of 98.92%, indicating its ability to correctly identify most positive instances. Looking into F1 score, roBERTa is better than BERT indicating a balanced performance between precision and recall.

Overall, both BERT and roBERTa outperform the initial RNN network on the IMDB dataset in terms of accuracy. roBERTa demonstrates a slight improvement over BERT in accuracy, precision, and F1 score, indicating its effectiveness in text classification tasks. roBERTa's enhanced performance over BERT may stem from several factors. Firstly, roBERTa has a larger parameter count of nearly 117 million compared to BERT's 111 million (approx.), enabling it to capture more nuanced features in text data. Additionally, roBERTa benefits from training on a larger and more diverse dataset, which enhances its ability to generalize to various tasks like sentiment analysis on the IMDB dataset. Furthermore, advancements in pre-training techniques, including dynamic masking and extended training durations, contribute to roBERTa's superior performance. Overall, roBERTa's combination of increased parameters, diverse training data, and improved pre-training methods results in its impressive performance in text classification tasks.

Future Work

In our future work, we plan to extend our binary classification task to encompass multi-class classification, with the goal of improving accuracy across various classes. This entails exploring alternative models such as DistilBERT, fine-tuning hyperparameters, and employing advanced techniques to enhance model performance within the context of multimodal classification.

References

A, Zheng, and Casari A. 2018. “Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists.”

Calvo, Miguel R. 2018. “Dissecting BERT Part 1: The Encoder | by Miguel Romero Calvo | Dissecting BERT.” Medium. https://medium.com/dissecting-bert/dissecting-bert-part-1-d3c3d495cdb3.

IBM. 2021. “IBM Cloud Docs.” IBM Cloud Docs. https://cloud.ibm.com/docs/watson-knowledge-studio-data?topic=watson-knowledge-studio-data-evaluate-ml.

Liu, Pengfei, Xipeng Qiu, and Xuanjing Huang. 2016. “Recurrent Neural Network for Text Classification with Multi-Task Learning.” *arXiv:1605.05101*, (5). https://arxiv.org/abs/1605.05101.

N, LAKSHMIPATHI. n.d. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

Contribution

Our contribution as part of this project toward the paper selected is that we implemented some newer neural network based models like Bert and roBERTa to improve the accuracy of the text classification task on the IMDB dataset.

We divided the analysis evenly so that we can analyze them further and exchange our ideas about the advantages and disadvantages of each one. Specifically, Baldeep was in charge of roBERTa, presentation preparation. Karthiga worked on BERT, hyperparameter tuning and presentation preparation and Somya focused on researching on the paper selected, presentation preparation and implementing an extension of BERT - DistillBERT. We did not include results from the DistillBert implementation in the final report because of the limited time and computational resources, we will pick that as part of future work.