Performance Comparison of Various BERT Models on Binary Classification Tasks

Yunrong Liu[*]
yunrong.liu@mail.utoronto.ca
Kuang Ren
kuang.ren@mail.utoronto.ca
Bonus distribution: Yunrong Liu:20%, Kuang Ren:10%

April 12, 2023

Abstract

The Bidirectional Encoder Representations from Transformers (BERT) model, proposed by Devlin and his colleagues, has demonstrated state-of-the-art performance on multiple natural language processing tasks (Devlin et al. 2019). In this study, we tested different BERT-based models, including BERT, RoBERTa, XLM, and ALBERT, on binary classification tasks using the SST, CoLA, and MRPC datasets. Results varied across models and datasets, emphasizing the importance of selecting the right model for a specific task. For instance, XLM achieved the highest accuracy on the SST dataset (0.951), while RoBERTa performed best on the MRPC dataset (0.833). However, on the CoLA dataset, BERT achieved the highest F1 score (0.723), closely followed by RoBERTa (0.746). Our research aims to compare the strengths and weaknesses of different BERT-based models to provide insights for improving future NLP models.

1 Background

Natural language understanding is a fundamental task of artificial intelligence with a wide range of applications in areas such as machine translation, sentiment analysis, and chatbots. However, it is also a challenging problem due to the complexity of language, which exhibits many nuances requiring deep contextual and semantic understanding. Traditional models, such as RNN(recurrent neural networks) and CNN(convolutional neural networks), were not efficient in capturing long-range dependencies, making them less effective for NLP(natural language processing) tasks. In contrast, the Transformer architecture invented by Vaswani et al, which relies on self-attention mechanisms to attend to different

[*] TEAM: We donno whaaat we r doin, team leader

parts of the input sequence and create a better representation, was introduced as a more effective solution for NLP tasks(Vaswani et al., 2017).
The main contribution of Transformer-based models is the pre-training technique, where the model is trained on a large corpus of text to learn the language's representation. BERT, RoBERTa, XLM, and ALBERT are examples of pre-trained models that have shown state-of-the-art performance in various NLP tasks. However, it is important to note that their performance is not always consistent and may vary depending on the task. For instance, in a machine translation task, the Transformer architecture is able to capture complex relationships between words in the source and target languages, resulting in more accurate translations. On the other hand, if the input text has significant noise or if it is from a domain that the model has not been trained on, the performance of the Transformer model may be negatively impacted. For example, if the model has been trained on news articles but is applied to informal social media text, it may struggle to generate coherent and contextually accurate responses.
Given that Transformer-based models have demonstrated impressive performance in various NLP tasks, it is essential to understand that their performance may not always be linearly related and may require further investigation to optimize their use. In this paper, we explore the existing research on various Transformer-based models, including BERT, RoBERTa, XLM, and ALBERT, and compare their performance in the fine-tuning stage on a variety of NLP tasks. To this end, we conduct comparative experiments to infer changes in performance based on the models' properties. By doing so, we aim to contribute to the development of more effective and efficient natural language processing solutions.

1.1 Transformer

As the algorithms examined and evaluated in this paper are variants of BERT that are based on the transformer architecture, a brief introduction to the transformer is deemed essential. Transformers are a type of neural network architecture that has garnered significant attention in recent years due to their impressive performance on various natural language processing (NLP) tasks(Vaswani et al., 2017). At a high level, a transformer consists of two essential components: an encoder and a decoder, both of which contain multiple layers of self-attention and feedforward neural networks.

The encoder component is responsible for processing the input sequence, which it does through a series of N identical layers, each of which contains two sub-layers. The first sub-layer is a multi-head self-attention layer that treats the embedded input as both query and key to compute the attention scores. The second sub-layer is a position-wise fully connected feedforward network.
The decoder component, on the other hand, generates the output sequence and contains N identical layers, each of which has three sub-layers. The first sub-layer is a masked multi-head self-attention layer that allows the decoder to only attend to previous positions in the output sequence. The second sub-layer is another multi-head attention layer that uses the encoder outputs as its query and key. Finally, the third sub-layer is a position-wise fully connected feedforward

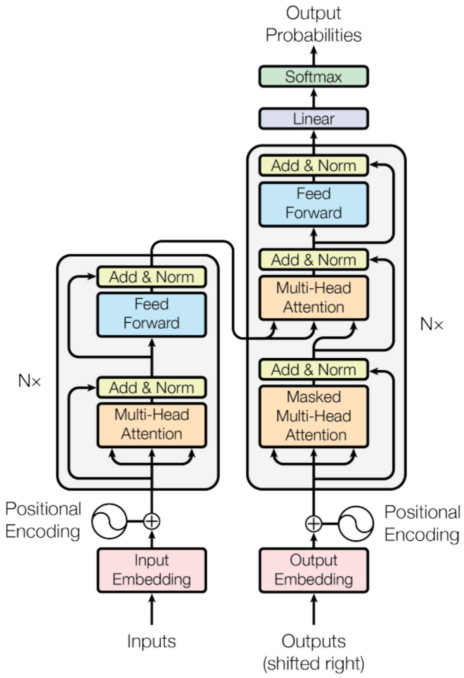

Figure 1: Transformer(Vaswani et al., 2017)

network.
The final output of a transformer model is typically generated through a linear transformation followed by a softmax activation function. The linear transformation maps the output of the last decoder layer to a vector in the same dimensionality as the vocabulary size, where each element in the vector corresponds to the probability of the corresponding word in the vocabulary being the next word in the output sequence.
All sub-layers of the encoder and decoder components are combined with skip connections to facilitate training and prevent the vanishing gradient problem. Additionally, the transformer uses positional encoding to capture the positional information of the input sequence, which is critical for modeling the order of the words in the sequence. Shown in figure 1

2 Variations of Transformer

We selected four prominent language models, namely BERT, RoBERTa, XLM, and ALBERT, for the purpose of comparing their respective performance. Each of these models has achieved state-of-the-art results on various natural language processing (NLP) tasks, and they have been widely adopted by the research community and industry alike.

2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a deep learning model introduced by Google in 2018 that has achieved state-of-the-art performance on a wide range of natural language processing tasks(Devlin et al. 2019). BERT is a type of transformer model, which is a neural network architecture that allows for parallel processing of input sequences. Unlike traditional language models that process text in a unidirectional way, BERT is a bidirectional model that considers both the left and right context of each word in a sentence.
The algorithm of bert can be divide into two parts, pre-training and fine-tuning:
The pre-training stage is unsupervised, where the model learns the language's representation by predicting the masked words and next sentence prediction tasks.
The fine-tuning stage involves training the pre-trained model on specific NLP tasks, such as sentiment analysis or question answering. The pre-trained model's weights are fine-tuned on the specific task's dataset, and the model is then used to predict the task's output. Shown in figure 2

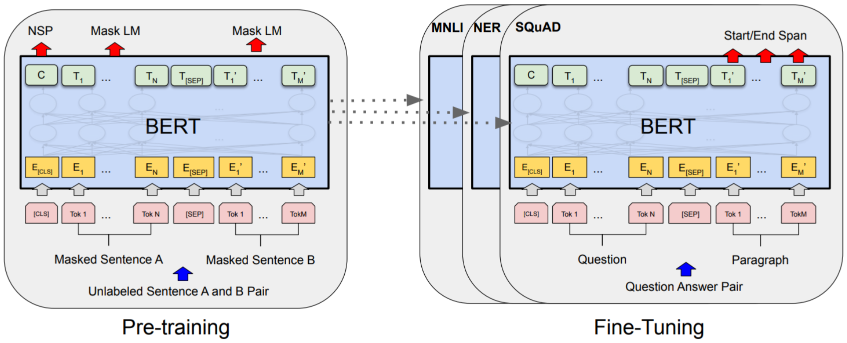

Figure 2: BERT

During the pre-training stage of BERT, sentences are processed with sentence structure markers to indicate sentence boundaries. BERT then undergoes Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) steps.
In the MLM stage, BERT masks 15% of the input words, replacing 80% of them with [MASK], 10% with a random word, and leaving 10% unchanged. It learns to predict the original masked words using a bidirectional approach based on transformer architecture, which considers both left and right context, unlike other language models. During the NSP stage, BERT trains the model to predict whether a sentence B follows sentence A, and labels the pairs as either **IsNext**

or **NotNext**.
In the fine-tuning stage, BERT simply takes in specific inputs (in this case, binary classification of sentences) and applies them to the pre-trained model, adding a softmax layer to generate the desired output.

2.2 ROBERTA

Roberta is a modified version of the BERT model (Liu et al., 2019). The modifications made to the original BERT model are straightforward, which includes training the model for a longer period, with larger batches and on more data, removing the objective of predicting the next sentence(NSP), training the model on longer sequences, and dynamically changing the masking pattern used during training.

2.3 XLM

Cross-lingual language models (XLM) and BERT share a common framework that consists of both pre-training and fine-tuning stages(Lample & Conneau, 2019). However, XLM is distinguished by its ability to handle multiple languages. The pre-training phase of XLM comprises two critical components: Masked Language Modeling (MLM) and Translation Language Modeling (TLM). MLM is akin to the masked language modeling technique employed by BERT and is based on the transformer architecture, with the additional inclusion of a language embedding indicating the language of each word. Notably, in the MLM pre-training stage of XLM, sentences are sourced from the same language. Conversely, the TLM objective expands MLM to pairs of parallel sentences. This objective aims to harness the shared semantic representations between corresponding sentences in distinct languages, thereby enhancing the model's capability to comprehend and translate across languages.
The fine-tuning stage of XLM involves training the model on specific downstream tasks, such as text classification or sequence labeling, as typically evaluated in empirical studies. During fine-tuning, the pre-trained XLM model is further optimized to enhance its performance on the specific task of interest. This process involves initializing the model parameters with the pre-trained weights and then updating them through backpropagation using task-specific loss functions.
An advantage of using XLM for fine-tuning is its capacity to handle multiple languages, enabling efficient training and improved performance on cross-lingual tasks. Moreover, XLM supports zero-shot transfer learning, facilitating the transfer of knowledge across languages without requiring any language-specific fine-tuning. Shown in figure 3.

2.4 ALBERT

ALBERT is a pre-trained language model introduced by Lan et al. that utilizes two parameter reduction techniques, namely factorized embedding parameterization and cross-layer parameter sharing, to improve its parameter-efficiency

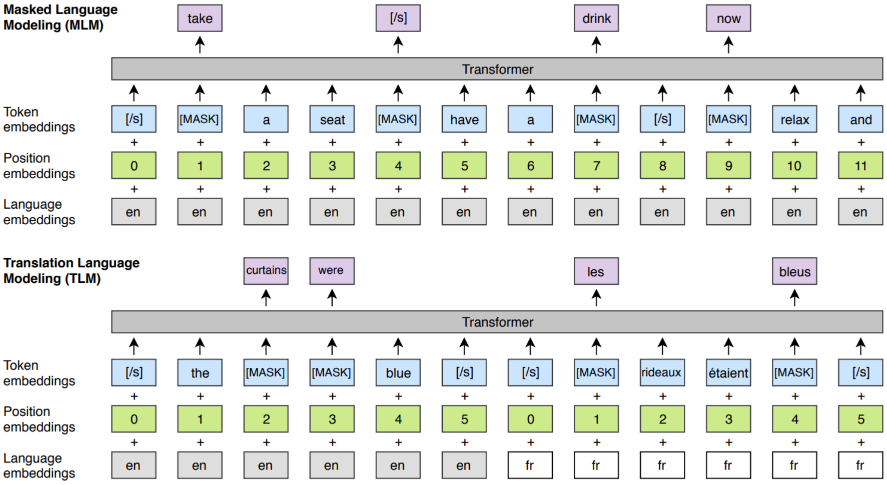

Figure 3: XLM

and serve as a form of regularization(Lan et al., 2020). In addition, ALBERT introduces a self-supervised loss for sentence-order prediction (SOP) to address the shortcomings of the next sentence prediction (NSP) loss used in BERT and improve inter-sentence coherence.

Factorized embedding parameterization involves untying the WordPiece embedding size (E) from the hidden layer size (H) in BERT. Instead of directly projecting the one-hot vectors into the hidden space of size H, we first project them into a lower dimensional embedding space of size E, and then project it to the hidden space, thus significantly reducing the number of embedding parameters when H » E.

Cross-layer parameter sharing is another technique used to enhance parameter efficiency in ALBERT. Various ways to share parameters exist, including only sharing feed-forward network (FFN) parameters across layers or only sharing attention parameters. However, the default option in ALBERT is to share all parameters across layers.

In order to enhance the effectiveness of inter-sentence modeling, ALBERT dispenses with the next sentence prediction (NSP) stage and introduces a loss function that emphasizes coherence modeling. Specifically, we adopt a sentence-order prediction (SOP) loss that prioritizes inter-sentence coherence over topic prediction. Unlike NSP, which employs a positive example consisting of two segments that occur consecutively in the training corpus and a negative example created by randomly pairing segments from different documents, SOP employs a negative example created by reversing the order of the positive example's two consecutive segments. Shown in figure 4

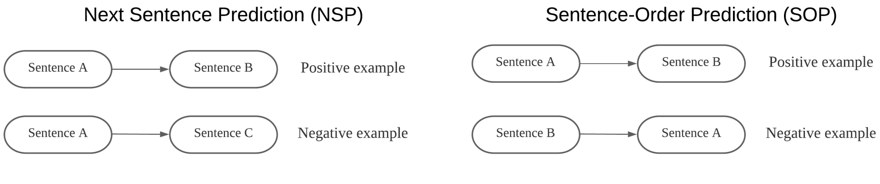

Figure 4: NSP and SOP
Sentence A and sentence B are consecutive, whereas sentence A and sentence C, or reversing the order of A and B, does not result in consecutive sentences.

3 Test and result

3.1 Choosing data sets

We choose three datasets for comparison:

SST: The SST (Stanford Sentiment Treebank) dataset consists of movie review sentences with human-annotated sentiment labels. The sentence lengths are variable, and some sentences are quite long while others are short(Socher et al., 2013).

CoLA: The CoLA (Corpus of Linguistic Acceptability) dataset consists of English sentences that are grammatically correct or not. The sentences are relatively short and contain various types of grammatical errors, such as gender mismatches, missing words, or extra words(Warstadt et al., 2019).

MRPC: The MRPC (Microsoft Research Paraphrase Corpus) dataset contains pairs of long sentences, where each pair may be either a paraphrase or not. The dataset is imbalanced, with 68% of the samples being positive and 32% being negative(Dolan & Brockett, 2005).

These datasets are chosen for different reasons. SST is a popular dataset for sentiment analysis tasks, and the human annotations provide a high-quality source of labeled data. CoLA is used to evaluate models' ability to handle grammatically correct sentences, which is important for natural language processing tasks. MRPC is used to evaluate models' ability to identify paraphrases, which is essential for tasks such as question answering or machine translation. All three datasets provide unique challenges to machine learning models and are widely used for evaluating the performance of models in natural language processing tasks.

3.2 Setup

To ensure a fair comparison of the BERT, RoBERTa, XLM, ALBERT on binary classification tasks, we build a consistent set up for these 4 models. The model code is from YJiangcm. (n.d.).SST-2 sentiment analysis[6]]. The YJiangcm code builds a sample BERT, RoBERTa,XLM,ALBERT model. We used it to test with our data set: SST, COLA, MRPC. The cleaned data set and modified model

is uploaded in GitHub repository[11]]. The work we done follows the process: Data processing, Model Tuning, Evaluation.
In order to compare different models' performance in different data set. For BERT model, we fix the parameter and only change the data set. Similarly, for RoBERTa, XLM, ALBERT, we fix the parameter and only change the data set. Below is the test result table.

3.3 Test Result

Table 1: The comparison Between Algorithm and Results

	ALGORITHM			
DATASET	BERT	ROBERTA	XLM	ALBERT
SST	Accuracy: 0.921 Precision: 0.922 Recall: 0.921 F1: 0.921	Accuracy: 0.946 Precision: 0.946 Recall: 0.946 F1: 0.946	Accuracy: 0.951 Precision: 0.951 Recall: 0.951 F1: 0.951	Accuracy: 0.897 Precision: 0.913 Recall: 0.897 F1: 0.896
COLA	Accuracy: 0.795 Precision: 0.798 Recall: 0.703 F1: 0.723	Accuracy: 0.791 Precision: 0.761 Recall: 0.737 F1: 0.746	Accuracy: 0.686 Precision: 0.343 Recall: 0.500 F1: 0.407	Accuracy: 0.686 Precision: 0.343 Recall: 0.500 F1: 0.407
MRPC	Accuracy: 0.749 Precision: 0.713 Recall: 0.727 F1: 0.718	Accuracy: 0.833 Precision: 0.824 Recall: 0.773 F1: 0.791	Accuracy: 0.796 Precision: 0.769 Recall: 0.741 F1: 0.751	Accuracy: 0.727 Precision: 0.774 Recall: 0.575 F1: 0.554

4 Comparison and Analysis

The SST dataset evaluates the model on sentiment analysis tasks. With the XLM has the highest accuracy of 0.951, and the RoBERTa, BERT,ALBERT follows and accuracy of 0.946,0.921,0.897. All of them performs well and have similar accuracy. When coming into COLA dataset, the performance varies. Only BERT and ROBERTA have acceptable accuracy:0.975,0.971. While the XLM and ALBERT have extreme low accuracy. Coming into MRPC, BERT,ROBERTA,XLM have acceptable accuracy along with other metric. But ALBERT only has acceptable accuracy and precision, other metrics are unacceptable. There could be several reason that cause the difference between these model. Although shown in result, the RoBERTa and XLM is better than BERT. We do observe that ALBERT has a lower performance than others. These result can have potentional reasons, listed below:
1: The size of training data. The performance of the model depends on the size of data set and the type of work we are classifying. We do have same data set in use. But due to the ALBERT's small size model and light structure. If our data

set is not enough, then ALBERT can not capture the detail of the dataset,some of the parameter inside the ALBERT model is not well trained.
2: Model Structure. For example: The ALBERT's light structure allows faster training, but it might lose pattern in the training process. This cause lower performance comparing to BERT. For another model XLM, it uses similar structure as BERT, with adding the architecture of processing multi-language. This can be the reason that it has higher performance than BERT at SST and MRPC data.
3. The type of training work: the SST data set tends to be easier and simpler since it is sentiment analysis. This kind of data set tends to be easier to classified since we have specific words to express the mood. However, CoLa and MRPC tends to be harder training data. With Cola is deciding whether the sentences are grammar correct or not, some of these sentence are even hard to be seen by human, and some of the sentence are controversial. Native speakers may have a different view of these sentences than foreign speakers. For MRPC data, the sentence tends to be longer and imbalanced, makes model harder to capture the true feature.
Below if the table of analysis for the above result, why some model performs well, and why some not.

Table 2: The Analysis Between Algorithm and Results

	ALGORITHM			
DATA	BERT	ROBERTA	XLM	ALBERT
SST	Use as standard	Different masking, No next sentence prediction	Add multi-language structure	Lighter architecture
COLA	Controversial data	Masking, no next prediction makes it harder to learn grammar	Multi language makes it harder to identify grammar correctness	Lighter architecture is not well trained
MRPC	Longer, Imbalanced data	Masking works well for longer data	Multi language works well at longer data	Lighter architecture is not well trained for long data

5 Limitation

Note that these result might not generalized to other NLP tasks, and the performance of them might vary on different task, dataset,tuning process,and training time. The insight we can achieve is that our findings still can show some

strengths and weakness of each model. Which can provide future researchers a short cut into choosing their model for their tasks.
This study has some limitations that should be considered when interpreting the results. Firstly, due to the high requirement for training data and computational resources in Transformer-based models, we used smaller datasets, which may limit the generalizability of our findings to larger datasets.
Secondly, this study focused on four popular Transformer-based models, but there are other variants and adaptations of these models that may have different performance characteristics. The results of this study may not be generalized to these other models.
Also, while we focused on binary classification tasks, other NLP tasks, such as sentence generation and question answering, were not explored in this study.

6 Conclusion And Future Directions

In this study, we provide a comparative analysis of the performance of four transformer-based models-BERT, RoBERTa, XLM, and ALBERT-on a variety of natural language processing tasks. Our results show that while RoBERTa and XLM generally outperform BERT, ALBERT performs lower than BERT on the tasks considered in this study. It is important to recognize that the performance of these models depends on several factors, such as the type and size of the training data, model structure, and tuning process. The performance of these models can be further improved by optimizing the fine-tuning process and incorporating task-specific additional mechanisms or techniques. For future work, we recommend more experiments and analyses to explore additional factors that affects the performance of these models and to determine the best model for each NLP task. By understanding the strengths and weaknesses of each model, researchers can make more informed decisions when selecting and tuning models for their specific tasks.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I. (2017). *Attention is all you need.* arXiv:1706.03762.

[2] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* arXiv:1810.04805.

[3] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019). *RoBERTa: A robustly optimized BERT pretraining approach.* arXiv:1907.11692.

[4] Lample, G., Conneau, A. (2019). *Cross-lingual language model pretraining.* arXiv:1901.07291.

[5] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2020). *ALBERT: A lite BERT for self-supervised learning of language representations.* arXiv:1909.11942.

[6] YJiangcm. (n.d.).*SST-2 sentiment analysis.* GitHub. Retrieved April 8, 2023, from https://github.com/YJiangcm/SST-2-sentiment-analysis

[7] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 64-69).

[8] Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7, 625-641. (CoLA)

[9] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1631-1642). (SST)

[10] Dolan, B., & Brockett, C. (2005). Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005).* (MRPC)

A Appendix

Code used in the article:

[11] Kuma110011. (n.d.). *STAD80 Team: We Don't Know What We're Doing.* GitHub. Retrieved April 8, 2023, from https://github.com/Kuma110011/STAD80-Team-We-donno-whatt-we-doin