2023 3rd International Conference on Intelligent Technologies (CONIT)
Karnataka, India. June 23-25, 2023

2023 3rd International Conference on Intelligent Technologies (CONIT) | 979-8-3503-3860-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/CONIT59222.2023.10205622

Comparative Study of Bert Models and Roberta in Transformer based Question Answering

N Akhila, Sanjanasri J.P, Soman K.P
Center for Computational Engineering and Networking (CEN),
Amrita School of Engineering, Coimbatore,
Amrita Vishwa Vidyapeetham, Coimbatore, India
akhilananduri@gmail.com, jp_sanjanasri@cb.amrita.edu, kp_soman@amrita.edu

***Abstract*—Using deep learning technique, transformer-based self-supervised pre-trained models have revolutionised the idea of transfer learning in natural language processing (NLP). The self-attention mechanism makes transformers more prevalent in transfer learning across broad range of NLP tasks. In this study, seven prominent models are compared., including the Bertbase- uncased, Distilbert-base-cased, Distilbert-base-uncased, and Roberta models, in terms of their effectiveness (using 3 epochs). Bert base cased, Bert-medium-squad2-distilled, and Electra-basesquad2 on the Stanford Question Answering Dataset were utilised models with two epochs (SQuAD). The analysis shows that in three epochs, Roberta models provide the most accuracy, while Distilbert-base cased models provide the maximum accuracy when compared to Bert-base uncased, Distilbert-base uncased, and Distilbert base cased models. While Electra-base-squad2 performs better than Bert base cased and Bert-medium-squad2- distilled in cases with two epoches. Although all Bert and Roberta models take a long time to run, increasing accuracy requires more time to train the dataset.**

***Index Terms*—Natural Language Processing, Bert Models, SQuAD, question-answering, transformer**

I. Introduction

Finding a pertinent piece of information quickly becomes more challenging as the amount of information in daily life grows. As a result, the requested information can be presented effectively using a Question-Answering (QA) system. Question Answering (QA) is a crucial real-world application of NLP, which is a particular kind of information retrieval technique. The task of question answering (QA) in natural language processing has grown significantly since the invention of transformers. Extractive Issue Answering is the process of taking a section of text from a context paragraph and using it as the response to a given question. The task of extractive question responding has been demonstrated to be remarkably successful by many pre-trained language models such as BERT, which utilize the Transformer architecture to develop many language models for a variety of NLP tasks specified by benchmarks, such as GLUE etc. The QA system attempts to automatically identify the contextually and semantically appropriate response to the given textual inquiry. Question classification, information retrieval, and answer extraction/generation are typically the three parts of a QA system.Reading Comprehension is one method to get a machine to answer questions, even though QA is not without its difficulties (RC). It is difficult for machines to read a text and respond to it since it requires both an understanding of natural language and awareness of the outside world. A Question Answering dataset is the initial step in the process to construct such a system. The Stanford Question Answering Dataset, or SQuAD, is a well-known benchmark QA dataset produced by Stanford University. Bidirectional Encoder Representations from Transformers, or BERT, is a deep learning model that is based on Transformers. In Transformers, each output element is connected to each input element, and the weightings between them are dynamically determined based upon their connection. The BERT implementation only executes one static mask. During training, each training sequence is shown four times while wearing the same mask. By creating the mask pattern corresponding to each input sequence that we feed RoBERTa without NSP loss, big minibatches, and a higher byte-level BPE, the model is trained via dynamic masking. It is a robustly optimised approach that outperforms Bidirectional Encoder Representations from Transformers, or BERT, for pre-training natural language processing (NLP) systems. A transformer is a Deep Learning (DL) model that uses the self-attention process and weights the importance of each component of the input data differently. It is largely utilised in the disciplines of computer vision and natural language processing (NLP) (CV).

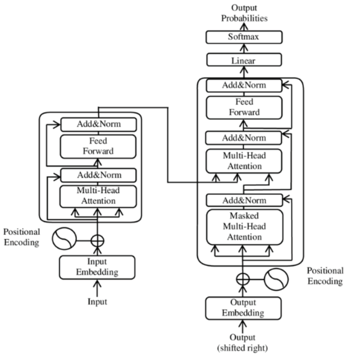


Fig. 1. Architecture of Transformer

979-8-3503-3860-7/23/$31.00 ©2023 IEEE

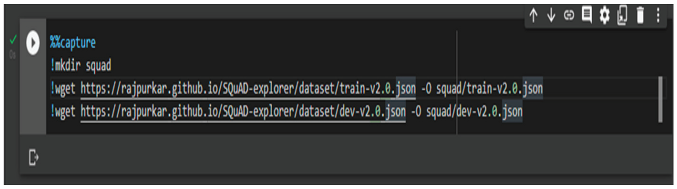
II. Related Works

A few papers are taken into consideration, [1] is the used models are BERT, RoBERTa, and ALBERT models, all of which are based on the Transformers franchise. The outcome is for multiclass sentiment analysis, BERT model is best when training duration and accuracy are high. Another paper [2], where the models used here are BERT, ALBERT, XLNet, RoBERTa , ConvBERT, BART.Given that Roberta is a dynamic masking method, it has been demonstrated in this study that it performs better than Bert.RoBERTa and BART are two of the models in this study that perform the best. [3]The models employed in this paper are the BERT model, Logistic Regression, Gaussian Mixture Model, Support Vector Machine, Random Forest, and XGBoost, and it is based on the Transformers concept. It has been established that ML models are ineffective (although data samples or epochs increased in most situations, we still saw over-fitting or subpar performance). The best is BERT.The paper [4] employs the idea of a transformer and the models BERT DISTILBERT to answer questions involving a health-related dataset.According to the COBERT system's ability to generate features, both models were satisfactory. [5]Dataset with few languages In order to get responses that are believable, AVAD-HAN compares the classifiers Support Vector Machine (SVM), Logistic Regression (LR), and Multi-Layer Perceptron (MLP). Following several studies, SVM gives the better results.

III. Comparative Study

A. Data Description

The Stanford Question Answering Dataset is the dataset under consideration (SQuAD). The dataset is known to contain 1,00,000 questions based on Wikipedia articles. It is known that the SQuAD dataset contains data in the form of .json files that comprise messages, queries, and replies. As a closed form dataset, this one always includes the answer to a question as well as a continuous range of context. There is a context (paragraph), a question, and an answer text for each observation in the training set. There are now two versions of SQuAD available; however, for the purposes of comparison, we choose SQuAD v2. The objective is to anticipate the answer text given any new context or inquiry. Finding the appropriate sentence in the passage, followed by finding the appropriate text within that sentence, will enable you to accomplish the aforementioned objective. We create the sentence embeddings using a technique developed by Facebook researchers called InferSent, which gives semantic representations for English sentences. We can obtain contextual embeddings with bigger weights to the most significant/ important words of the sentence after being pretrained on a larger corpus and after developing a preliminary vocabulary on SQuAD. The question-answering job now includes more challenging questions with context based on inferences thanks to SQuAD. Each question's solution is a passage of text, or span, taken from the reading article under consideration. In the second iteration of SQuAD, 50,000 unanswerable questions that resembled answers were introduced. We can determine which sentence most closely corresponds to the response by calculating the Exact Match and F1 score between sentences and questions and visualising the results in the multidimensional vector space. Additionally, we made an effort to employ Bert models to choose the most appropriate sentence embedding.



```
%%capture
!mkdir squad
!wget https://rajpurkar.github.io/SQuAD-explorer/dataset/train-v2.0.json -O squad/train-v2.0.json
!wget https://rajpurkar.github.io/SQuAD-explorer/dataset/dev-v2.0.json -O squad/dev-v2.0.json
```

Fig. 2. SQuAD dataset loading

B. Methodology

Firstly in case of BERT and ROBERTA approaches, we will load the data and read the data which is in the form of .json files. Dataset contains text, queries and answers, we will fetch the data from training and validation set. Then data pre-processing should takes place which includes tokenizing data, knowing start and end positions of question-answers as it is important in question answering task to know starting and ending positions of each question. It is important to find out starting and ending position ID's.

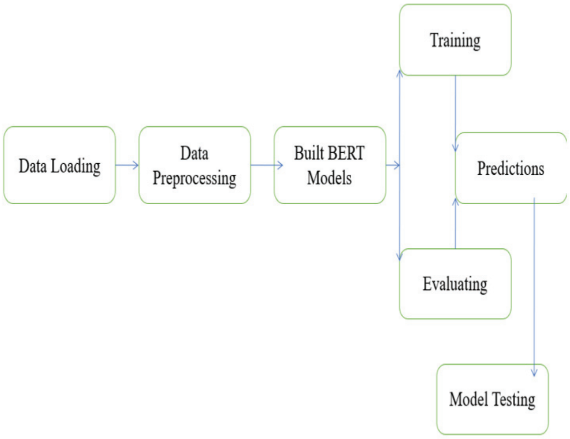
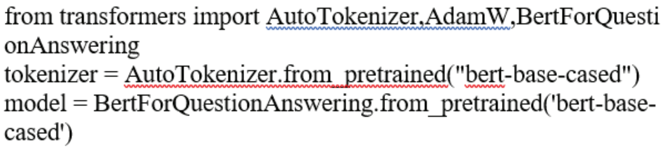


Fig. 3. Methodology

Then it is important to tokenize the passages and queries, and as we know BERT is known to work with tokens and hence SQuAD dataset is tokenized. Then it is important to create the SQuAD dataset class because it is used for easy handling and converting the data encodings into the datasets. For the data loaders, consider the batch size = 8.

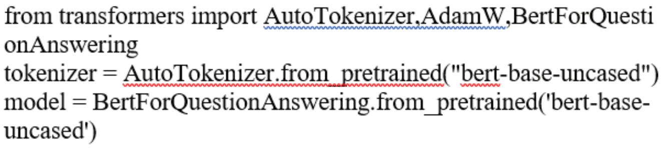
C. Creating Bert Models

Bidirectional Encoder Representations from Transformers, or BERT, is a deep learning model that is based on Transformers. In Transformers, each output element is connected to each input element, and the weightings between them are dynamically determined based upon their connection. In the past, language models could only interpret text input sequentially — either from right to left or from left to right — but not simultaneously. BERT is unique since it can simultaneously read in both directions. Bi-directionality is the term for this ability, which the invention of Transformers made possible. BERT is pre-trained on two distinct NLP tasks using this bidirectional capability: Discreet Language. On the SQuAD v2 dataset, the Question-Answering job has been carried out using pre-trained Bert-for-Question-Answering and Auto- Model-For-Question-Answering. AdamW is the optimizer that is employed, and it incorporates weight decay as well as gradient bias correction.



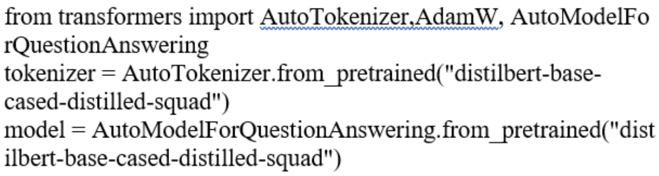
```
from transformers import AutoTokenizer,AdamW,BertForQuesti
onAnswering
tokenizer = AutoTokenizer.from_pretrained("bert-base-cased")
model = BertForQuestionAnswering.from_pretrained('bert-base-
cased')
```

Fig. 4. The syntax for Bert base cased



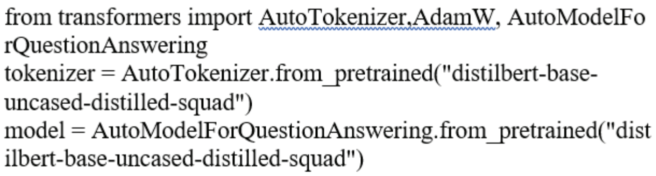
```
from transformers import AutoTokenizer,AdamW,BertForQuesti
onAnswering
tokenizer = AutoTokenizer.from_pretrained("bert-base-uncased")
model = BertForQuestionAnswering.from_pretrained('bert-base-
uncased')
```

Fig. 5. The syntax for Bert base uncased



```
from transformers import AutoTokenizer,AdamW, AutoModelFo
rQuestionAnswering
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-
cased-distilled-squad")
model = AutoModelForQuestionAnswering.from_pretrained("dist
ilbert-base-cased-distilled-squad")
```

Fig. 6. The syntax for Distil Bert base cased



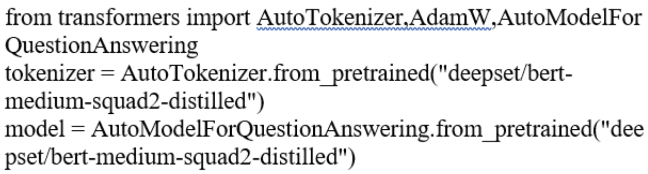
```
from transformers import AutoTokenizer,AdamW, AutoModelFo
rQuestionAnswering
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-
uncased-distilled-squad")
model = AutoModelForQuestionAnswering.from_pretrained("dist
ilbert-base-uncased-distilled-squad")
```

Fig. 7. The syntax for Distil Bert base uncased

D. Creating Roberta Model

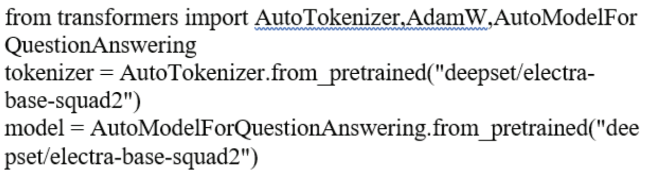
By creating the mask pattern corresponding to each input sequence that we feed RoBERTa without NSP loss, big minibatches, and a higher byte-level BPE, the model is trained via dynamic masking. It is an enhanced version of Google’s selfsupervised Bidirectional Encoder Representations from Transformers (BERT) approach for pre-training natural language processing (NLP) systems.

Since RoBERTa lacks token-type-ids, you are not required to specify which tokens correspond to which segments. Just use the separation token tokenizer to divide up your portions. ROBERTA (parameters L = 24, H = 1024, A = 16, and 355M)



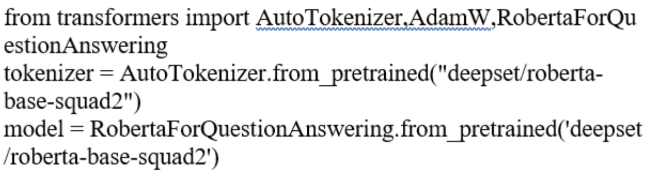
```
from transformers import AutoTokenizer,AdamW,AutoModelFor
QuestionAnswering
tokenizer = AutoTokenizer.from_pretrained("deepset/bert-
medium-squad2-distilled")
model = AutoModelForQuestionAnswering.from_pretrained("dee
pset/bert-medium-squad2-distilled")
```

Fig. 8. The syntax for deepset/bert-medium-squad2-distilled



```
from transformers import AutoTokenizer,AdamW,AutoModelFor
QuestionAnswering
tokenizer = AutoTokenizer.from_pretrained("deepset/electra-
base-squad2")
model = AutoModelForQuestionAnswering.from_pretrained("dee
pset/electra-base-squad2")
```

Fig. 9. The syntax for deepset/electra-base-squad2



```
from transformers import AutoTokenizer,AdamW,RobertaForQu
estionAnswering
tokenizer = AutoTokenizer.from_pretrained("deepset/roberta-
base-squad2")
model = RobertaForQuestionAnswering.from_pretrained('deepset
/roberta-base-squad2')
```

Fig. 10. The syntax for Roberta-base

E. After creating models, we perform training and evaluation of the models.

For the training purpose, I have considered 3 epoches for all the versions of BERT and ROBERTa models. I have trained the model with the parameters input ID’s, attention mask, start positions and end positions respectively. I have also calculated the loss values and back propagating them and applying optimizer on top of it which is ADAM. The same procedure is followed for the evaluation part. Therefore calculating the loss for each models used. The training model is hence saved as fine tune model and I have plotted graphs for Training Vs Validation loss. For prediction purpose I have mainly considered 2 parameters. Those are Exact Match(EM) and F1 Score

F. Exact Match

This statistic is as straightforward as it seems. If the characters of the model’s prediction perfectly match the characters of (one of) the True Response(s), for each question and answer pair, EM = 1, else EM = 0.

G. F1 Score

In QA, the F1 score is a popular indicator for classification issues. When precision and recall are equally important to us, it makes sense. In this instance, it compares each word in the prediction to each word in the correct response. The F1 score is based on the number of shared words between the prediction and the truth; recall is the ratio of shared words to the total number of words in the ground truth, and precision is the number of shared words to the total number of words in the prediction. The model is then tested with these prediction parameters and could able to achieve results for each model being considered in this paper. F1 score and Exact Match(EM).These are the parameters considered for the model testing.

IV. Results and Discussion

The following conclusions were reached when our dataset was subjected to a variety of BERT models and ROBERTA. When compared to other versions of BERT, it was discovered that ROBERTa produces the best results out of all of these models because its training and validation losses are lower. Distilbert-base-cased produces the best outcomes when BERT variants are taken into account. Here we have considered 3 epochs for few models and 2 epochs for other models and made a comparative study.

TABLE I
Comparative study of Bert versions and Roberta(First epoch)

Model	Training loss	Validation loss
Bert-base uncased	1.3338	1.1743
Distilbert-base-cased	1.0444	1.2978
Distilbert-base-uncased	1.1465	1.2387
Roberta-base	0.7487	0.7331

TABLE II
Comparative study of Bert versions and Roberta(second epoch)

Model	Training loss	Validation loss
Bert-base uncased	0.8848	1.1957
Distilbert-base-cased	0.8054	1.3330
Distilbert-base-uncased	0.83955	1.3163
Roberta-base	0.7718	1.0251

TABLE III
Comparative study of Bert versions and Roberta(third epoch)

Model	Training loss	Validation loss
Bert-base uncased	0.6905	1.3092
Distilbert-base-cased	0.6412	1.3518
Distilbert-base-uncased	0.6511	1.4162
Roberta-base	0.6517	1.0743

From the above three tables, we can clearly say that Roberta gives better results than any other Bert models. Among Bert models Dsitilbert-base-cased gives better results mainly in Training loss.

TABLE IV
Comparative study of other Bert versions(first epoch)

Model	Training loss	Validation loss
Bert-base cased	1.3683	1.2143
Bert-medium-squad2-distilled	1.0667	1.2082
Electra-base-squad2	0.7274	1.0618

TABLE V
Comparative study of other Bert versions(second epoch)

Model	Training loss	Validation loss
Bert-base cased	0.9610	1.2057
Bert-medium-squad2-distilled	0.8242	1.2316
Electra-base-squad2	0.5734	1.1531

Among these three version of bert, from the above two tables it is clear that Electra-base-squad2 models gives the better results.

V. Conclusion

As it was proven that, simple ML models could not perform well even when employing InferSent to generate contextual sentence representations (in most cases, we observed over-fitting or poor performance despite increasing data samples or epochs), in this paper we have considered Bert models and Roberta approaches to produce improved conclusions and outcomes regarding SQuAD. Roberta beats other Bert model iterations in terms of output since it uses a dynamic masking technique. One of the Bert models, distillbert-base-cased, produces the best results for three epochs. However, in terms of two epochs, Electra-base-squad2 produces the best results. The massive bidirectional transformer XLNet, which will be used in future research, employs better training techniques, larger data sets, and greater processing capacity to outperform BERT prediction metrics on a 20-language challenge. GPT-3 is 470 times larger than the BERT model in terms of size, having been trained with billions of additional parameters. We can later implement it by adding the responses' tags to the dataset, making the answer more precisely predictable. The graphs obtained are as follows:

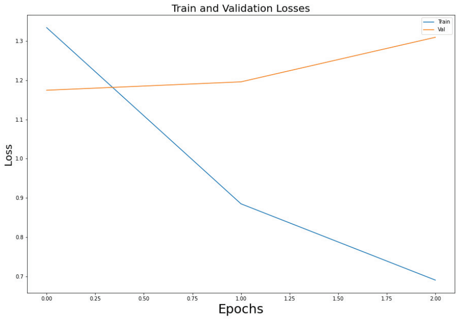


Fig. 11. The graph for Bert-base-uncased model

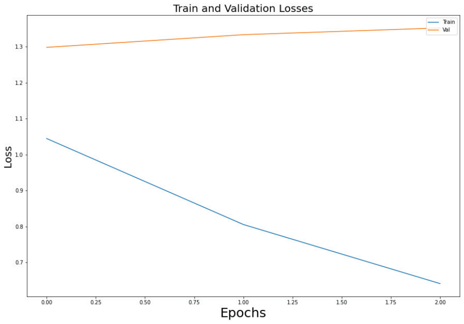


Fig. 12. The graph for Distilbert-base-cased model

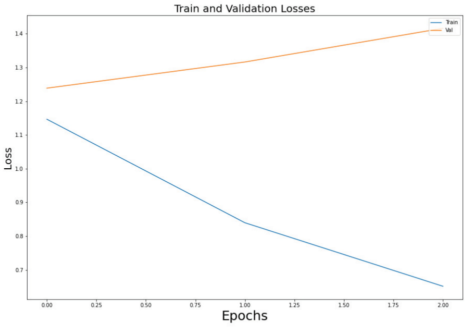


Fig. 13. The graph for Distilbert-base-uncased model

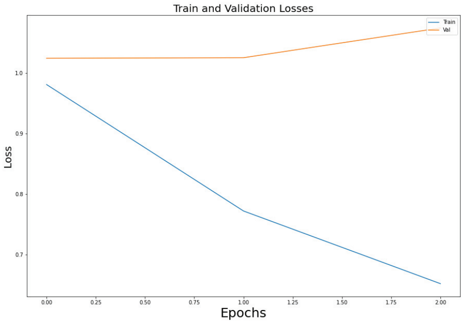


Fig. 14. The graph of Roberta-base model

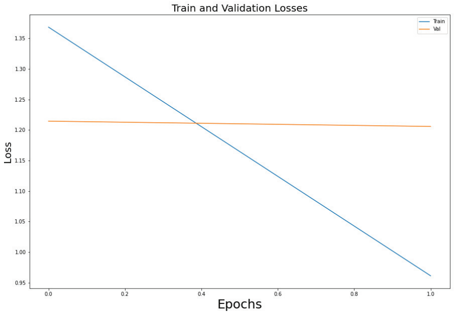


Fig. 15. The graph for Bert-base-cased model

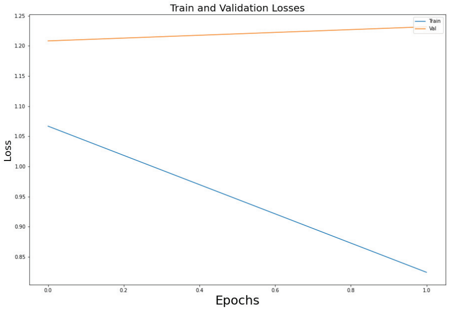


Fig. 16. The graph for Bert-medium-squad2-distilled model

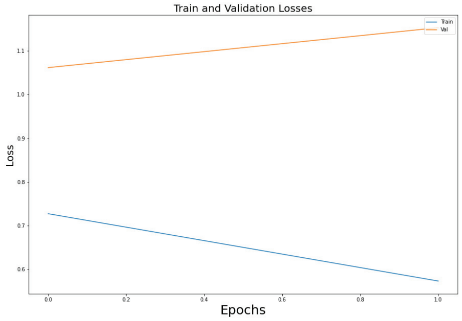


Fig. 17. The graph for Electra-base-squad2 model

REFERENCES

[1] Saurav Singla and N Ramachandra. Comparative analysis of transformer based pre-trained nlp models. *Int. J. Comput. Sci. Eng*, 8:40–44, 2020.

[2] Kate Pearce, Tiffany Zhan, Aneesh Komanduri, and Justin Zhan. A comparative study of transformer-based language models on extractive question answering. *arXiv preprint arXiv:2110.03142*, 2021.

[3] Devshree Patel, Param Raval, Ratnam Parikh, and Yesha Shastri. Comparative study of machine learning models and bert on squad. *arXiv preprint arXiv:2005.11313*, 2020.

[4] Jafar A Alzubi, Rachna Jain, Anubhav Singh, Pritee Parwekar, and Meenu Gupta. Cobert: Covid-19 question answering system using bert. *Arabian journal for science and engineering*, pages 1–11, 2021.

[5] Priyanka Ravva, Ashok Urlana, and Manish Shrivastava. Avadhan: System for open-domain telugu question answering. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 234–238. 2020.