International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 05, May 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Machine Reading Comprehension

Akanksha Pal
Computer Science and Engineering
Indira Gandhi Delhi Technical University for Women, Delhi, India
akanksha096btcse18@igdtuw.ac.in

Bhavika Rastogi
Computer Science and Engineering
Indira Gandhi Delhi Technical University for Women Delhi, India
bhavika133btcse18@igdtuw.ac.in

Aashi Verma
Computer Science and Engineering
Indira Gandhi Delhi Technical University for Women, Delhi, India
aashi102btcse18@igdtuw.ac.in

Sanchi Bisht
Computer Science and Engineering
Indira Gandhi Delhi Technical University for Women,Delhi, India
sanchi021btcse18@igdtuw.ac.in

Abstract

Machine-Reading-Comprehension is a widely used yet demanding task in the Natural language processing domain. Here the mission is to analyse and form a context to a given context and then answer queries based on the comprehension. Question Answering (QA) System is a powerful technique as most of the problems based on deep learning can be considered question answering problems. Natural language processing is one of the most popular and researched fields in Computer Science nowadays. This study exploits deep learning models and transformers like BERT (a State-of-the-Art model) and datasets like Stanford Question Answering Dataset (SQUAD) to train the QA system to understand textual data and get answers to questions that are unanswerable as well as answerable. The study of various models and word embeddings has been made to determine the accuracy and F1 scores with different models. So far, our best-proposed single model built using BERT base uncased model has achieved an F1 score of ***78.01*** *and BERT large uncased model has achieved an F1 score of* ***80.01****.*

Keywords: Natural Language Processing, BERT, SQUAD, BERT-Large, BERT-Base, MLM, NSP

1.INTRODUCTION

NLP (Natural Language Processing) has become an increasingly important component of deep learning, which has a huge potential of applying conversational machines to solve problems across a diverse range of industries. Sentiment analysis, named entity recognition (NER), machine translation, etc., are among the applications of NLP.

Reading comprehension by machines is a cutting-edge technology of Natural Language Processing (NLP). In this, we teach machines to understand and obtain correct answers to the questions based on the given passages. It is the responsibility of machine reading comprehension not only to answer questions but also to avoid unanswerable questions tactfully. In the same way, a person reads a document, machine reading comprehension scans documents and extracts meaning from them. As you ask machine reading comprehension questions about a document, machine reading comprehension will use different parts of the material until an answer is formulated.

Stanford NLP collected more than 100,000 questions and answers on Wikipedia for their Question Answering Dataset in 2016. To automate the process of answering questions based on a contextual document, we had to train a model. The model would return part of contextual document most likely to answer a question when given a contextual document (free form text). Top AI practitioners around the world tackled this problem but there was no such model that beat the human benchmark. The brilliant minds at Google Brain introduced BERT (Bidirectional Encoder Representations from Transformers) a language understanding model. This model surpasses the human benchmark when compared against the SQuAD test set through some fine-tuning.

As stated above, Question Answering from comprehension is a challenging task of natural language processing and various studies have also been done in this field using various embedding models like Global Vectors, Embedding from Natural Language models, and Bidirectional Encoder Representation for Transformers (BERT). But among all of them, BERT proved to be the most efficient one, thus it is the state-of-the-art model solution.

Also, studies in the past were done on the squad 1.1 dataset which could not handle the unanswerable questions from the paragraph because it contains only question answers pairs as the dataset. Thus there comes a newer dataset squad version 2.0 which combines squad version 1.1 and unanswerable questions in it. The concept of giving answers to unanswerable questions as well-motivated us to work on the research towards it, allowing us to build models that answer both unanswerable as well as answerable questions as demanded by the systems.

The area of research can also be extended to build various applications that can be designed to generate questions and answers to the paragraphs after uploading the pdf. Not only this but it can also help the student and teachers to generate questions and answers to prepare for various competitive examinations which will also be our future area of study.

2. DATA

There are two versions of SQUAD: SQUAD version 1.1 and SQUAD version 2.0. Crowdsourced questions about Wikipedia articles make up this comprehension of reading dataset. Furthermore, every question has a response that is either a fragment of text from the reading passage or the question is unanswerable.We have used the official SQuAD 2.0 dataset train and dev sets for the scope of our research. The dataset can be downloaded from Rajpurkar's github page. SQUAD dataset was created by choosing around 500+ articles from Wikipedia. They retrieved a total of 2300 plus unique paragraphs from each of the selected articles (making sure to filter for paragraphs that were too small). The dataset is divided into three sets based on no. of articles: 80% in the training set, 10% in the development set, and 10% in the testing set.

International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 05, May 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Table 1

	Train	Development	Test
Total examples	130319	11873	8862
Negative examples*	43498	5945	4432
Total articles	442	35	28
Articles with negatives*	0	35	28
Range of number of context tokens	[23, 408]	[27, 448]	-
Mean number of context tokens	116	122	-
Percentage of examples with number of tokens > 300	0.9%	3.5%	-
Range of number of question tokens	[4, 28]	[4, 17]	-
Mean number of question tokens	10	10	-

Note: "Negative examples" stands for the number of questions that are unanswerable while "Articles with negatives" refers to the number of articles which all questions are unanswerable.

SQUAD version 1.0 is the previous version of the SQUAD dataset, which contains more than 100000 question-answer pairs on hundreds of articles. SQuAD version 2.0 combines the more than 100000 questions from SQUAD version 1.1 with over 50000 unanswerable questions. The system should not only respond to the possible question but also handles cases when no answer is supported by the paragraph and refrain from responding. The are there three categories of datasets : Train set : It contains 130319 data points, all of these examples are from the official SQUAD version 2.0 training dataset set. Dev set: It contains around 11873 data points randomly selected half of the development collection. Test set : It contains around 8862 data points the rest of the official dev set.

EXAMPLE:

Context:

The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. They were descended from Norse (\"Norman\" comes from \"Norseman\") raiders and pirates from Denmark, Iceland and Norway who, under their leader Rollo, agreed to swear fealty to King Charles III of West Francia. Through generations of assimilation and mixing with the native Frankish and Roman-Gaulish populations, their descendants would gradually merge with the Carolingian-based cultures of West Francia. The distinct cultural and ethnic identity of the Normans emerged initially in the first half of the 10th century, and it continued to evolve over the succeeding centuries.

- **Answerable:**

Question: In what country is Normandy located?

Answer: France

- **Unanswerable:**

Question: Who did King Charles III swear fealty to?

Answer: nil

3. METHODOLOGY

Word embedding model and fine-tuning BERT. Word embeddings convert each word of text into vectors which can then be taken as an input to our Machine Learning model. Our predictive model, which is created by fine-tuning BERT, i.e adding an additional output layer to BERT and changing some of the hyperparameters, can then be used to generate the predicted output. From this predicted output, loss is calculated and then back propagated to update weights. Our goal is to minimize the loss and hence generate better accuracy for predictions. The vectors that are sent into the neural network, which gives us the prediction of where the context paragraph will begin and end. The projected response to the question is the span of text limited by the two spots. The model will conclude that the query has no response if the network's estimated starting and ending positions do not exceed any no-answer threshold.

3.1 BERT

BERT uses a mechanism known as Transformer to learn associated relationships between words in a text. The Transformer has two algorithms: an encoder for reading input text and a decoder to produce predictions. Since the purpose of BERT is to build a language model only an encoder is needed. Consequently, the model is considered bidirectional, since it can contextualize a word using all the information available based on all surrounding factors. SQuAD 2.0 is an MRC dataset that receives a comprehension as input and the answer is a span from the input passage (Rajpurkar et al., 2018). In our research the SQUAD 2.0 dataset uses the following existing settings of BERT model: BERT base and BERT large.

BERT is based on encoder layers that are stacked one on top of the other. The number of encoder layers is the difference between BERT base and BERT Large. The BERT base model has 12 encoder layers layered on top of one another, but the BERT large model has 24 encoder layers placed on top of one another.

The number of parameters (weights) and attention heads rise as the number of levels in the BERT large increases. The BERT base has 110 million parameters and 12 attention heads (which allow each token in the input to focus on other tokens). BERT large, on the other hand, contains 16 attention heads and 340 million parameters. The BERT base contains 768 hidden layers, whereas the BERT large has 1024.

International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 05, May 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

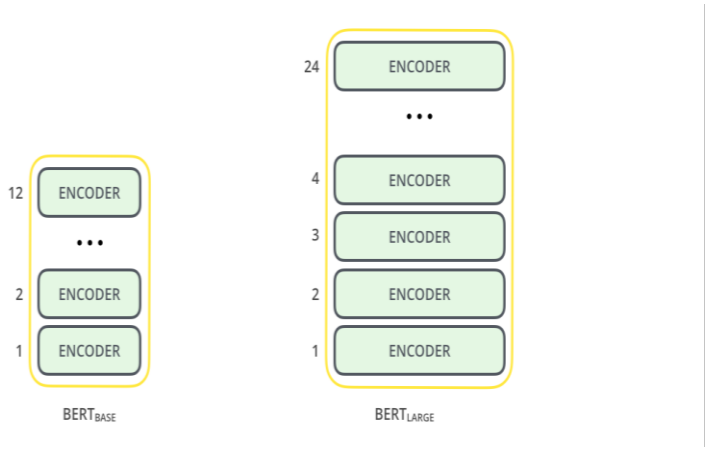

Figure 1: BERT base vs Bert Large

3.1.1 BERT Pre-training

As part of the BERT model training, Masked Language Modelling and Next Sentence Prediction are employed together.

Masked LM (MLM)

In masked LM around 15% of the words are substituted with mask token before being sent into the BERT. The model predicts the real value of the words which are masked based on the situations present by the other non-masked words.

Next Sentence Prediction (NSP)

The approach followed while training bert involves : i) sending pairs of sentences to model and ii) getting efficient in predicting whether the second sentence in the text will come next in the original document. Figure 2 shows three embeddings that are utilized by the model to help distinguish sentences during training. For each token, the representation consists of a token, segment, and position embedding.

1. The sentences start with: [Cls] and end with : [Sep].
2. To each token, there is an addition of embedding representing sentence (a) or sentence (b).
3. Further,in the positional embedding, the token contains a message labeling its place in the sequence.

International Journal of Advance Research in Science and Engineering
Volume No. 11, Issue No. 05, May 2022
www.ijarse.com
IJARSE
ISSN 2319 - 8354

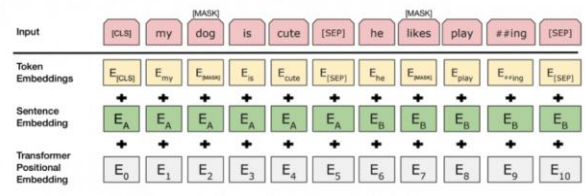

Figure 2: Word Embedding

3.1.2 BERT Fine-tuning

For our study we have used the BERT Fine-Tuning predictive model. This predictive model concatenates the context paragraph with the question provided by the user. Then the attention between the comprehension and the question is calculated. Feed forward neural network is used to assign weights to the tokens. On the basis of loss function the required weights are updated in each iteration. The inputs are transferred into a neural network or deep learning based predictive model to estimate the answer's start positions and end positions in the comprehension provided. The question can be judged as an unanswerable answer if the probabilities of prediction are not exceeding a fixed threshold defined.

We managed to run our model for some hyperparameters that we believed could perform the best. We consulted the lectures of BERT for the values we want to try. Some of the values of hyperparameters were: batch size as either 8 or 16, for learning rate the values were 5e−5 (which we used to train) , 2e−5, 3e−5 and the number of epochs were taken as 2, 3 and 4.

We kept the values of batch size as 8, number of epochs as 3 and learning rate as 5e-5 for optimal hyperparameters. Concerning the *number of epochs* we began with 2 and we realize that overfitting happened from the very first epoch. So we understand that there isn't a reason to try much more epochs. When we tried to train my model for 4 epochs, the connection and cuda never let me, so we trained my model with only 3 epochs. Moreover, it doesn't matter so much, as overfitting happened in the first epoch and the gap between the training and validation loss was getting bigger and bigger as the epochs increased. As for *learning rate*, We tried all of these three values, and nothing changed for overfitting, so we kept the value of 5e-5, as its performance was a little better (overfitting started from a later epoch). We selected the fine tuning technique that trains the entire architecture. The pre-trained weights of the model can be updated with the loss derivative which means the back propagated error. This is done for the new dataset. Furthermore, QA tasks are very complicated and it is important to have the information of all layers.

4. EXPERIMENTS

4.1 Evaluation Method

We employ the usual SQuAD performance measures of Exact Match (EM) and F1 to evaluate our models. With respect to the development set, our main focus is on the Exact Match score and F-1 scores for our project.

Exact Match is a binary metric used for determining if the output of any system is exactly matching the true answer or not.

F1 is a less strict metric, and it is the harmonic mean of precision and recall. F1 = 2×precision×recall precision + recall . Here by precision we mean ratio of true positives to the sum of true positives and false positives. Recall means the ratio of true positives to the sum of true positives and false negatives.

When the system's answer is a subset of the ground truth answer, it will have 100 percent precision and 50 percent recall if just one of the two words in the ground truth output is used. When there is no answer to a question, the F1 and EM scores are 1 if the model predicts no answer and 0 if it does not.

4.2 Experimental Details

The default BERT-base model setups were used in various investigations. All of our suggested models had the same training hyperparameters: training batch size=8, learning rate=5 × 10-5, number of training epochs=3.0, maximum sequence length=384, doc stride=128. Every experiment was run on a server with two GPUs. A model's training time was generally between 8 and 12 hours.

4.3 Results

We kept the values of batch size as 8, number of epochs as 3 and learning rate as 5e-5 for optimal hyperparameters. We found out that among the two models, model BERT-large uncased performs significantly better than Bert-base uncased. Our best-proposed single model built using BERT base model has achieved an F1 score of 78.01 and BERT large model has achieved an F1 score of 80.01.

5. CONCLUSION

In this study, we experiment with a variety of hyperparameters, such as learning rate, dropout, and training epochs, to see how powerful the basic BERT model is. By fine-tuning the parameters in each layer, we may improve the performance of our model even further. In-depth comparison between BERT-Base and BERT-Large was made. We can also notice that there is still a disparity in has-answer and no-answer accuracy based on the error analysis.

In terms of training data, hyperparameter tuning, and model design, there are still areas where the QA model can be improved in the future. The long-term goal of this research effort should be to demonstrate the ability to answer questions in big articles or papers rather than brief paragraphs (as in SQuAD).

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[2] Weihao Yu, Zihang Jiang, Yanfei Dong & Jiashi Feng. RECLOR: A READING COMPREHENSION DATASET REQUIRING LOGICAL REASONING. arXiv:2002.04326v3 [cs.CL] 22 Aug 2020

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL] 26 Jul 2019

[4] Zhuosheng Zhang, Junjie Yang, Hai Zhao,Retrospective Reader for Machine Reading Comprehension. arXiv:2001.09694 [cs.CL] 11 Dec 2020

[5] Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108v4 [cs.CL] 1 Mar 2020

[6] Sewon Min, Victor Zhong, Richard Socher, Caiming Xiong. Efficient and Robust Question Answering from Minimal Context over Documents. arXiv:1805.08092v1 [cs.CL] 21 May 2018

[7]Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin,Attention Is All You Need,arXiv:1706.03762[cs.CL]6 Dec 2017

[8]Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, Hannaneh Hajishirzi,Bidirectional Attention Flow for Machine Comprehension.arXiv:1706.03762[cs.CL]21 Jun 2018

[9]Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016

[10] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822, 2018.

[11] Sam Cheung Hiu Fan. Deep learning in NLP - Constructing a Machine Question Answering Model.

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pre training for language understanding. arXiv preprint arXiv:1906.08237, 2019.