BERT-BASED QUESTION ANSWERING TASK

Wu Hao
MSc in Big Data Technology
Hong Kong University of Science and Technology
20711289
hwubx@connect.ust.hk

Chan Chunkit
MSc in Big Data Technology
Hong Kong University of Science and Technology
20728737
ckchancc@connect.ust.hk

Yang Siting
MSc in Big Data Technology
Hong Kong University of Science and Technology
20714786
syangcd@connect.ust.hk

ABSTRACT

In this report, we investigate the performance of 3 different BERT related models for question answering task on the RACE dataset. Those models include BERT, Roberta, and ALBERT. Furthermore, we implement different experiments which include fine-tuned hyperparameter on the Albert-base-v2 and implementing the Easy Data Augmentation on the RACE dataset. Finally, our Albert-xxlarge-v2 model obtained 67.69% test accuracy on the RACE dataset.

1 INTRODUCTION

Question answering (QA) is a task within the fields of information retrieval and natural language processing (NLP), which answering the questions posed by humans (normally reading comprehension question). In this paper, we will focus on the automated multiple-choice reading comprehension on RACE dataset. We train different BERT related models and select best one which obtain better performance among those models. These models which include BERT, Roberta, and ALBERT.

In this paper, we firstly review the background of Question answering task, dataset and the models. Moreover, we will demonstrate how we preprocess the data and perform several experiments to train different models. Furthermore, we also display how to use Easy Data Augmentation and fine-tuning different hyperparameter to obtain higher performance.

2 LITERATURE REVIEW

2.1 QUESTION ANSWERING (QA)

Question Answering (QA) is a multidisciplinary task that includes natural language processing, information technology, artificial intelligence, and cognitive science. The QA system attempts to understand the question pose in natural language and then find out the correct answer given a set of documents or passages (Gupta & Gupta, 2012). The question answering system includes three subtasks which are question classification, information retrieval, and answer extraction. Question classification is to classify the question based on the type of entities or other information of question while the information retrieval is extracting and identifying the candidate answer by using the intelligent question answering system. After completing these two subtasks, the answer extraction is ranking and validating those candidate answers.

In general, the question answering system attempts to generate a concise, comprehensible, and correct answer by referring to the word, sentence, paragraph, audio fragment, image, or even an entire

document (Kolomiyets & Moens, 2011). The main goal of the Question-Answer System is to find out "WHO did WHAT to WHOM, WHERE, WHEN, HOW, and WHY?" (Guda et al., 2011).

2.2 DATASET

The RACE dataset is a large-scale reading comprehension dataset for benchmark evaluation of methods in the reading comprehension task. This dataset was collected from English examinations for middle school and high school Chinese students aged range between 12 to 18. Compared with other benchmark datasets for reading comprehension, the proportion of question required reasoning in the RACE dataset is much larger. Therefore, there is a wide gap between some state-of-art models (43%) and the ceiling human performance (95%) (Lai et al., 2017).

2.2.1 BERT

BERT is a language representation model introduced by Google AI (Devlin et al., 2018), standing of Bidirectional Encoder Representations from Transformers. This model is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers. Inspired by the Cloze task (Taylor, 1953), it used a "masked language model" and demonstrate the importance of bidirectional pre-training for language representations. BERT is also the first language representation model which is based on fine-tuning and achieves state-of-the-art performance.

2.2.2 ROBERTA

Standing for A Robustly Optimized BERT Pretraining Approach, RoBERTa was proposed by Facebook AI in 2019 (Liu et al., 2019) to measure the impact of many key hyper-parameters and training data size. The modifications are simply, including: (1) training the model with bigger batches over more data; (2) removing the Next Sentence Prediction loss and using longer sequences while training; (3) dynamically generating the [MASK] pattern applied to the training data.

2.2.3 ALBERT

ALBERT (A Lite BERT) is a lite vision of BERT , introduced by Google Research in 2020 (Lan et al., 2019). It solves the memory consumption problems and long training time of BERT by introducing a factorized embedding parameterization and cross-layer parameter sharing, which significantly reduce the number of parameters. Moreover, ALBERT also introduces a self-supervised loss for Sentence-Order Prediction (SOP) addressing the ineffectiveness of the Next Sentence Prediction loss in the original BERT (You et al., 2019) .

3 METHODOLOGY

3.1 DATASET

The RACE dataset includes more than 28,000 passages and nearly 100,000 questions to covers a variety of topics designed to evaluate the ability in understanding and reasoning of students. In this benchmark dataset, different models can be evaluated based on the accuracy obtained on the total dataset (RACE), middle school examinations (RACE-m), and high school examinations (RACE-h).

Table 1: RACE Dataset

Dataset	**RACE-Middle**			**RACE-High**		
Subset	Train	Dev	Test	Train	Dev	Test
Passages	6,409	368	362	18,728	1,021	1,045
Questions	25,421	1,436	1,436	62,445	3,451	3,498

Moreover, compared with other benchmark datasets such as SQuAD (Rajpurkar et al., 2016), the RACE required more reasoning to find out the answer and cannot directly extract the answer from the passage. There are several challenges to find out the correct answer:

* Involve a variety of question types such as Context Matching, Deduction, Inference, and Summarization.
* Cover a wide range of various domains and writing style.
* Required high level of reasoning and calculation techniques.

Passage:
"Tomorrow is Saturday. I'm not going to work, and my brother isn't going to school. We are going to play table tennis. We are going to have lunch in a restaurant. We're coming home at five. My parents are going to visit my grandparents. They are going to get home at half past five. We are going to help my mother cook the dinner. After supper, I am going to dance with my friends, and my brother is going to watch TV with my parents."
Questions:
1): What day is it today?
A. Saturday B. Sunday C. Friday D. Monday
2): My brother is a_.
A. student B. teacher C. worker D. manager
3): My brother and I are going to _ tomorrow.
A. have a picnic B. visit our friends C. go fishing D. play table tennis
4): My parents are going to come back_.
A. at 7:30 B. at 5:30 C. at 6:30 D. at 7:00
5): I'm going to dance_.
A. after breakfast B. with my friends C. in the morning D. with my patents
Answers: ["C", "A", "D", "B", "B"]
Id: "middle24.txt"

Figure 1: Sample reading comprehension problems in RACE dataset.

3.2 Dataset Processing

In this project, in order to preprocess the data for BERT-based deep learning models, we have the following several steps:

* Tokenize the text
* Add special tokens - [CLS] and [SEP]
* Pad the sentences to a designed length

After tokenizing the text, we construct the input sequence for the BERT model by concatenating a passage, a question, and an option together with [CLS] and [SEP]. Since each question has 4 options, there is 4 input sequence for each question. Input:

[CLS] passage [SEP] question option 1 [SEP]

[CLS] passage [SEP] question option 2 [SEP]

[CLS] passage [SEP] question option 3 [SEP]

[CLS] passage [SEP] question option 4 [SEP]

Output: the predicted label of an option

Token embedding:[3,242,543,645,767,213,532,33]

segment embedding: [0,0,0,0,1,1,1,1]

Attention mask:[1,1,1,1,1,1,1,0,0,0,0]

3.3 DATA AUGMENTATION

The Data Augmentation will be used in this project and try to improve the performance of ALBERT-base model. We adopt some operations of Easy Data Augmentation (Wei & Zou, 2019) in this project to construct the augmented passages and hope that those operations can help ALBERT-base model to prevent overfitting during the training process and will build a robust model.

Table 2: Sentence sample for 4 operations Easy Data Augmentation

method	**sentence**
original sentence	This Project will focus on the data augmentation techniques in NLP
Word Swap	This Project will focus on the augmentation data techniques in NLP
Synonym Replacement	This Project will focus on the augmentation data method in NLP
Word Insertion	This Project will focus on write-up the augmentation data techniques in NLP
Word Deletion	This Project will focus on the augmentation data in NLP

There are 4 operations of Easy Data Augmentation in this project. The first operation is word swap which randomly selects two words of the same sentence to swap their positions. The next operation is synonym replacement that we randomly select some words to be replaced with their synonyms. The third operation is word insertion which we insert a random synonym of a random non-stop word from the sentence, this synonym will be inserted into random position of this sentence. The last operation is word deletion which we randomly remove words in a sentence and the operation associated with a lower probability.

3.4 MODELS

3.4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a language representation model that is designed to pre-train deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context in all layers. Its architecture is a multi-layer bidirectional Transformer encoder with the job of mapping a token sequence into a sequence of vectors.

For a given sequence, its input representation is constructed by summing the corresponding token, segment, and position embeddings (shown in Figure 1). It used WordPiece embeddings as token embedding with each embedding standing for the corresponding token. Additionally, the first token of every sequence is always a special classification token ([CLS]). For sentence pairs differentiating, it first separates them with a special token ([SEP]) and then indicates whether it belongs to sentence A or B by adding segment embeddings with '0' stands for sentence A and '1' stands for sentence B. And the position embeddings are used to inject information about token positions in order to make use of the order of sequence.

There are two steps in the framework: pre-training and fine-tuning. In pre-training, BERT is trained on unlabeled data using two unsupervised tasks Masked Language Model (MLM) and Next Sentence Prediction (NSP). In the MLM task, some input tokens will be randomly masked to train a deep bidirectional representation and predict some masked tokens. And the only 15% of the token

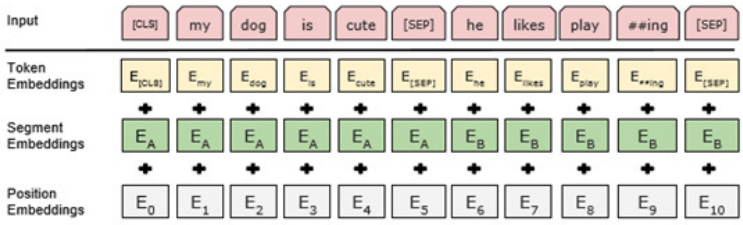

Figure 2: Input of Bert.

positions will be randomly chosen for prediction. In the NSP task, when choosing sentence pairs, 50% of the time B is the actual next sentence that follows A with labeled as [IsNext], and 50% of the time it is just a random sentence labeled as [NotNext]. With that, a binarized next sentence prediction task is pre-trained to understand the relationship between two sentences.

In fine-tuning, BERT is first initialized with the pre-trained parameters which will be all fine-tuned using labeled data from the downstream tasks. By swapping out the appropriate inputs and outputs, BERT uses the self-attention to unify single text and text pairs involved as encoding a concatenated text pair includes bidirectional cross attention between two sentences.

3.4.2 ROBERTA

Standing for A Robustly Optimized BERT Pretraining Approach, RoBERTa was proposed by Facebook AI in 2019[5] to measure the impact of many key hyper-parameters and training data size. The experiments have proved that bigger batches size can bring to better performances so that RoBERTa increases the size of mini-batches over more data, which directly results in longer training time. Secondly, RoBERTa removes the Next Sentence Prediction loss and trains on longer sequences after comparing the performances of segment + NSP in BERT style, sentence pair + NSP, full-sentences and doc-sentence from which it was found out the input format doc-sentences without NSP loss performed the best. Moreover, RoBERTa changes the masking pattern applied to the training data from static in the original BERT to dynamic, which means that the [MASK] pattern will be dynamically generated every time when the training data feed the model, instead of generating in the data pre-processing period once for each sample.

Additionally, RoBERTa changes the way of text encoding from char-level to bytes-level, solving the [UNKNOWN] pattern generating problems when encoding the input data.

3.4.3 ALBERT

As increasing model size makes pre-training harder due to the limitations of GPU/TPU memory and longer training times. ALBERT (A Lite BERT) was introduced by Google Research[3] to lower memory consumption and increase the speed of training while without seriously hurting performance with two-parameter reduction techniques used.

The first one is factorizing embedding parameterization. For the original BERT, the size of Word-Piece embedding equals to the size of the hidden layers so that increasing the size of hidden layers will directly increase the size of the embedding matrix, which may be further enlarged by vocabulary size. ALBERT separates the size of hidden layers from the size of vocabulary embedding by decomposing a large vocabulary embedding matrix into two small matrices, which makes it easier to increase the size of hidden layers without significantly increasing the parameter size of vocabulary embeddings. And this approach will significantly reduce the number of parameters when the hidden size is much larger than the size of vocabulary embeddings. The second is cross-layer parameter sharing, which avoids the size of parameters increasing with the depth of the network.

Furthermore, to improve the performance, ALBERT introduces a self-supervised loss for Sentence-Order Prediction (SOP), focusing on inter-sentence coherence to address the ineffectiveness of the NSP loss in the original BERT.

With all the above changes based on BERT, ALBERT can be scaled up to much larger configurations that still have fewer parameters than BERT-large but achieve significantly better performance.

4 EXPERIMENT

4.1 MODEL PERFORMANCE

We applied 3 different models with different parameter size to the data set. The first one was the application of the BERT-large model which has a bigger parameter size than base BERT with more hidden layers, more units as well as more attentions. Spending up to almost 4 times of training time than BERT-base, we got a higher accuracy of 56.12% using BERT-large. We applied 3 different models with different parameter size to the data set. The first one was the application of the BERT-large model which has a bigger parameter size than base BERT with more hidden layers, more units as well as more attentions. Spending up to almost 4 times of training time than BERT-base, we got a higher accuracy of 56.12% using BERT-large.

Table 3: Accuracy and Execution time

Model	**Accuracy**	**Parameter**	**Execution time(minute)**
Bert-base	54.10	108M	90
Bert-large	56.12	334M	352
Roberta-base	56.76	125M	138
Roberta-large	61.23	335M	492
Albert-base-v2	58.62	12M	306
Albert-xxlarge-v2	67.69	235M	2056

Then we turned to the other two improved versions of BERT. The first one is the RoBERTa. With some robust improvement on the batch size, sequence length and masking way as well as encoding method[5], RoBERTa-base got a better performance than both BERT-base and BERT-large, and can be further improved with more parameters in RoBERTa-large.

The other one is ALBERT. With a much smaller parameter size resulting from a factorized embedding parameterization and cross-layer parameter sharing[3], ALBERT got the best accuracy but the longest training time.

4.2 HYPERPARAMETER TUNING (BASED ALBERT-BASE)

In the project, we experimented the following hyper-parameters on the ALBERT-base model.

4.2.1 LEARNING RATE

Based on the ALBERT-base model, we implemented 3 learning rates, 1e-5, 1e-4 and 5e-6. The results are shown in Table 4. we can get the best results when the learning rate is 1e-5. As the Figure 3 shown, When the learning rate is set to 1e-4 and 5e-6, the model does not converge at all.

Table 4: Learning rate

Model	**Accuracy**	**Learning rate**
Albert-base-v2	58.62	1e-5
Albert-base-v2	21.52	1e-4
Albert-base-v2	22.32	5e-6

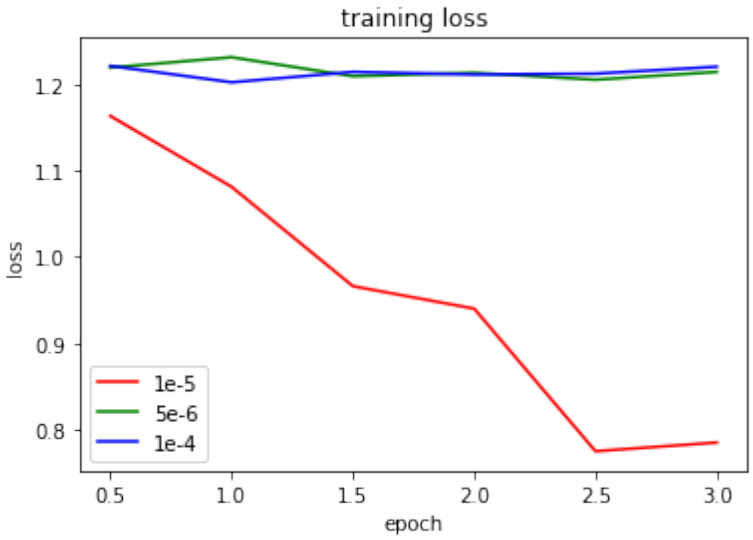

Figure 3: Training loss in different learning rate

4.2.2 L2 Regularization

We also tried an L2 Regularization with different coefficients of 0.1, 0.01 and 0.001. And the corresponding accuracy turns out to be 58.62%, 58.65% and 58.63%. Hence, we got the best performance when the L2 coefficient is 0.01.

4.2.3 Max Sequence Length

In this project, max sequence length is another important hyper-parameter to affect the performance of BERT models. Hence, we design the experiment to test different max sequence length. In this experiment, the max sequence length includes 40, 60, 80. The best model is the Albert-base-v2 with 80 max sequence length and obtains 58.62% accuracy. Another interesting finding is that with the increase in the max sequence length, the performance also increases based on the experiment result of this project.

Table 5: Max sequence length

Model	**Max seq len**	**Accuracy**
Albert-base-v2	80	58.62
Albert-base-v2	60	48.72
Albert-base-v2	40	45.47

4.3 Data Augmentation (Based ALBERT-BASE)

After performing the Easy Data Augmentation with 10% of the overall training set and the parameter α set to be 0.2, the accuracy of our Albert-base increase from 58.62% to 59.26%. In this experiment, we will choose 10% of the overall training set to implement the data augmentation to the randomly selected passages and their corresponding questions and options. The percentage of words changed in each passage by using each data augmentation operation will be indicated by the parameter α. For instance, when α set to be 0.2, by implementing the Easy Data Augmentation, all operations which include word swap, synonym replacement, word insertion and word deletion will be implemented with 20% of the total number of words in each passage (included their corresponding questions and options).

Table 6: Data Augmentation

Model	Data Augmentation Parameter(α)	Accuracy
Albert-base-v2	0.1	58.64
Albert-base-v2	0.2	59.26
Albert-base-v2	0.5	58.26

In this project, we implement the experiments in which the parameters α set to be 0.1,0.2 and 0.5, and those parameters result in 58.64%, 59.26% and 58.26% respective respectively in ALBERT models. Based on their performance, we discover that this parameter set to be 0.2 obtained the best performance.

5 ANALYSIS

5.1 MODELS

Table 3 has shown the comparison between three different types of models in different size. Albert-large performs best in these models. Comparing with the three models: Bert,Roberta,Albert-large, and we also found that the parameters of it are smaller than the bert-large and roberta-large. Considering the structure of the model, Albert change the task of Next Sentence Prediction(NSP) in Bert to the Sentence-Order Prediction (SOP). It is obviously that this task can let model learn the full text information better. As for the results of the model, Albert modified only one task of the pre-training model, which significantly improved the model's performance on the reading comprehension task.

However, the factorizing embedding parameterization and cross-layer parameter sharing in the albert did not perform as well as it had hoped. These two tasks aim to reduce the space consumption and the time consumption of the albert. But based on the results of the experiment. Although it theoretically reduces the number of arguments, it takes more time to train and iterate, and the space taken up by the model is not significantly reduced.

In conclusion, Roberta increased the size of the model, so it got better results than Bert. And the Albert performs best because of the SOP task. For the same model, a model with a larger number of parameters is trained more slowly, but the results will be better.

5.2 MAX SEQUENCE LENGTH

Due to the lack of hardware resources and the long article, it is difficult to input the entire article into the model for processing. So we had to do something with the text, and depending on how Google handled it, we chose to use truncation. We set a max sequence length before our preprocessing. Then we remove all the words that exceed the max sequence length and keep only the words within it.

According to our analysis of the experimental results, this kind of truncation will affect the performance of the model. That's because it is equivalent to the reduction of some information in the data. So it will have a certain negative impact on the model. Obviously, this truncation method is not conducive to model fitting. And the shorter the max sequence length is, the more negative impact it will have on the model.So we should either make the max sequence length as large as possible, or use some other way to deal with long text.

5.3 DATA AUGMENTATION

After performing the Easy Data Augmentation with 10% of overall training set and the parameter α set to be 0.2, the accuracy of our Albert-base-v2 increase from 58.62% to 59.26%. In this experiment, we will choose 10% of overall training set to implement the data augmentation to the randomly selected passages and their corresponding questions and options. The percentage of words changed in each passage by using each data augmentation operation will be indicated by the parameter α. For instance, when α set to be 0.2, by implementing the Easy Data Augmentation, all operations which

include word swap, synonym replacement, word insertion and word deletion will be implemented with 20% of total number of words in each passage (included their corresponding questions and options).

In this project, we implement the experiments which the parameters α set to be 0.1,0.2 and 0.5, and those parameter result in 58.64%, 59.26% and 58.26% respectively in ALBERT-base-v2 models. Based on their performance, we discover that this parameter set to be 0.2 obtained the best performance.

6 CONCLUSION AND FUTURE WORK

In this paper, we implemented 3 different BERT related models for this Question Answering task and the best models (Albert-xxlarge-v2) obtained 67.69% test accuracy on the RACE dataset. In the experiment, we try to fine tune different hyperparameter which include learning rate, L2 regulation, and max sequence length. Moreover, we implement the Data augmentation on the Albert-base-v2 models to increase 0.64 accuracy.

In future work, we suggest some extensions of this project for further research. The first one is that we should try more different deep learning such as the XLNet and other ensemble deep learning models. The next one we should consider that try to add some features (such as named entity recognition) to the models. Moreover, we should perform other Data Augmentation Methods such as Mixmatch and UDA. Finally, we hope that we can try more higher max sequence length such as 300 max sequence length to obtain better performance.

7 FINAL INSTRUCTIONS

7.1 CONTRIBUTION

WU Hao (20711289): Data Preprocessing, Model Setup/Coding, Experiment and Results, Discussion

CHAN Chun Kit (20728737): Literature Review, Methodology, Model Setup/Coding, Experiment and Results, Discussion

YANG Siting (20714786): Literature Review, Methodology, Model Setup/Coding

REFERENCES

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

Vanitha Guda, Suresh Kumar Sanampudi, and I Lakeshmi Manikyamba. Approaches for question answering systems. *International Journal of Engineering science and technology (IJEST)*, 3(2): 990–995, 2011.

Poonam Gupta and Vishal Gupta. A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4), 2012.

Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

Wilson L. Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433, 1953.

Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*, 2019.

Yang You, Jing Li, Jonathan Hseu, Xiaodan Song, James Demmel, and Cho-Jui Hsieh. Reducing bert pre-training time from 3 days to 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.