Question Answering for Reading Comprehension with BERT-based Approaches on SQuAD 1.1 and RACE

Shenshun Yao and **Yutong Zhang** and **Ziang Gao**
McGill University
Montréal, QC, Canada
{shenshun.yao,yutong.zhang2,ziang.gao}@mail.mcgill.ca

Abstract

In this paper, we propose a question answering (QA) system for text comprehension on BERT and its variants DistilBERT and RoBERTa. Our experiments involve fine-tuning BERT and comparing those models using the SQuAD 1.1 and RACE datasets. We collect exact match (EM) and F_1 scores for SQuAD 1.1 and accuracy for RACE as the metrics to measure each model's performance and make high-level observations about their relative strengths. Our results show that BERT-large give the overall best results on both datasets, but DistilBERT provides the best *bang for the buck* with respect to the performance given the same number of parameters. Finally, we also investigate some possible future research for optimization and improvement to our work.

1 Introduction

As more people benefit from smart voice assistants such as Apple's *Siri*, the question answering (QA) problem has become increasingly popular in the NLP research community. An interesting and useful application would be combining QA with language acquisition. Our motivation for this work originates from the text comprehension exercises we regularly practice for standardized tests while learning English, where one is required to answer a few questions after reading a passage. It is a huge relief and convenience for both students and teachers if they have easy access to such resources.

In this paper, we propose a QA system for English reading comprehension that receives a passage and a question and predicts the answer. Our system is based on Bidirectional Encoder Representations from Transformer (BERT) introduced by Devlin et al. (2019), a powerful transformer model trained on masked language modeling and next sentence prediction that exhibits excellent performance on multiple language tasks. The datasets we train on are

(1) **S**tanford **Qu**estion **A**nswering **D**ataset (SQuAD 1.1). It collects ~108K question-answer pairs on 536 Wikipedia articles, where each answer is a span of text from the passage. Figure 1 shows an example passage from SQuAD 1.1 and two question-answer pairs. (Rajpurkar et al., 2016)

(2) **R**e**A**ding **C**omprehension Dataset from **E**xaminations (RACE). It consists of ~28K passages and ~98K questions from English exams for middle and high school Chinese students, where each answer is a choice of multiple-choice questions. Figure 2 shows an example passage and two multiple choice questions with their answer keys marked in bold. (Lai et al., 2017)

The war was fought *primarily along the frontiers between New France and the British colonies*, from Virginia in the South to Nova Scotia in the North. It began with a dispute over control of the confluence of the Allegheny and Monongahela rivers, called the Forks of the Ohio, and the site of the French Fort Duquesne and present-day Pittsburgh, Pennsylvania. The dispute erupted into violence in the Battle of Jumonville Glen in May 1754, during which Virginia militiamen under the command of 22-year-old George Washington ambushed a French patrol.

$\mathbf{Q}_1$: Where was war fought?
$\mathbf{A}_1$: *primarily along the frontiers between New France and the British colonies*
$\mathbf{Q}_2$: When did violence start in war?
$\mathbf{A}_2$: May 1754

Figure 1: An example passage and two question-answer pairs in SQuAD 1.1.

2 Related Work

Ever since its advent, BERT has inspired a lot of interesting work to improve its performance or address its issue. Perhaps one of the most attractive upgrades is ALBERT (Lan et al., 2020). This is a lightweight version of BERT that shrinks down the

One of the most difficult problems a young person faces is deciding what to do. Some people, however, from the time they are six years old "know" that they want to be doctors or teachers or firefighters, but most of us do not get around to making a decision about a job until someone or something forces us to face the problem. Choosing a job takes time, and there are a lot of things you have to think about as you try to decide what you would like to do. You may find that you will have to take special courses for a particular kind of work, or you may find out that you will need to get enough knowledge for a particular job. Fortunately, there are a lot of people you can turn to for advice and help in making your decision. At most schools, there are teachers to give you information about jobs. And you can talk over your ideas with family members and friends who are always ready to listen and to offer suggestions.

$\mathbf{Q}_1$: The passage tells you ____ for a particular job.
A. you should have ideas when you are a child
B. it's impossible for you to get enough knowledge
C. you have to face the problem
D. you may enter a class to study
$\mathbf{Q}_2$: Making a decision about choosing your job ____.
A. needs friends
B. needs time
C. cost money
D. cost your ability

Figure 2: An example passage and multiple-choice questions in RACE. The answer key is marked in bold.

number of parameters by 89%[1] mainly through factorization of the embedding parameterization, yet it yields even stronger performance on 12 NLP tasks as reported by researchers. In fact, according to the RACE leaderboard[2] as of December 2020, Jiang et al. (2020) achieves the state-of-the-art 91.4% accuracy using an ensemble single-choice ALBERT combined with transfer learning. Although the current state-of-the-art result on SQuAD 1.1[3] (90.2 exact match score, 95.4 F_1 score) is achieved using LUKE (Yamada et al., 2020), the ALBERT-xxlarge model still provides a very competent performance (88.3 exact match score, 94.1 F_1 score).

There are also some similar QA datasets introduced in previous literatures. For instance, MCTest (Richardson et al., 2013) is in the same format as RACE but is based on children's stories. However, this dataset only contains 500 passages and 2,000 questions while RACE has significantly more data and is not domain-specific, that is, it covers much broader topics such as advertisements and scientific news. Therefore, our work to implement a BERT-based QA system using SQuAD and RACE builds on larger datasets and is more difficult.

3 Approach

In this section, we will briefly explain the models and our experimental settings.

3.1 Models

Our work mainly involves implementing BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2020) and RoBERTa (Liu et al., 2019) with some fine-tuning processes.

BERT. BERT is a method of pre-training language representations, meaning that we train a general-purpose "language understanding" model on a large text corpus (BooksCorpus, 800M words and Wikipedia, 2500M words), then use the model for downstream NLP tasks (fine-tuning) that we care about. BERT's architecture is a multi-layer bidirectional Transformer encoder based on (Vaswani et al., 2017) where the multi-headed self attention models context, the feed-forward layers computes non-linear hierarchical features, the layer norm and residuals makes training deep networks healthy and the positional embeddings allows model to learn relative positioning.

BERT can handle single sentences or pairs of sentences with a one token sequence. Each input sequence is generated by sampling two spans of text, the first of which receives the sentence A embedding and the second of which receives the sentence B embedding. The input representation sums up token embeddings (WordPiece embeddings), segment embeddings (representing whether the word belongs to sentence A or B), and position embeddings. Note that in RACE, there are three components of each example: passage, question and answers. We construct four input sequences for each example, one for each option among the four candidate answers. Following the notation of (Devlin et al., 2019), we input the passage as sentence A and the concatenation of the question and the candidate answer as sentence B. The input sequence can be denoted as

[CLS] Passage [SEP] Question + Option [SEP].

The pre-training for BERT includes two novel unsupervised prediction tasks: masked language modeling (LM) and next sentence prediction. The

[1] This comparison is based on ALBERT-base and BERT-base, which contains 12M and 110M parameters, respectively.

[2] Retrieved on December 15, 2020 from https://qizhexie.com/data/RACE_leaderboard.html.

[3] Retrieved on December 15, 2020 from https://rajpurkar.github.io/SQuAD-explorer/.

masked LM task involves masking some percentage of input tokens at random and predicting only those masked tokens. The next sentence prediction task is specifically designed to bolster question answering ability, as developing an understanding of the relationship between two sentences is not directly captured by language modeling.

In order to fine-tune BERT, we train two new sets of parameters, the starter vector $\mathbf{S} \in \mathbb{R}^H$ and the end vector $\mathbf{E} \in \mathbb{R}^H$. Given $\mathbf{T}_i \in \mathbb{R}^H$, the final hidden vector from BERT for the i^{th} input token, we can calculate the probability of word i being the start of the answer span as a dot product between T_i and S followed by a softmax over all of the words in the paragraph, namely,

$$\mathsf{P}_i^{\mathbf{S}} = \frac{e^{\mathbf{S}\cdot\mathbf{T}_i}}{\sum_j e^{\mathbf{S}\cdot\mathbf{T}_j}}, \quad \mathsf{P}_i^{\mathbf{E}} = \frac{e^{\mathbf{E}\cdot\mathbf{T}_i}}{\sum_j e^{\mathbf{E}\cdot\mathbf{T}_j}}.$$

The end of the answer span is similarly computed. The maximum scoring span is used as the prediction.

DistilBERT. DistilBERT (Sanh et al., 2020) is a distilled (approximate) version of BERT, retaining 95% performance but using only half the number of parameters. Specifically, it does not has token-type embeddings, pooler and retains only half of the layers from BERT. DistilBERT uses a technique called distillation, which approximates the original BERT, i.e. the large neural network by a smaller one. The idea is that once a large neural network has been trained, its full output distributions can be approximated using a smaller network.

RoBERTa. RoBERTa (Liu et al., 2019) is an improved recipe for training BERT models with simple modifications including: (1) training the model longer, with bigger batches, over more data; (2) removing the next sentence prediction objective; (3) training on longer sequences; (4) dynamically changing the masking pattern applied to the data.

3.2 Experiments

Our experiments include the following parts: (1) fine-tune BERT on SQuAD 1.1; (2) fine-tune BERT on RACE; (3) implement DistillBERT and RoBERTa on SQuAD 1.1. All experiments are carried out using PyTorch on a 3.6 GHz Intel i9-9900k CPU with 32 GB RAM and an NVIDIA GeForce RTX 2080 Ti GPU. Due to the limited time and GPU resources, we only dedicate to fine-tuning BERT. For the remaining models, we loaded and ran the pre-tuned models adapted from the Huggingface[4] library as a performance comparison.

Model	$\boldsymbol{F_1}$	**EM**
BERT$_{\text{base}}$	85.7	83.6
BERT$_{\text{large}}$	91.4	88.8
DistilBERT (base)	84.3	82.1
RoBERTa (base)	80.0	67.9

Table 1: F_1 and exact match (EM) scores for BERT (base and large), DistilBERT and RoBERTa models we evaluate on SQuAD 1.1.

For the scope of our project, we use the official SQuAD 1.1 train and development sets. This dataset already has a train / test split, but we are going to further divide up our training set to use 98% for training and 2% for validation. This validation set will help us detect over-fitting during the training process. Moreover, for the RACE dataset, we split 5% data as the development set and 5% as the test set for RACE-M, RACE-H and RACE, respectively. To evaluate our models we use the Exact Match (EM) score (a binary measure of whether the system output matches the ground truth answer exactly) and F_1 score

$$\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})}$$

for SQuAD 1.1 and accuracy for RACE.

4 Results

Table 1 shows the F_1 and exact match (EM) scores on SQuAD 1.1 for different models, namely, BERT-base, BERT-large, DistilBERT and RoBERTa. Both BERT models produce promising results with BERT-base exceeding BERT-large by 5.7 and 5.2 in F_1 and EM scores, respectively. Notice the results we report for DistilBERT verify the claim made by Sanh et al. (2020) that DistilBERT preserves 95% of BERT's performance. This also shows that DistilBERT provides a better *bang for the buck* with respect to the performance given the same number of parameters Surprisingly though, our RoBERTa model only achieves 67.9 and 80.0 in EM and F_1 scores, respectively, which is far from our expectation. A possible explanation would be the pre-tuned version of RoBERTa we apply should be further trained on SQuAD 1.1.

[4] https://huggingface.co/

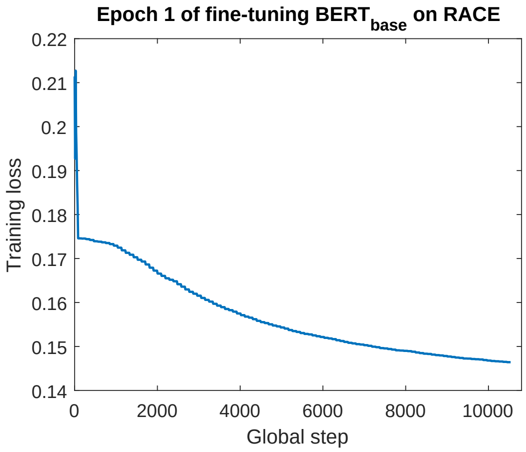

Figure 3: Training loss vs. global step count for one epoch of our BERT-base fine-tuning process on RACE. Training takes 2 hrs 50 min on our GPU with 87,866 examples and 10,983 global steps in total.

Model	RACE	RACE-M	RACE-H
BERT$_{\text{base}}$	54.3%	59.7%	52.1%
BERT$_{\text{large}}$	57.2%	63.6%	54.5%

Table 2: Evaluation accuracy on the RACE dataset after fine-tuning BERT for 1 epoch. As of hyperparameters, we choose batch size of 32 and learning rate of 5×10^{-5} for BERT-base, batch size of 8 and learning rate of 10^{-5} for BERT-large.

According to Figure 3 and Table 2, we found that after fine-tuning BERT for 1 epoch on the RACE dataset, the accuracies for both BERT-base and BERT-large are not competitive compared with other recent results on the RACE leaderboard but still beat many before 2019. To our knowledge, we can still improve our results by training more epochs or using ensemble methods.

Observing from the Table 2, BERT performs better in RACE-M than RACE-H. Since middle school's reading is relatively easier than high school's reading, it is intuitively true that models are more likely to yield higher accuracy on lower-level datasets. Besides, we also observe that BERT-large has better results than BERT-base. It is reasonable to infer that equipping with more parameters does improve model's performance.

5 Discussion

In this section, we will first explain the mathematical correctness of our methods with a heuristic illustration. Then we would like to point out some potential weak spots of our approaches to obtain a robust evaluation.

5.1 Mathematical Heuristic Justification

Heuristic justification of our experiments from the perspective of conditional probability can illustrate that our approach sends the original BERT onto the right track of improvement. Let $\mathbf{B}$ denote some fixed BERT model and $\mathbf{G}$ denote the information space which is used in the pre-training of $\mathbf{B}$. Besides, in the reading comprehension problem, we assume a paragraph or a short article denoted by $\mathbf{H}$ and the question $\mathbf{Q}$ is given to the BERT model $\mathbf{B}$. The reading comprehension problem has the following characteristics. For a correct solution $A \in \mathbf{A}$ to the question $\mathbf{Q}$, it is very likely that

$$\mathsf{P}(A \mid \mathbf{H}) \geq \mathsf{P}(A).$$

For an incorrect solution $A' \in \mathbf{A}^{\complement}$, it is very likely that

$$\mathsf{P}(A' \mid \mathbf{H}) \leq \mathsf{P}(A').$$

This observation shows that there is a potential improvement by using conditional information. Conditioning on some information polarizes the correct predictions and the incorrect predictions. The inference can be further developed by considering the tower property in the context of conditional probability. If we further assume $\mathbf{H} \in \mathcal{H}$, $\mathbf{G} \in \mathcal{G}$ where $\mathcal{H}$, $\mathcal{G}$ are two sub σ-algebras and $\mathcal{H} \subset \mathcal{G}$, it then follows that

$$\mathbb{E}\left[\mathbb{E}\left[\mathbf{B} \mid \mathcal{H}\right] \mid \mathcal{G}\right] = \mathbb{E}\left[\mathbf{B} \mid \mathcal{H}\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbb{B} \mid \mathcal{G}\right] \mid \mathcal{H}\right].$$

This observation implies correctness of our methods that the smaller-sized conditioned information $\mathbf{H}$ matters more than the total information $\mathbf{G}$.

5.2 Limitations

We explore various extensions to BERT and the reading comprehension problem in this paper, however, we also observe some clear drawbacks. First, it is difficult for our work to reach the same level of performance as those state-of-the-art research. Part of the reason why this happens is due to the time and computational resource constraints. In particular, we do not perform any hyperparameter search or train longer for those downstream task. We also choose not to implement some common algorithmic tricks such as data augmentation since our datasets are already very large.

Furthermore, as Bender and Koller (2020) point out in their paper, we should pay attention to the fact that large language models (LMs) like BERT cannot learn meaning despite their excellent performance on multiple language tasks. We have excessively relied on large LMs to solve natural language understanding (NLU) problems. However, the authors warn that overusing large LMs may end up with incorrect research objective, hence it is worth considering the *top-down* approach to tackle NLU problems. In other words, the focus is on the remote end goal of offering a complete, unified theory for language understanding field. Even if these neural network models are capable of extracting *some* relations from language datasets, it is questionable that these models are able to truly understand and correctly use languages.

6 Conclusion and Future Work

We present some variants of a BERT-based QA system with a collection of experiments to understand their behaviors and relative performance. Specifically, we compare the original BERT, DistilBERT and RoBERTa models on the SQuAD 1.1 and RACE datasets. Our results show that these models are capable of producing highly reasonable answers given a passage and a few questions as Table 1 and Table 2 illustrate. There are a number of interesting directions for future work. For example, we could create ensemble BERT models or augment the training set by incorporating English thesaurus and switching existing words to their synonyms. Since these huge pre-trained language models usually involve millions or even billions of parameters, it would also be exciting to push the limit even more like DistilBERT and further reduce the size of these models while retaining comparable performance as their larger counterparts in the future. Besides, it is worth noting the "complement" of the current problem in which a model is trained to produce a small paragraph from a given topic. Generating arguments with proper logic is a much harder problem since the model is required to output more sentences based on limited information.

Statement of Contributions

Shenshun mainly contributes to implementing our QA system and describing of our approach and results. Yutong and Ziang are responsible for fine-tuning and evaluations of those language models. Also, Yutong presents the introduction of our work and related research in the field. Ziang proposes the initial idea of this work and discuss the strengths and limitations of this paper in his writing.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yufan Jiang, Shuangzhi Wu, Jing Gong, Yahui Cheng, Peng Meng, Weiliang Lin, Zhibo Chen, and Mu li. 2020. Improving machine reading comprehension with single-choice decision and transfer learning. *arXiv preprint arXiv:2011.03292*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Matthew Richardson, Christopher J.C. Burges, and Erin Renshaw. 2013. MCTest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203, Seattle, Washington, USA. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.