A BERT based model for Multiple-Choice Reading Comprehension

Kegang Xu
tosmast@stanford.edu

Jingjie Tin
jjtin1@stanford.edu

Jungyoun Kim
jyk423@stanford.edu

Abstract

In this paper, we proposed a novel Deep Comatch Network (DCN) based on a BERT model which significantly improved the performance compared to baseline model in RACE dataset. In Deep Network, we designed a novel comatch attention layer following 5 layers of coattention after Bert encoder. Our DCN benefits from the question-aware passage representation and answer-aware passage representation through the attentions between Passage/Question and Passage/Answer. We fine-tuned hyperparameter of the baseline model and got 62.2%. We applied Easy data augmentation and gained 0.6% performance. We achieved 66.2% accuracy on our DCN model. Finally, we build the ensemble model based on all the models and reached 67.9% performance which could rank top 6 in Race lead board.

1. Introduction

Automated reading comprehension can be applied to many commercial applications including financial reports, technical support and troubleshooting, customer service, and the understanding of healthcare records. This project focus on automated multiple-choice reading comprehension on RACE [1] dataset. Compared to SQuAD[2], the answers of RACE could not directly be extracted from the passage. Answering the question needs more level reasoning and it has the following challenges:

- Created by domain experts to test high and middle students reading comprehension skills and requires a high level of reasoning and calculation techniques.
- Includes a wide variety of question types, like Summarization, Inference, Deduction and Context Matching.
- A Broad coverage in various domains and writing styles.

2.Related work

In Literature, previous papers focused on pairwise sequence matching to improve the reading comprehension capability of Neural Networks. This is performed by either matching a passage and the concatenation of the passage's questions and answers [3], or by matching a passage with its question before selecting a possible answer [4]. In Literature, many types of attentions have been proposed to enhance neural network reasoning in the passage level. Xu et al.[5] used multi-hop reasoning mechanism and proposed the Dynamic Fusion Networks. Zhu et.al [6] proposed the Hierarchical Attention Flow model in order to better model the interactions among passage, questions, and candidates. Bidirectional Encoder Representations from Transformers (BERT) [7] achieved state-of-the art in eleven NLP tasks. Inspired by Dual Comatching Attention[12], we proposed Deep Comatch Networks based on the output BERT hidden state. In this project, we also fine-tuned the base model with all kinds of hypermeters and explored Easy Data Augmentation, Attention over Attention[11] and ensemble all of the models to achieve the best accuracy in multiple choice reading comprehension.

3. 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

Table 1. Race Dataset

RACE dataset includes middle and high dataset. The total number of passages and questions are 27,933 and 97,687 respectively. Middle dataset averages about 250 words per passage while the High dataset averages 350 words per passage.

3.1 Data Preprocess

We concatenate a passage, a question, and an option together with special tokens CLS and SEP as the input sequence for BERT model [4]. Thus, we will have 4 of such inputs for each question.

Input: [CLS] passage [SEP] question [SEP] option 1 [SEP]
[CLS] passage [SEP] question [SEP] option 2 [SEP]
[CLS] passage [SEP] question [SEP] option 3 [SEP]
[CLS] passage [SEP] question [SEP] option 4 [SEP]

Output: the label of an option

3.2 Data Augmentation

Easy Data Augmentation (EDA) [10] was originally developed to enhance text classification on small data sets. EDA intends to create the new and augmented passages by the following operations:

- **Synonym Replacement:** Randomly choose several non-stop words. Replace each of these words with one of their synonyms randomly
- **Word Insertion:** Find a random synonym of a random non-stop word in the sentence. Insert that synonym into a random position in this sentence.
- **Word Swap:** Randomly choose two words in the sentence and swap their positions.
- **Word Deletion:** Randomly remove words in a sentence with a lower probability.

Hence, EDA expands the size and diversity of the existing dataset. We hope this could prevent overfitting during training and help to build a more robust model.

4. Network Architecture

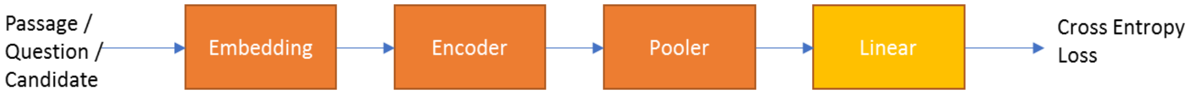

Figure 1. Baseline model

BERT [2] achieved the state-of-the art performance in 11 NLP tasks in 2018. Our baseline multiple-choice model would add one linear layer and SoftMax layer over the final hidden layer to predict the correct answer. Then we will fine-tune pretrained BERT base model [3] on RACE dataset with all kinds of hyperparameters. The total number of encoder layers could be either 12 or 24 depending on base or large model.

4.1 Attention over Attention

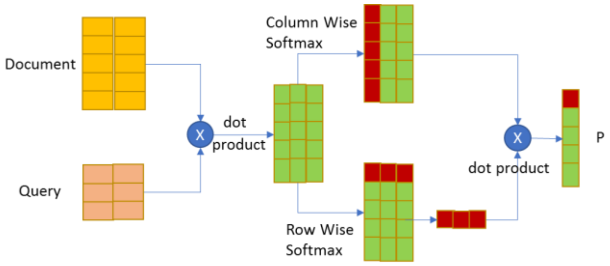

Figure 2 Attention Over Attention Layer

BERT includes the self-attention layer with multi-headed attention. It improves the performance in two ways. On one hand, it makes the model to capture the contextual correlation of words in long distance. On the other hand, it provides the ability to represent subspaces. This project will push the attention concept further and firstly introduce the Attention over Attention (AoA)[11] layer. It is a relatively new NN architecture which aims to place another attention mechanism over the existing document-level attention. Instead of solely summing or averaging individual attentions to get a final attention score for each word, an additional "importance" distribution on the query is carried out to determine which query words are more important given a single document word. Therefore, it could increase the information available to the network. A description of the architecture of AoA models is shown in Figure 2. We could get the column-wise softmax called α and row-wise softmax called β. Then $Si = \alpha^T \beta$. The final probability: $P(w|D,Q) = \sum_{i \in I(w,D)} Si \quad (w \in V)$

4.3 Deep Comatch Network

We designed a Deep Comatch Network by introducing the question-aware passage representation and answer-aware passage representation to fully explore the available information in the following triplet {Passage, Question, Answer}. Before getting to the comatch layer[12], our network goes through 5 layers of P2Q/Q2P and P2A/A2P coattention. The deep comatch attention inputs the hidden context of passages, question and answer which are output by Bert encoder Layer. In the end, the classifier layer will output the predicted answer after the maxpooling layer in the comatch block.

Firstly, the passage, question and candidate answer are encoded as shown in (1), where H^p, H^q and H^a are sequences of hidden states generated by BERT. Here, P, Q and A are the sequence length of the passage, the question and the candidate answer respectively, and L is the dimension of the BERT hidden state.

$$H^p = BERT(P),\ H^q = BERT(Q),\ H^a = BERT(A) \qquad (1)$$

where $H^p \in R^{P \times L}$, $H^q \in R^{Q \times L}$ and $H^a \in R^{A \times L}$ are sequences of hidden state generated by BERT.

Then we do coattention operation [13] between P2Q/Q2P and P2A/A2P using the following formulate. This is different from the previous papers. In our project, we also explored the relationship or attention between passage and answer.

$$W = \text{Coattention}(Q, K, V) = \text{softmax}\left(\frac{Q * K}{\sqrt{d}}\right)V \quad (2)$$

$$M^p = WH^a,\ M^a = W^TH^p,\ M^{p'} = WH^a,\ M^q = W^TH^p \quad (3)$$

$$S^p = \text{Relu}([M^a - H^a ; M^a \cdot H^a]W_1) \quad (4)$$

$$S^{p'} = \text{Relu}([M^a - H^a ; M^a \cdot H^a]W_2) \quad (5)$$

$$S^a = \text{Relu}([M^p - H^p ; M^p \cdot H^p]W_3) \quad (6)$$

$$S^q = \text{Relu}([M^q - H^q ; M^q \cdot H^q]W_4) \quad (7)$$

-/ · are the element-wise matrix subtraction and multiplication and [;] is the column-wise matrix concatenation. We use different weights so that we have S^p and $S^{p'}$.

Then we concatenate them after maxpolling.

$$C^x = \text{maxpooling}(S^x) \quad x \in (p,p',q,a) \quad (8)$$

$$C = [C^p;C^a;C^{p'};C^q] \quad (9)$$

Finally, we compute the cross entropy loss after a classifier layer.

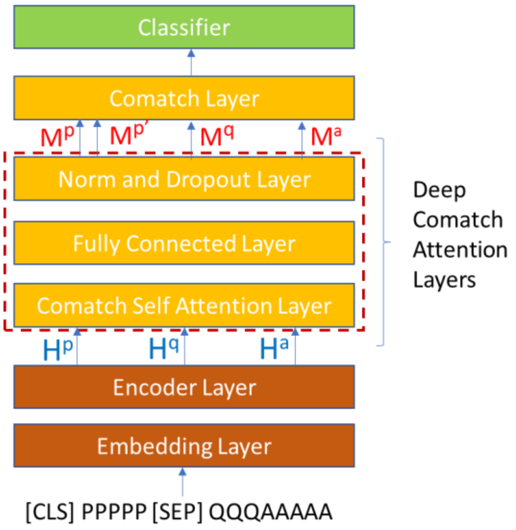

Figure 3. Deep Comatch Nework

4.4 Model Ensemble

After building different models with different hyper parameters such as large/base/cased/uncased BERT, DCN, EDA, we will assemble these models to produce a new model. Thereafter, we will explore the relative weight of each model to achieve a better performance.

4.5 Loss and Metric

In our project, the accuracy metric used to evaluate our model performance is defined as follows:

Accuracy = the number of correct answers / the number of total questions

Loss = Cross entropy Loss

5. Experiments

5.1 Hyperparameter tuning

In the project, we experiment the following hyperparameters on the base model.

1) Learning Rate

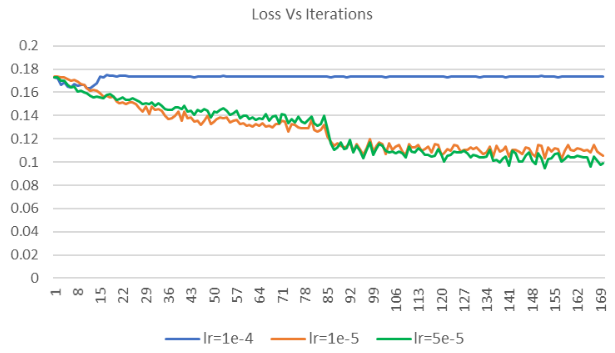

Figure 4. Loss Vs iteration with various Learning Rates

Based on the baseline model, we tried three initial learning rates, 1e-5,5e-5 and 1e -4. The results are shown in Figure 4. A learning rate of 1e-4 is represented by the blue line, and the loss was found to maintain a flat line which indicates that the learning rate is too big. For the learning rates of 1e-5 and 5e-5, both losses are decreasing with increasing iterations. However, a learning rate 5e-5 achieves the least loss and delivers the best performance. The unit of the iteration is 1K samples. In the middle of the figure, there is a big drop because of a programmed change in learning rates.

2) Freeze Bert Encoder Layers

Freeze Layers	Dev Accuracy
No Freeze layers	62.2%
Freeze first 1 layer	61.2%
Freeze first 3 layers	60.4%
Freeze first 5 layers	57.9%
Freeze first 7 layers	55.8%

Table 2. Accuracy Vs Number of freeze layers

We explored how the number of freeze layer impacts the performance. With the max sequence 320 and batch size 8 due to GPU memory limit, the results are shown in the above table. From the table, we know the performance is getting lower as the freeze number increases. And the performance gets much worse once the freeze layer number exceeds 3.

3) L2 Regularization

In addition, we explored weight decay parameters with L2 Regularization coefficients of 0.1, 0.01 and 0.001. We got the corresponding performance, 62.1%,62.2% and 62.1%. Hence, the best performance was obtained with an L2 regularization of 0.01.

4) Batch size and Max Sequence

Batch size and max sequence length were one of the critical hyper-parameters to design the BERT model.

Since the BERT is a huge network, we are using gradient accumulation technique to reduce the GPU memory requirement. We experimented three different batch size: 16, 24, 32, and 64. We started off with batch size of 32 for BERT base model training and we got 61.9% performance result; however, as we decreased the batch size to 24 and 16, accuracy increased a little bit to 62.3% and 62.5% respectively. Also when we increased the batch size to 64, the accuracy decreased. So it seems like batch sizes of 32, 24, and 16 falls into a good range to achieve the best model, but batch size 16 yielded the best performance, which is 0.22% higher than batch size 24 and 0.65% higher than batch size 32.

Hyper-parameter max sequence length plays a very critical role in the BERT based on model. As the max sequence length increases from 320 to 450, the performance increased gradually by 1.5%. We first tested on max sequence length equal to 320 on BERT base model and it resulted in 61.1% accuracy. As we increased the max sequence length, the performance of the model improved. For max sequence lengths 380, 420, and 480, we got accuracies 61.3%, 61.8%, and 62.6%. Thus, we concluded that the larger max sequence length can aid increasing the accuracy of our model. However, the model size is proportional to max sequence length and leads to more memory requirement. We could not explore more length due to the resource limit.

5.2 Data Augmentation

We augmented the overall training set by 10%, exactly half from RACE-middle and another half from RACE-high. We only applied the augmentation on the randomly selected passages and conducted the same preprocess by concatenating them to original questions and options. In EDA, there is a parameter α which indicates the percentage of words changed in each passage by each augmentation technique. For example, if α is set to 0.1, our EDA implementation would change 10% of words to their synonyms, random word insertion and swap will occur on 10% of the total number of words, and 10% of the words will be randomly deleted. We experimented with this parameter set to 0.1, 0.2, 0.3, and 0.5, and each parameter resulted in 65.2%, 65.6%, 64.9%, and 64.9% accuracy respectively. Thus, α =0.2 gave the best performance although the difference in accuracy wasn't very large. However, compared to the accuracy obtained by large BERT model without EDA, 10% of addition of training data augmented with augmentation parameter 0.2 can improve the model accuracy by 0.6%.

Therefore, we found the best performance reached 65.6% as shown in Table 3 and compared to the original RACE dataset, it's about 0.6% of gain. There are two possible reasons for not gaining so much. One is we might need to tune more hyper parameter of EDA, using different ratios for different operation, synonym replacement should be less noisy but other 3 operations add more noises. The other reason is Easy Data Augmentation prefers to work well on a small dataset, but our RACE includes 6400 passages and is a big dataset comparatively.

5.3 Deep Comatch Network

We tried to apply some new attention ideas to Multiple Choice Reading Comprehension on Race. The first idea which was explored was the application of an additional Attention over Attention layer to the base BERT model. However, poor accuracy of 25.1% was achieved, which is attributed to the fact that the self-attention layer of BERT conflicts with the AoA layer when both layers are combined in the encoder layer.

Model	**Accuracy**
Base BERT	61.6%
Fine-tuned Base BERT	62.6%
Large BERT	65.0%
Easy Data Augmentation	65.6%
Deep Comatch Network	66.2%
Ensemble of models	67.9%

Table 3. Accuracy of models

Then we turned to designing Deep comatch Network inspired by[12]. We created multiple layers of Deep Comatch Network following Bert hidden output. Due to GPU resource limit, we trained DCN with 3 layers. We got 66.2% accuracy in single mode of DCN shown in the below.

5.4 Ensemble

After exploring all of these different models, we created the ensemble model to achieve the best performance. We used random search the weighted accuracy between Base, Large, EDA and DCN while keeping the best performance of the model. Finally, we got 67.9% accuracy in ensemble model as shown in Table 3.

6. Analysis

The below figures show the comparison between DCN and Base model. DCN performs better than Base in both high and middle dataset. DCN performance is even much better in long and complex passage in high dataset.

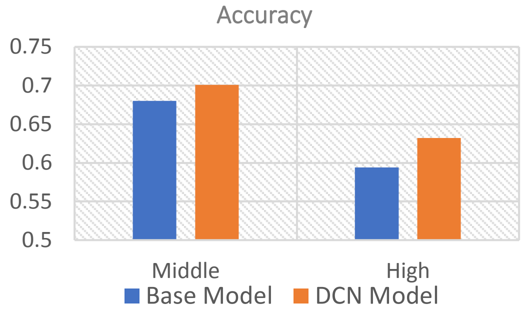

Figure 5 Base Accuracy Vs DCN Accuracy

Crucially, upon analysis of the questions which BERT had answered wrongly, it was discovered that the model was unable to perform simple numerical reasoning. For example, in the following mislabeled case, the first option is correct. Although the predicted (second) option has higher probability, the answer should have been 45-18=27 years ago.

Question: *From the passage we can know that __*

Options: *"Stewart was depressed at one time", "**Stewart lost his left arm 22 years ago**", "Stewart never complained about the unfairness of life", "Stewart was persuaded to kayak through the Grand Canyon"*

Passage: *For most of his life, the **45**-year-old man has lived with only his right arm. He lost his left arm ... when he was **18**. He became a bitter young man, angry at the unfairness of what had happened, and often got into fights.*

The following samples shows DCN correctly predicted the answer *"Today's News"* but the baseline model failed. It means DCN could more accurately learn what the news means and its relationship with the time.

Questions*: "In which part of a newspaper can you most probably read the passage?"*

Options*: "Today's News","Culture","Entainment", "Science"*

Passage*: "Three cattle farms in Andong...were..infected with ...disease,Nov.2 2010,Thursday....On Monday, the disease .. detected on two pig farms in Andong...The laboratory tests today showed that all three cattle farms were infected with the disease," an official said. Two newly infected cattle farms...indicating the disease will likely continue... has culled more than 33,000 animals ...No suspected cases ...*

7. Conclusion

We presented a novel Deep Comatch Network by adding several comatch attention layers after BERT hidden output base on Dual Comatch Network. In the end, we finally got 66.2% accuracy in a single model and 67.9% in an ensemble model. Compared to the state of the art performance 69.7%, the reason we are a little behind might be that we have to freeze several layers and could not explore the higher sequence due to the GPU resource limit. We found using larger max sequence was helpful. Institutively it is true since the length of some passages exceeds 400 words. When the maxsquence increase 320 to 450, the performance gains by 1.6%. We also experimented Easy Data Augmentation. This brought us 0.6% performance improvement. It wasn’t dramatic since the number of original training dataset was already huge.

In the end, we use random search to find the weighted of base, large, EDA and DCN in ensemble model. We gained a total of 1.7% increase in accuracy compared to DCN.

We also experiment Attention over Attention, but it seems hard to be integrated to Bert. And the result does not look to help much.

Our models and datasets were too large that it took a very long time to train and test each experiment. In the future works, we will train DCN on EDA dataset of RACE. We will explore more deep layers of DCN with bigger max sequence length. We would like to train on another dataset first and conduct a transfer learning on RACE dataset.

8. Contribution

All three authors equally contribute the project. Kegang Xu mainly worked on the architecture of the network, and race baseline model training. Jingjie Tin focused on the Attention Over Attention design and train the network. Jungyoun worked on dataset preprocess and augmentation and ensemble model training. We all contributed to the tuning, debugging, and writing of the final project.

Our code is on Github:
https://github.com/tosmaster/bert-race/

References

[1] Lai, Guokun and Xie, Qizhe and Liu, Hanxiao and Yang, Yiming and Hovy, Eduard,2017. RACE: Large-scale Reading Comprehension Dataset from Examinations. arXiv preprint arXiv:1704.04683.

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

[3] Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. 2016. Attention-based convolutional neural network for machine comprehension. arXiv preprint arXiv:1602.04341 .

[4] Haichao Zhou, Wei Furu, Qin Bing, and Liu Ting. 2018. Hierarchical attention flow for multiple-choice reading comprehension. In Proceedings of AAAI Conference on Artificial Intelligence.

[5] Xu, Y.; Liu, J.; Gao, J.; Shen, Y.; and Liu,X. 2017. Towards human-level machine reading compre-hension: Reasoning and inference with multiple strategies.arXiv preprint arXiv:1711.04964.

[6] Zhu, H.; Wei, F.; Qin, B.; and Liu, T. 2018.Hierarchical attention flow for multiple-choice reading com-prehension. InProceedings of AAAI-18.

[7] 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.

[8] https://github.com/huggingface/pytorch-pretrained-BERT

[9] Alec Radford, Karthik Narasimha, Tim Salimans, lya Sutskever, 2018. Improving Language Understanding by Generative Pre-Training.

[10] Jason W Wei and Kai Zou. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196, 2019.

[11] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. arXiv preprint arXiv:1607.04423,2016.

[12] Shuailiang Zhang, Hai Zhao, Yuwei Wu, Zhuosheng Zhang, Xi Zhou, Xiang Zhou, Dual Co-Matching Network for Multi-choice Reading Comprehension. arXiv:1901.09381,2019.

[13] Chadha,Ankit;Sood,Rewa, BertQA - Attention on Steroids,2019, https://github.com/ankit-ai/BertQA-Attention-on-Steroids.