Project Report: Using BERT for Qualitative Reasoning

Zhikang Zhang zzhan362@asu.edu
Jianfeng Wu jianfen6@asu.edu
Weihai Shen wshen24@asu.edu
YanCheng Wang ywan1053@asu.edu

Abstract

In this project, we will use BERT, Bidirectional Encoder Representations from Transformers (Devlin et al., 2018), which is a newly proposed language representation model, to solve qualitative reasoning problem. BERT has been proved to be very powerful that obtains the state-of-the-art results on almost all natural language processing tasks. The dataset we are using is QUAREL (Tafjord et al., 2018)

1 Introduction

Many natural language questions require recognizing and reasoning with qualitative relationships, especially in science, economics and medicine domain. Existing qualitative model based tools can support such reasoning. However, the semantic parsing step which maps natural language questions to these models is proved to be quite a challenging task. In this project, we will use recently proposed language representation model called BERT to tackle this difficulty and We build two different models to compare with each other. One is similar with the last home work, which used BERT to embedding the sentences and the other is based on the next-sentence-prediction model. We will introduce the two model in detail in the following sections.

2 Problem Description

The dataset we would use in this project is called QUAREL, which was proposed by (Tafjord et al., 2018). This dataset has 2771 questions relating 19 different types of quantities. And example of the data sample in this dataset is like fiture 1

BERT is a general language representation model. It models the probablity distribution of the given language datasets. To be more precise, it takes a tokenized sentence or a pair of tokenized sentences as input. The output is the hidden states of each layers which would be used as features for specific task. The output can't be directly used as the result of our task. We will need to design additional component that takes the raw data and the output of BERT as input and outputs the final result.

Qualitative statement:
Alan noticed that his toy car rolls further on a wood floor than on a thick carpet. This suggests that:
Optional answers:
(A) The carpet has more resistance
(B) The floor has more resistance
Correct answer: (A) The carpet has more resistance

Figure 1: Example for qualitative statement and answers in QUAREL dataset

This dataset contains total of 2771 questions. We will follow the same setup as in (Tafjord et al., 2018), dividing the original dataset into training set and test set. The metric to measure the performance of the model would be the accuracy of the answering, or the percentage of correct answers.

3 Model Description

Different from other models, like OpenAI GPT Radford (2018) and ELMo Peters et al. (2018), BERT has a architecture with a multi-layer bidirectional Transformer encoder as shown in Fig. 2. By connecting both left and right context in all layers, BERT could model complicated context information.

Qualitative reasoning can be viewed as a specific type of Question-Answering problem. Given a few sentences of description and a question, we need to choose the right answer from the given two choices.

We have proposed two approaches to solve this

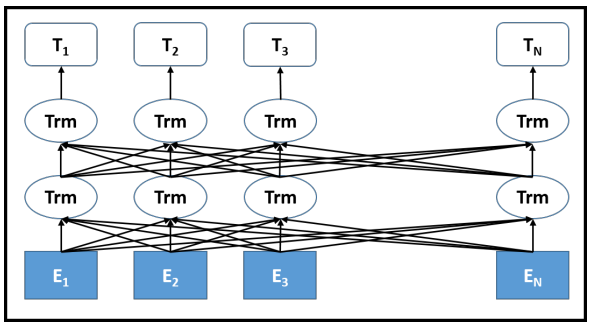

Figure 2: Pre-train architecture of BERT uses a bidierctinal Transformer. It jointly conditioned on left ad right context in all layers.

problem. For the first approach, we designed procedures to formulate it into a classification task and then solve it using a classifier we designed. For the second approach, we formulate the problem into a next-sentence prediction task then solve it.

3.1 Classification

3.1.1 Fine Tuning

We take the pretrained BERT-base uncased model as our starting point due to limited computation resources. The QUAREL consists of 1941 training samples, 278 dev samples and 552 test samples. For each sample, we produce two corresponding sentences: one is the original statement with correct answer, the other is the original statement with incorrect answer. We take all the statements with correct answers of train set to form the text corpus for fine tuning. The method we used to fine tune is LM Fine-tuning. Roughly speaking, it added an intermediate step in which the model is fine-tuned on text from the same domain as the target task and using the pre-training objective before the final stage in which the classifier head is added and the model is trained on the target task itself. More details can be found in code.

3.1.2 Feature Generation

After the fine tuning step, we transform all the samples into features using fine-tuned BERT model. BERT-base uncased model has 12 transformers, it produces 12 768-dimension vector for each word in the sentence. If we use all the features produced, the dimensionality of the features would be too large which would brings a lot of difficulties for later classification task. So instead, we only use the last feature vector of each word.

As the number of words varies from sentence to sentence, we first identify the longest sentence in the dataset which contains 137 words. Then for each word we transform it into a 768 dimensional vector, concatenate all the vectors together and use zero-padding to form a 137*768 dimentional vector. This would be the input of our classifier. The label is 1 if the answer is true, it's -1 otherwise.

3.1.3 Classifier Design

We designed a neural network to be our classification model. It contains three fully connected layers and two batch normalization layers. First layer is a fully connected layer, the input size if 137*768, the output size is 500; Second layer is a batch normalization layer ; Third layer is a fully connected layer, the input size is 500, the output size is 200; Fourth layer is a batch normalization layer; Fifth layer is fully connected layer, the input size is 200, the output size is 1. The activation function we used is tanh. We use the batch size of 50. The number of epoches is 100. The optimizer we used is Adam.

3.2 Next Sentence Prediction Model

This model is the same with the method method in task 2 in (Devlin et al., 2018). Since BERT has a architecture with a multi-layer bidirectional Transformer encoder, it can always understand the relationship between two text sentences. Therefore, we adapt this method to predict the correct answer from the two potential choice and the correct answer should be the next sentence.

3.2.1 Data Pre-processing

For this qualitative reasoning problem, the input representation is a pair of text sentences in a token sequence, which includes the question sentence and the answer sentence. And the problem can be converted to a binary classification task since each of the questions is given two answers and one of them is correct while the other one is a wrong answer. The positive sample consists of one question statement and the correct while the negative one is the pair of the same question statement and the incorrect answer. And each sample has a separation mark to divide the question statement and answer. Besides, the sample also includes a token which consists of the label. Since there are only two answers, we only have two labels here.

After we have combined the question statement and answer into a single sequence, we start to em-

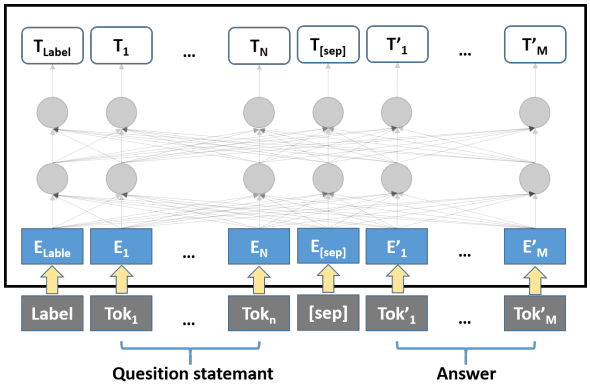

Figure 3: Sentence Pair Classification Task for the qualitative reasoning question.

bedding the whole sentence. The question statement and the answer are separated by a special token ([sep]). And a class label is added to the beginning for the input as shown in figure. 3.

3.2.2 Model Design

The next step is to build and train this model. Sentence A embedding is used to every token for the first question statement and another sentence B embedding to each of the token in the answer. After the embedding step, we generate a feature vector for each pair of question statement and answer.

Finally, another layer is applied at the end of the model for the fine-tuning step. For the sequence-level classification task, BERT fine-tuning is straightforward. In the first token of the input, we add a special label word embedding, which is the *Label* in figure. 3. We use $H \in \mathbb{R}^C$ as the input vector. Thus, the fine-tune classification layer can be represented by $K \in \mathbb{R}^{C*W}$, where W is the number of classifier labels. Here, the number of label is 2 since it is a binary classification. Therefore, we can calculate the probabilities $P \in \mathbb{R}^W$ by using a standard softmax layer, $P = softmax(HK^T)$. In the fine-tuning step, all the parameters of the model of BERT and K are updated to maximize the log-probability of the correct label.

In this project, the hyper-parameters are like followings.

- **Batch size:** 32
- **Learning rate(Adam):** 3e-5
- **Number of epochs:** 3

We run the above-mentioned model on the QUAREL dataset on a GPU, Geforce 1080 and it took 20 mintues to finish.

Model	Dev	Test
1st model	50.2	50.1
2nd model	55.4	53.3

Table 1: QUAREL results with different models. The first line are the results for the first model. The last line shows our results for the next-sentence-prediction model.

4 Experiment Results

4.1 Results

As the results shown in table 1, our model based on BERT can get better performance in the qualitative reasoning problem. While our classification approach does not perform well. In the training process, the training loss does not change too much.

4.2 Discussion and Future work

In our first approach, the fine-tuning step, the feature extraction step and the training of classifier all together determine the final accuracy of the solution. One main factor that may affect the performance is the design of the classifier. As the embedding of each word extracted by BERT having a size of 768, we really need a bigger network to perform the classification. Additionally, the training dataset is too small to train such a big neural network. Another fact that may cause the failure of the first approach is the variance of the length of the statement-answer pairs. In the training dataset, the longest one has 137 words, while some others have only 50 words. Thus, some samples are padded with too many zeros. That approach may be effective on dataset with more training samples and more standardized statement-answer pairs. We leave it in our future work.

For the second approach, we developed our model based on a library. However, there might be some misuse for the library or some format errors for the input data. We are still working on the code and try to improve the performance of the model.

5 Conclusion

In this project, we have proposed two approaches to solve the qualitative reasoning problem based on BERT model. For the first approach, we only get an accuracy of 50.1%. We transformed the QA

problem into a fine-grained binary classification problem, where the positive sample and the negative sample may only differs on small parts. While the small size of the training dataset and the large dimension of the feature does not support such a fine-grained classification task. In the second approach, we use bert to predict the next sentence with pre-trained model (bert-base-uncased) to get the accuracy on dev is 55.4 %, on test is 53.3 %.

References

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Oyvind Tafjord, Peter Clark, Matt Gardner, Wen-tau Yih, and Ashish Sabharwal. 2018. Quarel: A dataset and models for answering questions about qualitative relationships. *arXiv preprint arXiv:1811.08048*.