Intelligent Question Answering System Based on Machine Reading Comprehension

Qian Shang¹, Ming Xu²⁺, Bin Qin¹, Pengbin Lei¹ and Junjian Huang¹

¹College of Electronics and Information Engineering, Shenzhen University, Shenzhen, Guangdong, China
²Information Center, Shenzhen University, Shenzhen, Guangdong, China

* xuming@szu.edu.cn

Abstract. Question answering(Q&A) system is important for accelerating the landing of artificial intelligence. This paper makes an improvement on the Q&A system which uses the method of retrieval-machine reading comprehension (MRC). In the retrieval phase, we use BM25 to recall some documents and split these documents into paragraphs, then we reorder the paragraphs according to the correlation with the question, so as to reduce the number of recalled paragraphs and improve the speed of MRC. In the MRC stage, we design a multi-task MRC structure, which can judge whether the paragraph contains answer and locate answer accurately. Besides, we modify the loss function to fit the sparse labels during the training. The experiments are carried out on multiple data sets to verify the effectiveness of the improved system.

1. Introduction

Q&A system can help people acquire knowledge and solve questions. The Q&A system based on MRC can give short and accurate answers, and can perform well in understanding semantics and locating answers. However, the Q&A system based on MRC usually faces the following problems. First, it is difficult to achieve a balance between the effect of the model and speed of inference. If all recalled documents by retrieval are deeply inferred by MRC, it will cost a lot of time and computational resources. In addition, the premise of MRC is that the paragraph matching the question can be found in the document, which may not be true in reality. Therefore, the inferred answers may be unreliable.

In order to solve the above problems, we improve the intelligent Q&A system based on retrieval and MRC. The contributions of this paper are as follows:

- We design a two-stage document recall strategy at retrieval. First, we use BM25 to recall a certain number of documents. And then we split the recalled documents into paragraphs and design a structure named ParallelBERT-Performer to reorder the paragraphs. Only some paragraphs with high relevance to the questions will be selected for MRC;
- We design a multi-task MRC structure. After recalling relevant paragraphs, the first task is to let the model judge whether the question is relevant to the paragraph. The second task is to accurately locate the answer from the paragraph containing the answer. In this process, we also improve the loss function of the model because of the sparse labels.

2. Related work

Minjoon Seo et al. proposed BIDAF [1] in 2017, which doesn't summarize the text into a fixed-length
vector, but flows the vector to reduce the loss of early information weighting. The bi-directional attention mechanism proposed by the paper has become a part of a general encoder. Google Brain proposed QANET [2] in 2018. It uses CNN and self-attention instead of traditional RNN to build MRC model, which has a significant improvement in speed.

Google released BERT [3] in the end of 2018, which also covers simple data sets of MRC. Unlike the complex structural design of BIDAF and QANET, BERT can complete Chinese MRC tasks only by adding a simple answer predict module to extract the starting and ending positions of the answers. When using BERT to fine tune the downstream MRC task, the input form to BERT is [CLS] + question + [SEP] + text. Two full connection layers connect with the output vector of BERT to represent the index of the start position and end position respectively, and cross entropy loss function is used to optimize the parameters of BERT and downstream structure.

The $O(n^2)$ complexity of attention mechanism in Transformer limits the speed and computational power of the model, which makes the model difficult to land. To this end, Tay Y et al. [4] have summarized the improved methods of Transformer. Among them, Performer [5], proposed by Google in 2020, has a slightly lower performance and takes up less memory while improving speed. Due to the linearized attention calculation, Performer can make the forward calculation of the model faster and process a longer input sequence. Highway Networks [6] draws lessons from gate function in LSTM and alleviates the problem of gradient disappearance, so it can train deeper networks. We improve the performance of the model by adding these downstream structures in the right place to the model.

3. Algorithm design

Our algorithm is divided into two stages: paragraph recall and reordering; MRC for answer extraction.

3.1. Paragraph recall and reordering

The retrieval method we propose is pipeline. First screen the related documents from the massive document set, and then use deep networks to reorder the paragraphs for fine recall.

3.1.1. Rough paragraph recall with BM25. The task of this stage is to calculate the BM25 scores of a question $Q$ and document $D_k$ in set one by one. The inverse document frequency (IDF) of the word $q_i$ in $Q$ is the weight $\omega_i$, and the term frequency (TF) is the correlation score $R(q_i, D_k)$ between the $q_i$ and the $D_k$. We calculate the BM25 score between the $Q$ and $D_k$ as shown in Formula (1). Sorting all these scores and selecting the first K documents recalled as the result of preliminary screening.

$$\text{score}_{(Q,D_k)} = \sum_{i=0}^{\text{length}} \omega_i * R(q_i, D_k)$$ (1)

3.1.2. Reorder with ParallelBERT-Performer. BM25 uses the TF-IDF strategy but ignores the semantics of the context, which makes it difficult to guarantee a reliable recall rate under limited performance. Therefore, we split the documents obtained from the BM25 rough recall into paragraphs and reorder them, and then we propose the ParallelBERT-Performer as shown in Figure 1. The relevant paragraphs are recalled by sorting the output probabilities of binary classification.

In the training phase, two BERTs are used to extract the feature vectors of question $Q$ and paragraph $P$ respectively, and the deep semantic representation $E_Q$ and $E_P$ are obtained. We use Performer for semantic interaction between $E_Q$ and $E_P$, that is, $E_{interaction} = \text{Performer}([E_Q, E_P])$. After fully connecting a neuron to this interaction vector, the output probability $p$ of the model is obtained by using the Sigmoid activation function. The cross entropy between output and label is used to optimize the model by gradient descent and loss of back propagation, as shown in Formula (2).

$$\text{Loss} = -y \cdot \log p - (1 - y) \cdot \log (1 - p)$$ (2)
In the inference phase, all the fixed documents in set are split into paragraphs and stored in the right part of Figure 1 offline. The real-time inference results of the question and the candidate paragraph searched by BM25 are concatenated for semantic interaction, and then the correlation between the question and the paragraph can be inferred. After sorting the degree of correlation, several documents at the top of the list are taken as the result of the fine recall.

The complexity of Transformer is $O(n^2 \times d)$, where $d$ is the dimension of the hidden layer of Transformer and $n$ is the length of the input text. The length of common questions is about 20 to 30 words, while that of paragraphs is about 500 words. Therefore, in the inference phase, the model only infers the question, which can theoretically speed up $(500/(20\sim30))^2$, i.e. 250 ~ 600 times, compared to inferring all the data.

### 3.2. MRC for answer extraction.

Despite the rough recall and fine recall, we still can’t guarantee that the answer must be in the recalled paragraphs. Besides, we should avoid inferring different answers from different paragraphs. Therefore, while constructing the MRC task, we should also construct the task of whether the question and the paragraph match, that is, multi-task learning.

For a pair constructed by question $Q$ and candidate paragraph $P$, the model encodes, extracts features and interacts semantic information to gets the output vector $T$. The vector $T$ contains a lot of deep semantic information through multi-layer transformer. We design three tasks for this vector, as shown in Figure 2.
Figure 2. Multi-task MRC model

Task 1 is to predict whether $Q$ and $P$ are related. The output vector of the $cls$ position in the pre-training model is used to determine whether the two sentences are connected, so firstly, use the vector $E_{cls} \in \mathbb{R}^{1 \times h}$ to fully connect a neuron, and then the $Sigmoid$ activation function is used to get the matching degree $p_{(Q,P)}$ of the $Q$ and $P$. As shown in Formula (3), the cross entropy between the predicted result $p_{(Q,P)}$ and the real label $y_{yes,no}$ is calculated as loss for back propagation.

$$loss_1 = - (y_{yes,no} \cdot \log p_{(Q,P)} + (1 - y_{yes,no}) \cdot \log(1 - p_{(Q,P)}))$$  \hspace{1cm} (3)

Because there are paragraphs with no answer as negative samples in the training, the model loss is difficult to converge. To this end, we introduce Task 2 to assist Task 1 for learning. As shown in the following formulas, the shared $cls$ vector is passed through a layer of Highway Network to semantically filter $E_{cls} \in \mathbb{R}^{1 \times h}$ that contains complex semantic information, leaving a feature vector $H_{cls} \in \mathbb{R}^{1 \times h}$ that contains a simple relationship between texts.

$$H_{cls} = H(E_{cls}) \ast T(E_{cls}) + x \ast (1 - T(E_{cls}))$$  \hspace{1cm} (4)

$$T(E_{cls}) = Sigmoid(W^T E_{cls} + b)$$  \hspace{1cm} (5)

Bleu is an assistant tool for bilingual translation quality, which indicates the degree of similarity between machine translation text and reference text. $H_{cls}$ is fully connected to a neuron by $Relu$ activation function to get the predicted value $V$, which is used to reduce the mean square error (MSE) with the calculated $y_{bleu, score}$ to optimize parameters. The $M$ below is the length of the paragraph.

$$V = Relu(W^T H_{cls} + b)$$  \hspace{1cm} (6)

$$loss_2 = \frac{1}{M} \sum_{i=1}^{M} (V - y_{bleu, score})^2$$  \hspace{1cm} (7)

Task 3 is the MRC task. We use pointers to extract the starting and ending positions of the answer in the article. The cross entropy is modified because of sparse labeling. The hyper parameter $\theta \in (0 \sim 1)$ is introduced to give a greater weight to the token at the position of label “1”, so that the model pays...
more attention to the real task rather than fitting the label “0”.

\[
\text{loss}_\text{start} = - \frac{1}{N+M} \sum_{i=1}^{N+M} \left\{ \theta \ast y_{\text{start}_i} \ast \log p_{\text{start}_i}, \text{if } y_{\text{start}_i} = 1 \right\} \right.
\]
\[
\left. (1-\theta)(1-y_{\text{start}_i}) \ast \log(1-p_{\text{start}_i}), \text{if } y_{\text{start}_i} = 0 \right\}
\]
\[
\text{loss}_\text{end} = - \frac{1}{N+M} \sum_{i=1}^{N+M} \left\{ \theta \ast y_{\text{end}_i} \ast \log p_{\text{end}_i}, \text{if } y_{\text{end}_i} = 1 \right\} \right.
\]
\[
\left. (1-\theta)(1-y_{\text{end}_i}) \ast \log(1-p_{\text{end}_i}), \text{if } y_{\text{end}_i} = 0 \right\}
\]

Finally, the overall loss is Formula (10), where \( \alpha \) is a hyper parameter, which is used to adjust the emphasis of the model on different tasks when the loss level of the Task1 and Task3 is the same.

\[
\text{Loss} = \alpha(\text{loss}_1 + \text{loss}_2) + (1 - \alpha)(\text{loss}_\text{start} + \text{loss}_\text{end})
\]

4. Experiments

The experimental environment of this paper is CentOS7 of Linux, and the computational resources include 128G CPU memory and 2 Quadro GP100 16GB HBM2 graphics cards. We use python as the development language and pytorch as the framework of deep learning.

4.1. Paragraph recall and reordering

4.1.1. Dataset introduction. We conducted experiments on the data set COVID-19 Q&A Assistant (CQAA) [7]. CQAA is a competition released by the Beijing municipal government, and its data set was provided to help various industries understand the policy. The data set contains epidemic-related policy documents and answers to some questions. There are 5000 samples in training set, 1643 samples in test set, and 8844 documents in documents set. The training set sample is shown in Table 1. The model is required to give concise and correct answers to users' questions through learning.

| id          | 7721da4299f5337fbcf34eb59cdc43a9 |
|-------------|----------------------------------|
| docid       | f3bdcc0013733fb7a6a7f37549bb5e62 |
| question    | 如何处置哄抬物价、造假等行为?  (How to deal with the behavior of driving up prices, counterfeiting and so on.) |
| answer      | 给予处罚和刑事拘留 (Impose penalties and criminal detention) |

4.1.2. Analysis of experimental results. The results of the experiment are shown in Table 2. If BM25 is used simply to recall paragraphs, the recall rate is very low, although the speed is pretty fast. After adding BERT, the recall rate of the model is greatly improved, but the inference time is obviously impractical. If we use bleu_score to reorder the paragraphs recalled by BM25, the recall rate has been improved to a certain extent. This method suits for the scenes which requires not so high precision but pays attention to response speed.

ParallelBERT-Performer shows the same performance as the BM25+BERT in recall rate. The parallel structure of online question inferring and offline paragraph feature extraction greatly reduces the computation in the inference stage. Although it introduces an additional layer of Performer and BERT forward inference which slow down the speed, it still completes the reordering of 50 documents in only 1.5s, 60 times faster than BM25+BERT. The fast inference of the model is completed while the high recall rate is ensured. Therefore, this model is more suitable for the scenes which requires not only high accuracy but also high speed.
4.2. MRC for answer extraction

4.2.1. Dataset introduction. We carried out multi-task MRC experiments under two data sets to verify the generality of this method. The first is the CQAA MRC mentioned above, and the second is the Delta Reading Comprehension Dataset (DRCD) [8] released by Delta Research Center in China. It is an extractive MRC dataset based on traditional Chinese, sorting out 10014 paragraphs from 2108 wiki entries and tagging more than 30,000 questions. The model needs to find the start and end of the answer from the context. Besides, the training sets are divided into positive and negative samples, which can enhance the recognition of matching degree between question and paragraph. The positive samples are the actual samples of training sets. On CQAA, the negative samples are the top five non answer paragraphs by retrieval, and on DRCD, the negative samples are other randomly selected paragraphs without answer.

4.2.2. Analysis of experimental results. NEZHA [9] adds sinusoidal relative position coding in the training, which shows good performance in Chinese MRC tasks, and is the baseline model we choose. At the same time, we use QANET and BIDAF to compare, and measure the performance of the model through Rouge-L [10] and Bleu.

The experimental results are shown in Table 3. It can be seen that the performance of the model NEZHA+multi-task MRC we designed is better than that of NEZHA+MRC. The top5 Rouge-L on the data set CQAA reached 88.9% (top1 and top5 represent the number of recalled paragraphs in the retrieval). In addition, on the data set DRCD, F1 score and EM score are improved to a certain extent.

| Model                  | Top1 recall rate | Top5 recall rate | Top10 recall rate | Top20 recall rate | Time to infer 50 Document |
|------------------------|------------------|------------------|-------------------|-------------------|---------------------------|
| BM25                   | 0.575            | 0.773            | 0.828             | 0.870             | 66ms                      |
| BM25+BERT              | **0.863**        | **0.907**        | **0.921**         | **0.929**         | **90s**                   |
| BM25+bleu_score        | 0.633            | 0.801            | 0.855             | 0.895             | 0.4s                      |
| BM25+ ParallelBERT-Performer | **0.844**      | **0.899**        | **0.902**         | **0.921**         | **1.5s**                  |

5. Conclusion
We studied the retrieval-MRC based Q&A system deeply and proposed a practical framework. In retrieval phase and MRC phase, the corresponding improved models are designed and verified on the data sets. Experiments show that the retrieval model can achieve high speed and high recall rate, and the multi-task MRC model also shows its superiority and generality. In the future, we plan to apply and optimize this Q&A system on the school service platform.

Acknowledgments
The author would like to thank the research cloud platform of the information center of Shenzhen University for providing the computing environment.
References
[1] Seo M, Kembhavi A, Farhadi A, et al 2017 Bidirectional attention flow for machine comprehension. In: ICLR. Toulon
[2] Yu A W, Dohan D, Luong M T, et al 2018 Qanet: Combining local convolution with global self-attention for reading comprehension. In: ICLR. Vancouver
[3] Devlin J, Chang M W, Lee K, et al 2018 Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL. Minneapolis pp 4171-4186
[4] Tay Y, Dehghani M, Abnar S, et al 2020 Long Range Arena: A Benchmark for Efficient Transformers. arxiv.org/abs/2011.04006
[5] Choromanski K, Likhosherstov V, 2020 Dohan D, et al Rethinking attention with performers. arxiv.org/abs/2009.14794
[6] Srivastava R K, Greff K, Schmidhuber J 2015 Highway networks. arxiv.org/abs/1505.00387
[7] Beijing Municipal Bureau of Economy and Information Technology 2020 COVID-19 Q&A Assistant. https://www.datafountain.cn/competitions/424
[8] Delta Research Center 2019 Delta reading comprehension dataset. https://github.com/DRCKnowledgeTeam/DRCD
[9] Wei J, Ren X, Li X, et al 2019 NEZHA: Neural contextualized representation for chinese language understanding. arxiv.org/abs/1909.00204
[10] Lin C Y 2004 Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out: ACL Workshop. Barcelona pp 74–81