A NOVEL CHINESE READING COMPREHENSION MODEL BASED ON ATTENTION MECHANISM AND CONVOLUTIONAL NEURAL NETWORKS

CHIN-SHYURNG FAHN, YI-LUN WANG, CHU-PING LEE, AND MENG-LUEN WU

Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
No. 43, Keelung Road, Section 4, Da'an District, Taipei 106335, Taiwan, Republic of China
E-MAIL: csfahn@mail.ntust.edu.tw, M10615060@mail.ntust.edu.tw, D10215011@mail.ntust.edu.tw, D10015015@mail.ntust.edu.tw

Abstract:

This paper presents a novel machine reading comprehension model based on deep learning techniques in Chinese environment. In our manner, the training process can be performed using a general-level GPU, and the convergence of the training process can be accelerated for a shorter period of time. In the architectural design, two main constituting parts include Self-Attention Mechanism and Convolutional Neural Networks. To enhance the interaction between an article and questions, we carry out the operation of Context-Query Attention twice, so that our model is more effectively for acquiring the information of the questions related to the article and converges faster while training. In the experiment, we adopt the Delta Reading Comprehension Dataset for model evaluation in Chinese environment. The experimental results reveal that our model is able to reach the accuracy of 64.9% for EM and 79.0% for F1. The convergence time is less than 1 hour using the Titan XP GPU, and the memory usage is comparatively lower. The training performance is about 3 times faster than other models with state-of-the-art architecture.

Keywords:

Natural language processing, Chinese machine reading comprehension, attention mechanism, convolutional neural network, deep learning.

1. Introduction

In natural language processing (NLP), machine reading comprehension (MRC) using deep learning is a hot and useful topic nowadays. Although the accuracy of many existing MRC models has become higher, the scope of the models has also become larger, which requires greater computational power via equipping GPUs but need longer execution time yet. Unfortunately, the extensibility and flexibility of some recent MRC models are decreased as unexpected. In addition to this, most MRC models are developed for native English speakers, whereas in the world, Chinese, also used by billions of people, has less studies on their MRC models.

To overcome the above issues of MRC models, our proposed method accomplishes two epoch-making merits. One is to use Chinese as the main processing language, and another is to moderate the burden of model training and memory usage. There are two execution phases of our MRC model: the first is language pre-processing, and the second is reading comprehension. In the first phase, Chinese articles and questions are formatted into computer readable texts. In the second phase, we focus on reducing the computational cost while doing our best to attain high accuracy. Our architectural design is composed of Self-Attention, Context-to-Query Attention, and Convolutional Neural Networks (CNNs). Compared with other models, we emphasize the interaction between questions and an article to more efficiently find the answer of a question relevant to the article.

2. Related Work

In this section, we elaborate three types of architecture for MRC, including RNN-based, Attention- based, and Recently Huge Architecture. Here, RNN stands for the abbreviation of Recurrent Neural Network.

2.1. RNN-based Reading Comprehension

There are three RNN-based reading comprehension models built upon the SQuAD dataset [1], which comprise Match-LSTM [2], Bidirectional Attention Flow for Machine Comprehension (BiDAF) [3], and Ruminating Reader [4]. Table 1 lists the EM and F1 scores resulting from the three

models performing on the SQuAD dataset, which shows that the outcome of Ruminating Reader is the best, BiDAF is better, and Match-LSTM is the worst.

TABLE 1. Comparison of the Accuracy of Three Recent RNN-based MRC Models on Dataset SQuAD

Model	EM Score	F1 Score
Match-LSTM	64.1%	73.9%
BiDAF	67.7%	77.3%
Ruminating Reader	70.6%	79.5%

2.2. Attention-based Reading Comprehension

In 2018, Yu et al. released the entire QANet model [5] whose architecture is almost the same as that of BiDAF. The only difference is that the author defines an encoder block consisting of Self-Attention and Convolution layers instead of using the RNN to encode their MRC model. Accordingly, the QANet saves a lot of convergence time compared to the previous methods, and it has a significant improvement in the accuracy of reading comprehension. The experimental results manifest that in the NVIDIA p100 GPU environment, the QANet is 4.3 and 7.0 times faster than the BiDAF in training and prediction phases, respectively, and achieves high accuracy of EM and F1 scores by 73.6% and 82.7% individually on the SQuAD dataset.

2.3. Recently Popular Huge Architecture

Both the BERT and QANet serve as two fundamental reading comprehension models inspired to develop our MRC one. It is noticed that the base BERT already has high accuracy, whose EM score is 80.8% and F1 score is 88.5%; moreover, for the large BERT, the EM and F1 scores respectively arrive at 84.1%, and 90.9% [6].

All the above evaluation is carried out on the SQuAD dataset. Despite such preferable performance, the BERT requires a lot of memory space and computational resources to complete a training task. Furthermore, the modification on the architecture of BERT is rather difficult. Therefore, we aim at moderating computational resources unlike the BERT doing. By referring to the architecture of QANet [5], we will design a novel MRC model which only needs a low computational cost, but still keeps decent accuracy.

3. Natural Language Processing and Deep Learning

There are two essential parts in this section. In the first part, the pre-processing steps, including tokenization and embedding, are introduced; in the second part, the CNN architectures and Attention models for deep learning are described.

3.1. Tokenization and Embedding

Tokenization is a word segmentation method which divides continuous sentences or articles into character-based or word-based tokens. Because there is no blank space between two words in Chinese, word segmentation is a hard task in Chinese documents. To surmount this, we use the Jieba word segmentation toolkit [7] with the aid of an embedding thesaurus dictionary to tokenize words.

After that, in the embedding step, we convert each token into a positive integer index, and use the pre-trained embedding dictionary to convert the index into a vector in a contiguous space. Hence, we can transform words into computer readable vectors. In this task, we adopt the Tencent AI Lab Embedding Corpus, which is based on the Directional Skip-Gram proposed by Song et al. [8]. Figure 1 graphically shows the schematic diagram of Chinese tokenization and word embedding by virtue of the above methods. Consequently, each tokenized word or phrase in the given sentence can be changed into a 200-dimensional vector.

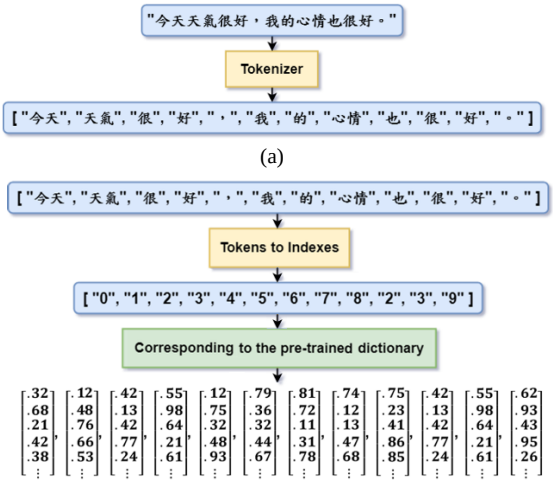

FIGURE 1. An example of Chinese tokenization followed by word embedding: (a) the tokenization result of a given Chinese sentence using the Jieba word segmentation toolkit; (b) the embedding result of (a) using the Tencent AI Lab Embedding Corpus.

3.2. Convolutional Neural Network

The CNN has strong feature extraction capabilities. Herein, we will apply a smart "Convolution" operation to text encoding. The input is defined as a $d \times h$ matrix, where d is the dimension of the vector space for word embedding and h is the number of words. To alleviate the computational time and memory usage in CNNs, the Depthwise Separable Convolution (DSC) [9] is exploited in our proposed MRC model.

- **Depthwise separable convolution**

There are two execution steps for DSC: depthwise convolution and pointwise convolution. Given a sentence of h words, in the depthwise convolution, each word is converted to a 200-dimensional vector in the embedding step, and results in a $200 \times h$ matrix. We input this matrix via 200 channels to do convolution with the corresponding filter. In Figure 2, supposing that the filter size is $1 \times n$, 200 feature maps are then obtained after convolution.

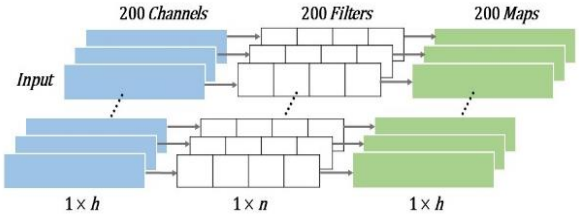

FIGURE 2. The first step of DSC: Depthwise convolution.

In the pointwise convolution, as illustrated in Figure 3, the 200 feature maps is hugely decreased to one by performing the convolution of these feature maps with 200 filters of size 1×1. Using this CNN architecture, both the amount of memory usage and the time of model training can be enormously reduced, which meets the requirement for developing our proposed model used in MRC.

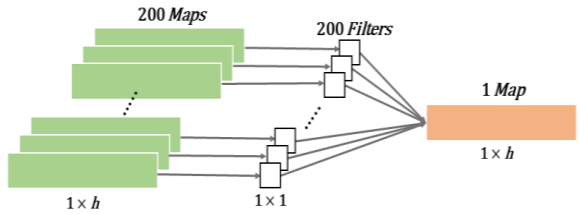

FIGURE 3. The second step of DSC: Pointwise convolution.

3.3. Attention Mechanism

There are some NLP models based on Attention Mechanism, such as QANet [5] and BERT [6]. These models have been proven to have faster training speeds and higher accuracy than some similar designs, like RNN-based models, do. To achieve such preferable performance, our proposed MRC model also adopts Attention-based designs, including Context-Query Attention and Self-Attention.

- **Context-query attention**

The Context-Query Attention was first used in the BiDAF model published by Seo et al. [3]. The main function of Context-Query Attention is to interact a given article with the information of an input question. This attention mechanism consists of two parts called Context-to-Query Attention and Query-to-Context Attention, which is accomplished by two steps. In the first step, assuming that the input article is represented by matrix C with size $d \times n$, and the input question is represented by matrix Q with size $d \times m$, we can get the element of a similarity matrix S with size $n \times m$ through the formula as follows:

$$s_{ij} = f(c_i, q_j) = w_0{}^T \cdot [q_j; c_i; q_j \odot c_i] \quad (1)$$

where c_i, q_j, w_0, $\odot$, and "[;]" stand for the ith word of the article, the jth word of the question, the trainable weight vector, the element-wise multiplication, and the operation of vectors concatenation across row, respectively. Once each element s_{ij} is calculated, we can build the similarity matrix S.

Given matrices C, Q, and S, in the second step, we acquire the matrix of Context-to-Query Attention A and the matrix of Query-to-Context Attention B by the operations depicted below:

$$A = \bar{S} \cdot Q^T \quad (2)$$

$$B = \bar{S} \cdot \bar{\bar{S}}^T \cdot C^T \quad (3)$$

with $$\bar{S} = softmax(the\ rows\ of\ S) \quad (4)$$

and $$\bar{\bar{S}} = softmax(the\ columns\ of\ S) \quad (5)$$

At the end, we set the output of the Context-Query Attention to $[c; a; c \odot a; c \odot b]$, where a and b are a row of attention matrices A and B, respectively; c is a row of article matrix C. By means of the above attention mechanism, parts of an input article to interact with input questions can be effectively strengthened. Accordingly, our MRC model can capture the answers of the questions relevant to the article more efficiently.

- **Self-attention**

Self-Attention was first proposed by Vaswani et al. [10]. Each word in the Self-Attention processing will refer to other words in the same sequence. The final result allows each word to comprise the information related to others, which possesses similar effects on the output of RNN.

Before the Self-Attention processing, we must merge the original input sequence that contains multiple vectors into one matrix, and then multiply this matrix by three trainable weight matrices W^Q, W^K, and W^V to get matrices $Q(query)$, $K(key)$, and $V(value)$, respectively. In the sequel, the input matrix of Self-Attention is expressed by the following formula:

$$Attention(Q, K, V) = softmax\left(\frac{Score}{\sqrt{d_k}}\right) \cdot V \quad (6)$$

with $Score = Q \cdot K^T$.

In the above formula, the multiplication of the matrices Q and K^T yields a matrix $Score$, which can be regarded as the correlation of each token in the input with all other tokens. The elements of $Score$ divided by a parameter $\sqrt{d_k}$ are to prevent from being too large, where d_k is the dimension of a key.

To sum up this attention mechanism, most of the current MRC models adopt a Multi-head design in which multiple sets of Q, K, and V are used. But, to further save execution time, we alternatively apply a Single-head design to our model encoder.

4. Our Machine Reading Comprehension Model

The architecture of our MRC model is graphically shown in Figure 4. In this section, we partition the model into five parts, namely Input Pre-processing Layer, Input Encoding Layer, Interaction Layer, Model Encoding Layer, and Output Layer. There are two characteristics of our model. First, in the Interaction Layer, twice the Context-Query Attention operation is employed to reduce the convergence time and achieve higher accuracy; second, in the Model Encoding Layer, the Multi-head Self-Attention design is replaced with a Single-head design to save memory usage.

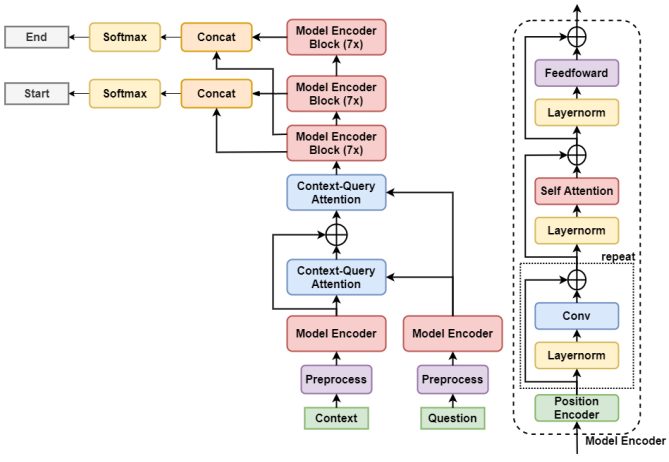

FIGURE 4. The architecture of our proposed MRC model.

4.1. Input Pre-processing Layer

In this part, we convert the original human language into the vector space that can be handled by our MRC model. Figure 5 shows the flow chart of pre-processing steps. To begin with, the processing of the original text is carried out by two kinds of tokenization, which are word-based and phrase-based. Herein, the phrase-based tokenization of Chinese is performed by the Jieba toolkit [7]. Particularly, an embedding thesaurus dictionary servers as an auxiliary tool for the Jieba toolkit to attain better match rate of phrases not only in the tokenization step but also in the embedding step.

We then perform the phrase embedding operation using Tencent AI Lab Embedding Corpus [8]. On the other hand, the word-based tokenization of Chinese is rearranged by a CNN. Finally, concatenate the sets of the embedding results from word-based vectors and phrase-based vectors into a single set. However, the numbers of word-based vectors and phrase-based vectors are different from each other. To deal with concatenation, we first make the number of word-based vectors be the same as that of phrase-based vectors. Usually, the former is larger than the latter. To unify their quantities, some zero vectors are added to the less. After this processing, we can concatenate the two sets of vectors into one set of vectors whose dimension is increased, namely d_{model}, and further sent to a series of two Highway Networks [11] such that the gradient explosion and gradient disappearing problems can be suppressed during the training process.

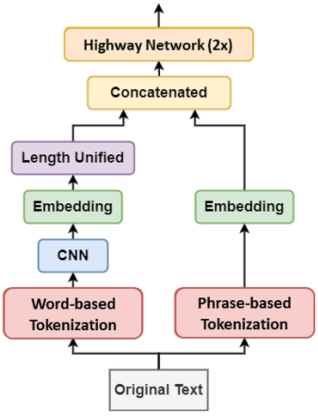

FIGURE 5. The flow chart of the pre-processing steps of our MRC model.

4.2. Input Encoding Layer

In this subsection, we will encode the output of the Input Pre-processing Layer using a model encoder whose quantifiable levels are d_{model}. Since the Self-Attention operation is unaware of the order of a word sequence, so we first have to find the word position that is encoded by the following formulas proposed by Vaswani et al. [10]:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}) \quad (7)$$

and $$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}) \quad (8)$$

Given a word position pos, through these formulas, it becomes a position vector whose nth element is $PE_{(pos,n)}$, $n = 0,1,\dots,d_{model} - 1$. After this positional encoding, each dimension of the position vector corresponds to a sinusoid. The wavelengths grow in terms of a geometric

progression, which are ranged from 2π to $10000 \cdot 2\pi$. For any fixed offset k, PE_{pos+k} can be presented as a linear function of PE_{pos}. At last, we add the calculated position vector to the original input vector, so that the original input can contain the position information.

Subsequently, perform the encoding process by Convolution, Self-Attention, and Feedforward Network. In the Convolution phase, the Depthwise Separable Convolution [9] is applied, the filter size is 7 and the number of filters is 128. The Convolution operation is repeated 4 times. In the Self-Attention phase, a Single-head design is exploited instead of Multi-head design to reduce memory consumption. In each Self-Attention operation, we use the residual method proposed by He et al. [12] to prevent data loss in neural networks comprising a large amount of layers. Besides, each phase of the encoding process is first adjusted with a layer normalization proposed by Ba et al. in 2016 [13]. After the above-mentioned phases, we can respectively encode the article and question that are ready to find their interaction in the next layer.

4.3 Interaction Layer

In this subsection, we will compute the interaction between the article and the question, and fuse the information of the question to the article. For the design of this layer, we refer to the concept of two pass interaction in Ruminating Reader developed by Gong and Bowman [4]. At the beginning, the Context-Query Attention is employed. The article before the processing is denoted as C, and the output of the Attention is denoted as T. Then merge the article and question through the following operation:

$$H = f \cdot G + (1 - f) \cdot C \tag{9}$$

$$\text{with } G = relu(W^G T) \text{ and } f = \sigma(W^f T)$$

where W^G and W^f are the trainable weights in form of matrices; both the $relu(\cdot)$ and $\sigma(\cdot)$ are two commonly used activation functions. The resulting H is treated as a new article, and the original question is putted to the Context-Query Attention again.

4.4 Model Encoding Layer

This layer encodes the output of the Interaction Layer. There are three blocks connected in series, each of which contains 7 model encoders. The model encoder used in this subsection is the same as that in the Input Encoding Layer, except that the CNN only be repeated twice here. We define the respective outputs of the three blocks as matrices M_0, M_1, and M_2 which will be adopted in the Output Layer.

4.5. Output Layer

This layer is the final step of executing our MRC model. Its function is to find the location of the answer in the encoded article. The way we find the answer is to use the boundary method, which means that the model only predicts the starting and ending positions of the answer in the article. We predict the probabilities of these two positions in the article to be p^1 and p^2, and the calculation is stated as follows:

$$p^1 = softmax(W_1[M_0|M_1]) \tag{10}$$

$$p^2 = softmax(W_2[M_0|M_2]) \tag{11}$$

where W_1 and W_2 are trainable matrices, and M_0, M_1 and M_2 are the corresponding output matrices of the three blocks of model encoders in the previous layer. Therefore, we can calculate the loss of the training result with the function defined below:

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N}\left[log(p^1_{y_i^1}) + log(p^2_{y_i^2})\right] \tag{12}$$

where y_i^1 and y_i^2 are the ground-truths of starting and ending positions of example i; $p^1_{y_i^1}$ and $p^2_{y_i^2}$ are the prediction probabilities of the starting and ending positions of example i, and N is the total number of examples.

5. Experimental Results and Discussion

In this section, we compare our MRC model with those from the leaderboard over the years in various situations. The effectiveness of our MRC model is verified in both the English and Chinese environment. In the aspect of Chinese environment, there is currently no published Chinese MRC model. For fair comparison, we will revise our MRC model to suitable for English environment. The experimental Chinese MRC system based on our proposed model is demonstrated in Appendix.

5.1. Test on Stanford Question Answering Dataset

Although our MRC model is designed for Chinese environment, it is still available under English environment. To show that, we choose the English dataset SQuAD v1.1 for evaluation. The characteristic of SQuAD v1.1 is that the answer must be a contiguous block in the article, whose position is also marked in the article.

To be applicable to the text structure of English articles, we replace the Chinese word embedding tools with the pre-trained dictionary “Glove” [14]. After training, we test our

MRC model together with some recent MRC models, and compare their resulting EM and F1 scores, as Table 2 lists.

TABLE 2. Comparison of the Accuracy of Some Recent MRC Models and Ours on Dataset SQuAD v1.1

Model	EM Score	F1 Score
Match-LSTM	64.1%	73.9%
BiDAF	67.7%	77.3%
Ruminating Reader	70.6%	79.5%
QANet	73.6%	82.7%
BERT(base)	80.8%	88.5%
BERT(large)	84.1%	90.9%
Ours	70.1%	79.4%

From these experimental results, we can observe that although our MRC model is not designed for performing on English datasets, its accuracy is higher than that of some other related work. Compared to QANet, our MRC model has a significant reduction in the constituting architecture, but its accuracy is slightly lower than that of QANet.

Another achievement accomplished by our MRC model is to shorten the convergence time that is required for reaching the F1 score higher than 77% on the SQuAD v1.1 dataset. Table 3 records the comparison of the training performance of BiDAF, QANet, and Ours.

TABLE 3. Comparison of the Training Performance of QANet, BiDAF, and Ours on Dataset SQuAD v1.1

Model	GPU	Convergence Time	Training Speed
BiDAF	TESLA P100	15 hours	37 samples/sec
QANet	TESLA P100	3 hours	259 samples/sec
Ours	Titan XP	3.5 hours	77 samples/sec

To accelerate the training process, we diminish the batch size that results in a relatively small number of samples processed per second. Even so, when the accuracy reaches the F1 score of 77%, the convergence time required for our MRC model is still almost the same as that for QANet, and is much less than that for BiDAF. As for the training speed, our MRC model is superior to BiDAF, but inferior to QANet.

5.2. Test on Delta Reading Comprehension Dataset

In Chinese environment, we propose two other architectures which are called $\text{Model}_{\text{simple}}$ and Model_{bi} .by modifying our original MRC model. In the $\text{Model}_{\text{simple}}$, the Context-Query Attention operation is only used once; while, in the Model_{bi}, we make a bi-directional design of the Self-Attention operation, which is similar to the architecture of BERT [6].

In the following experiment, we employ the cross-validation method to evaluate the aforementioned MRC models on the DRCD dataset [15]. The structure of DRCD is the same as that of SQuAD v1.1. The performance comparison of $\text{Model}_{\text{simple}}$, Model_{bi}, and our original MRC model is shown in Table 4.

TABLE 4. Comparison of the Accuracy of $\text{Model}_{\text{simple}}$, Model_{bi}, and Ours on Dataset DRCD

Model	EM Score	F1 Score
$\text{Model}_{\text{simple}}$	61.8%	75.6%
Model_{bi}	64.2%	78.3%
Ours	64.9%	79.0%

From the above results, it can be seen that our original MRC model using Context-Query Attention twice is more effective than both the $\text{Model}_{\text{simple}}$ and Model_{bi} do. Next, we compare the training performance in Chinese environment. In this experiment, all the three models are trained by the aid of NVIDIA Titan XP GPU, and the comparison result is listed in Table 5.

TABLE 5. Comparison of the Training Performance of $\text{Model}_{\text{simple}}$, Model_{bi}, and Ours on Dataset DRCD

Model	GPU	Convergence Time	Training Speed
$\text{Model}_{\text{simple}}$	Titan XP	1 hour 58 minutes	72.3 samples/sec
Model_{bi}	Titan XP	1 hours 34 minutes	56.7 samples/sec
Ours	Titan XP	43 minutes	80.6 samples/sec

As the performance shown in this table, our original MRC model requires less convergence time as well as has more training speed than the other two models do. The overall experimental results manifest that the two designs originated from modifying our model are not ineffective. The convergence time of our model is less than 1 hour under the computer equipped with NIVIDIA Titan XP GPU. As a result, we reach the goal of moderating memory usage and shortening the convergence time as the model is expected.

6. Conclusions

In this paper, we present a Chinese MRC model achieved by an economical computational cost, which possesses two main contributions. First, based on the design adopted by BERT, the second Context-Query Attention is added to enhance the relation between articles and questions. Second, to save memory consumption, we use the Single-head Self-Attention instead of a Multi-head design in the training process. In the experiment, the DRCD dataset is chosen to evaluate our model in Chinese environment, which arrives at the accuracy of 79.0% for EM and 64.9% for F1. Besides this, the convergence time and training speed are 43 minutes and 80.6 sample/sec, respectively. Compared to the simplified model using Context-Query Attention once and the redesigned model using bi-directional Self-Attention, our original MRC model has higher accuracy, less convergence time, and faster training speed, which is useful for the environment possessed of limited computational capability.

Acknowledgement

The authors thank the Ministry of Science and Technology of Taiwan (R. O. C.) for supporting this work in part under Grant MOST 107-2221-E-011-113-MY2.

Appendix

In this appendix, we show the screenshots of our experimental Chinese MRC system. As seen from Figure A.1, the interface of the system contains three regions which are the input article, user input question, and system output answer.

FIGURE A.1 Two examples of demonstrating our experimental Chinese MRC system: (a) Question: Where is the capital of Germany? Answer: Berlin; (b) Question: What is the figure of my father? Answer: A fatty.

References

[1] P. Rajpurkar et al., "SQuAD: 100,000+ questions for machine comprehension of text," arXiv:1606.05250 [cs.CL], Oct. 2016.

[2] S. Wang and J. Jiang, "Machine comprehension using Match-LSTM and answer pointer," arXiv:1608.07905 [cs.CL], Nov. 2016.

[3] M. Seo et al., "Bidirectional attention flow for machine comprehension," arXiv:1611.01603 [cs.CL], Jun. 2018.

[4] Y. Gong and S. R. Bowman. "Ruminating reader: Reasoning with gated multi-hop attention," arXiv:1704.07415 [cs.CL], Apr. 2017.

[5] A. W. Yu et al., "QANet: Combining local convolution with global self-attention for reading comprehension," arXiv:1804.09541 [cs.CL], Apr. 2018.

[6] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," arXiv:1810.04805 [cs.CL], May 2019.

[7] J. Sun, *Jieba Chinese Word Segmentation Tool*, Jun. 2012. Accessed on: Feb. 4, 2020. [Online]

[8] Y. Song et al., "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proceedings of the International Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 175-180, New Orleans, Louisiana, Jun. 2018.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1800-1807, Honolulu, Hawaii, Jul. 2017.

[10] A. Vaswani et al., "Attention is all you need," in *Proceedings of the Annual Conference on Neural Information Processing Systems*, pp. 5998-6008, Long Beach, California, Dec. 2017.

[11] J. G. Zilly et al., "Recurrent highway networks," in *Proceedings of the International Conference on Machine Learning*, pp. 4189-4198, Sydney, Australia, Jul. 2017.

[12] K. He et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, Las Vegas, Nevada, Jun. 2016.

[13] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," arXiv:1607.06450 [stat.ML], Jul. 2016.

[14] J. Pennington, R. Socher, and C. D. Manning. "Glove: Global vectors for word representation," in *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, pp. 1532-1543, Doha, Qatar, Oct. 2014.

[15] C. C. Shao et al., "DRCD: A Chinese machine reading comprehension dataset," arXiv:1806.00920 [cs,CL], May 2019.