Machine Reading Comprehension Based On Multi-headed attention Model

Hui Xu[1], Shichang Zhang[2] and Jie Jiang[1, a]

[1]*College of Systems Engineering, National University of Defense Technology,410073 Changsha, China*
[2]*School of Information Science and Engineering, Ocean University of China, Songling Road No. 238,266100 Qingdao, China*

Abstract. Machine Reading Comprehension (MRC) refers to the task that aims to read the context through the machine and answer the question about the original text, which needs to be modeled in the interaction between the context and the question. Recently, attention mechanisms in deep learning have been successfully extended to MRC tasks. In general, the attention-based approach is to focus attention on a small part of the context and to generalize it using a fixed-size vector. This paper introduces a network of attention from coarse to fine, which is a multi-stage hierarchical process. Firstly, the context and questions are encoded by bi-directional LSTM RNN; Then, more accurate interaction information is obtained after multiple iterations of the attention mechanism; Finally, a cursor-based approach is used to predicts the answer at the beginning and end of the original text. Experimental evaluation of shows that the BiDMF (Bi-Directional Multi-Attention Flow) model designed in this paper achieved 34.1% BLUE4 value and 39.5% Rouge-L value on the test set.

1 Introduction

The general form of reading comprehension is that the tester answers the relevant questions of the article by reading an article and understanding the meaning of the article. With the development of artificial intelligence (AI), using a machine for reading comprehension tasks has become a research hotspot. In the past few years, the development of machine reading comprehension has made great strides in the field of natural language. As computing power increases, this method which constructs complex machine reading comprehension models based on deep learning is now the mainstream method. At the same time, the introduction of the attention mechanism enables the model to focus on the target areas related to the problem in the context paragraphs, so that the deep learning model has been significantly improved [1].

From the research in recent years, currently, the attention mechanism of the machine reading comprehension model is single-pass, which is more common, based on deep learning. Therefore, this article is subject to the habit of repeated reading by humans when constructs the model. Multiple iterations of attention mechanism are introduced to simulate the habit of repeated reading by humans in the input layer and the attention layer of the network model so that the network has better learning ability. Experiments show that the model can better understand context semantics.

2 Related work

The availability of the task of machine reading comprehension data sets is driving the development of machine reading comprehension in recent years. Early data sets included MCTest [2], Children's Book Test [3] and so on.

Recently, Baidu released the DuReader datasets [4]. Compared with the previous dataset, the problem of DuReader comes from Baidu search and Baidu Knows of different domains, which are manual generated and full of challenges. This paper evaluates the performance of the model on the DuReader dataset.

In 2015, Hermann et al. first introduced attention mechanisms into the tasks of machine reading comprehension. It has been found that the attention mechanism can make the model study more efficient, so the attention mechanism is promoted in the task of machine reading comprehension task [5]. In 2016, Kadlec et al. introduced the pointer network into the machine reading comprehension task [6]. Trischler et al. solve the problems that need filling by combining the attention model with the ranking model [7]. Chen et al. discovered that using simple bilinear terms to calculate the attention vector in the same model could improve the accuracy tremendously [8]. Cui et al. proposed a bidirectional attention mechanism to encode contexts and problems [9]. Wang and Jiang et al. generated the answer boundary by using an approach that combines Match-LSTM with a pointer network [10]. Yu and Lee et al. solve the machine reading comprehension task by sorting rang of continuous text [11]. Xiong et al. proposed a dynamic pointer network to infer answers through an iterative approach [12]. Yang et al. proposed a fine-grained gating mechanism to dynamically combine word-level and character-level

[a] Corresponding author: jiejiang@nudt.edu.cn

representations and model the interaction between the question and the paragraph [13].

In addition, people have also studied the secret of the encoding of context words. Cheng et al. proposed a new type of LSTM network to encode words in sentences so that the model learns information about the relationship between the tag currently being processed and the tag in the memory [14].

Bi-Directional Attention Flow (BiDAF) is a deep learning model for machine reading comprehension proposed by Minjoon Seo et al. [15] Compare with previous work, BiDAF's biggest improvement is the introduction of a bidirectional attention mechanism in the Interaction layer. That is to say, firstly, our model calculates a similarity matrix of the original text and the problem; then we calculate the two attentions of Query2Context and Context2Query based on the matrix, and calculate the original representation of query-aware based on attention, and then use the bidirectional LSTM to aggregate the semantic information. In addition, the Embed layer is mixed with word-level Embedding and character-level embedding, the word-level embedding is initialized using the pre-trained word vector, and the character-level embedding is further encoded using CNN. The two embeddings are input through the 2-layer Highway Network for the coding layer [16]. Finally, BiDAF uses a boundary model to predict the location of the answer to the beginning and end to get the answer to the question.

Compared with the above model, the model designed in this paper introduces a self-attention mechanism when coding, so that the model can better learn the information contained in the sentence. At the same time, the introduction of an additional attention mechanism can reduce the loss of information, and it can make the generated answers more accurate.

3 Model

3.1 Network architecture

The overall architecture of the BIDMF model is shown in Figure 1. It is a layered multi-stage model, which is mainly divided into the following six layers:

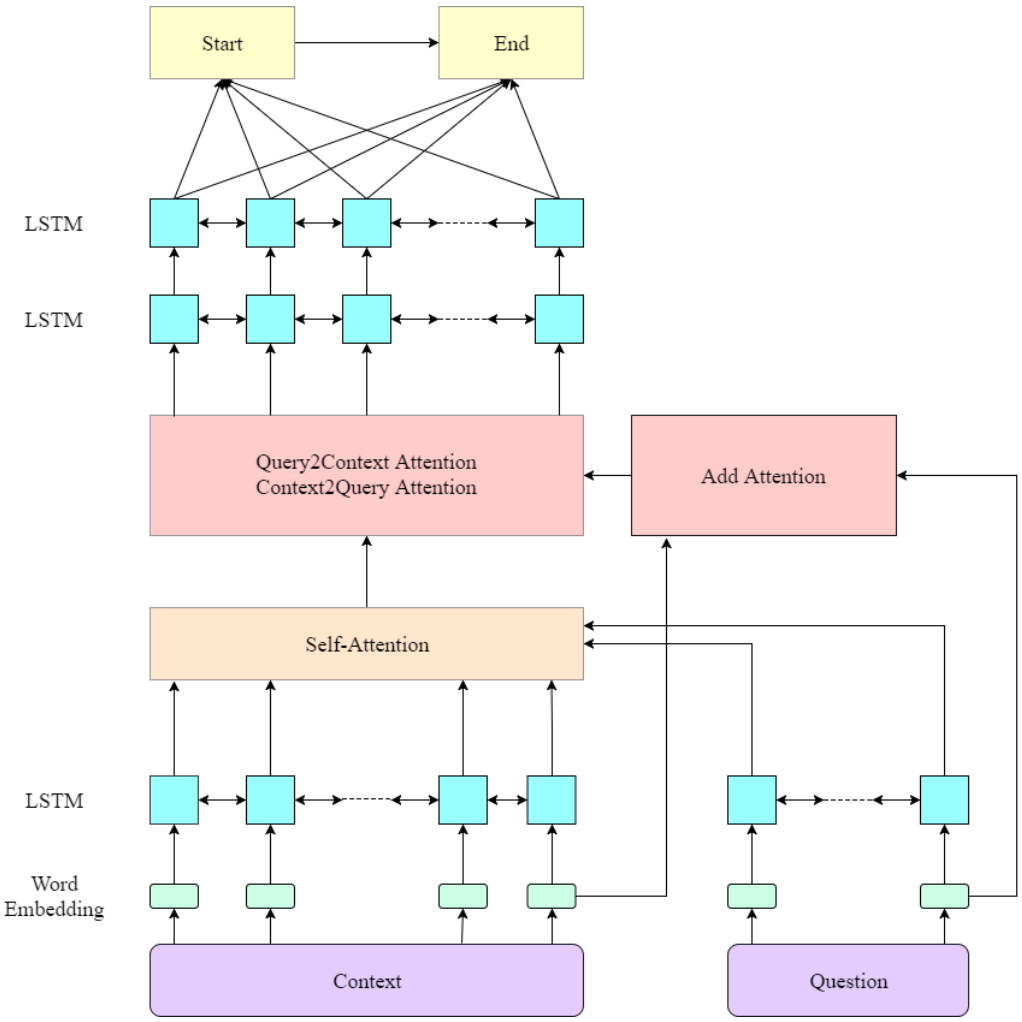

Figure 1 Network model

(1) **Word Embedding Layer:** It maps each word to a vector space using a pre-trained word embedding model.

(2) **Contextual Embedding Layer:** It inputs the word

vector generated by the Word Embedding layer into a layer of LSTM, and initially learns the information contained in the context.

(3) **Self-attention Layer:** It inputs the output of the Contextual Embedding Layer, and the model can more accurately learn the information contained in the context.

(4) **Attention Flow Layer:** Its role is to couple the query and context vectors and generates a set of query function feature vectors for each word in the context.

(5) **Modeling Layer:** It uses the recurrent neural network (RNN) to read the context information after learning.

(6) **Output Layer:** It outputs the answer to the question.

3.2 Algorithm implementation

1. Word Embedding Layer: It maps each word to a high-dimensional vector space by pre-training the word vector to obtain a fixed embedding of each word.

2. Contextual Embedding Layer: The machine reading comprehension model usually uses the Long Short-Term Memory network (LSTM) to simulate the interaction between words in a sentence. Therefore, this paper uses a bidirectional LSTM to capture the local relationship between the context **X** and the problem **Q** respective words, splicing the bidirectional LSTM output, and obtaining the $\boldsymbol{H} \in \boldsymbol{R^{2d*T}}$ context coding vector and $\boldsymbol{U} \in \boldsymbol{R^{2d*T}}$ problem coding vector. It is worth noting that each column vector dimension of H and U is two-dimensional because the bidirectional LSTM forward and backward output dimensions are one-dimensional.

3. Self-attention Layer: This layer introduces two kinds of attention mechanisms, one is scaled Dot-product attention and, the other is Multi-headed attention [17].

In fact, scaled Dot-product attention is to use the dot product to calculate the similarity, but only one more dimension to adjust so that the inner product is not too large. The formula for the concentration of attention is as follows:

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

Self-attention is to input a sentence so that each word in it must be a process of attention calculation with all the words in the sentence. The purpose is to let the model learn the dependencies inside the sentence and capture the internal structure of the sentence. The expression on the formula is to make the input satisfy Q=K=V.

Multi-headed attention refers to splicing the results of multiple scaled Dot-product attention. Firstly, we need to make a linear transformation on Q, K, V, and then input it into scaled Dot-product attention. The calculation of the scaled Dot-product attention mechanism does multiple times (number of times i) calculation that is called multi-head, and each time a head is counted, but the parameter W of linear transformation every time Q, K, V is different; Then splicing the results of the calculation i times; Finally, a linear transformation is performed to take the resulting value as a result of the self-attention mechanism layer. Performing multiple attention calculations allows the model to learn relevant information in different representation subspaces, which is also consistent with the habit of repeating multiple times when reading articles. The formula for Multi-headed attention is as follows:

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^v\right) \qquad (2)$$

$$MultiHead(Q,K,V) = Concat(head_i,..,head_h)W^O \qquad (3)$$

4. Attention Flow Layer: This layer is responsible for linking and synthesizing the information of context and questions, which is different than previous attention mechanisms. The attention mechanism used in this paper no longer summarizes the problem and context as a single feature vector, ut allows the attention vector and the previous Embedding layer of each time step to be input into the subsequent modeling layer. This reduces the loss of information during the delivery process.

The input to this layer is the vector representation H of the context and the vector representation U of the questions. The output of this layer is the fusion of context and questions information and the output of the previous layer. The similarity between the context and the questions is represented by $S \epsilon R^{T \times J}$, where S_{tj} represents the similarity between the *t*-th words of the context and the words of the *j*-th question. The similarity matrix is calculated by:

$$S_{tj} = \alpha\left(H_{:t}, U_{:j}\right) \in R \qquad (4)$$

Where α is a trainable scalar function that measures the similarity between two input vectors,$H_{:t}$ is the t-th column vector of H, $U_{:j}$ is the j-th column vector of U. Let $\alpha(h,u) = w_{(S)}^T[h;u;h^\circ u]$, Where $w_{(s)} \in R^{6d}$ is a trainable variable, °is element multiplication, and [;] is a vector connection between lines.

This layer involves three different forms of attention mechanisms. Next, the implementation details of the attention mechanism are introduced separately:

Context-to-query Attention: This attention is used to calculate which the word in the question is most relevant to the words of the context. Let $a_t \in R^J$ denote the attention weight of the words in the t-th context and the words in the question. For all t$\sum a_{ij} = 1$. Attention weight $a_t = softmax(s_{t:}) \in R^J$, the input problem vector $\widetilde{U}_{:t} = \sum_j a_{tj} U_{:j}$. When a_t is obtained, the output multiplied by the vector representing the context is then calculated by the formula (1)(2)(3) to obtain the final attention weight.

Query-to-context Attention: The word used to calculate which word in context is most similar to the word in question. Attention weight b = softmax$(max_{col}(S)) \in R^T$ Then use $\tilde{h} = \sum_t b_t H_{:t} \in R^{2d}$ to represent the weighted sum of the most important words in the context, and then calculate the attention weight by the formula (1)(2)(3).

Add Attention: With the idea of residuals, the model pays extra attention. The embedded word vector is directly input into the formula (2)(3), except that Q is the vector of context word, and K and V are vectors of questions the word.

Finally, we combine the context and attention vectors to get a vector G, where each column vector of G can be thought of like the attention distribution of the problem for each context word. The formula we define G is as follows:

$$G_{:t} = \beta\left(H_{:t}, \widetilde{U}_{:t}, \widetilde{H}_{:t}\right) \qquad (5)$$

Where $G_{:t}$ is the t-th column vector (corresponding to t-th the words in the context), and β is a vector that can be trained to fuse different inputs. In this paper, β is simply splicing the input to get a new Vector.

5. Modeling Layer: The input of the modeling layer is the vector G, which encodes the output obtained earlier. The purpose of this layer is to learn the interaction between contextual words that are conditional on the problem. Unlike the previous context embedding layer, the context embedding layer learns the interaction between context-independent words, but the modeling layer learns the interaction between context and problem words. The modeling layer model also uses bidirectional LSTM, and the final model obtains the matrix $M \epsilon R^{2d \times T}$ to predict the answer.

6. Output Layer: The model obtains the final result by predicting where the answer begins and ends in the context. The probability distribution of the answer start word in the context is

$$p^1 = softmax\left(w_{(p^1)}^T [G, M]\right) \quad (6)$$

Where $w_{(p^1)} \in R^{10d}$ is a weight that can be trained. Similarly, the probability distribution of the end of the answer is

$$p^2 = sofmax\left(w_{(P^2)}^T [G, M]\right) \quad (7)$$

7. Training: The loss function of this paper is defined as formula (8), and the loss function is minimized.

$$\mathrm{L}(\theta) = -\frac{1}{N}\sum_i^N log\left(p_{y_i^1}^1\right) + log\left(p_{y_i^2}^2\right) \quad (8)$$

Where θ is the set of trainable weights, N is the number of samples, and y_i^1 and y_i^2 are the indices of the start and end of the answer to the i-th example.

4 Experiment

This article uses the recently released Dureader dataset to evaluate the model. Dureader is a machine reading dataset on Baidu Q&A and Baidu search, containing more than 100,000 questions.

4.1 Dataset

Each sample in the DuReader dataset is a sequence of 4-tuples: {q, t, D, A}, where q is the problem, t is the type of question, D is the context in question, and A is the set of answers. DuReader divides the problem by two dimensions. First, the problem is divided into the entity class problem, the description class problem, and the right and wrong class problem. For the entity class question, the general form of the answer is a single definite answer. For example, 'when is the Huawei 10 released?'; The answer to the description class problem is generally long. It is a summary of multiple sentences, such as the typical how/why type question. 'Why is the fire truck red?'; For the right and wrong class problem, the answer is often simpler, yes or no, for example: 'Is it raining today?'; Second, the problem is divided into fact classes and opinion classes. The DuReader dataset comes from Baidu search, and Baidu Knows.

Table 1 Example of Dureader

Question	What will happen if we eat too much vitamin b2?
Question Type	*Entity-Fact*
Answer 1	Water-soluble vitamins such as vitamin B are easily excreted in the urine and cannot be accumulated in the body, so it is difficult to cause poisoning unless you eat too much (for example, 100 times the normal amount).
Document 1	Vitamin is the composition of a variety of necessary coenzymes of the body's various chemical reactions and metabolic processes. It cannot be synthesized in the body and provided from the food supply. Natural vitamin supplementation in the food and drink is good for the human body. ...
...	
Question	Must wisdom teeth unplug?
Question Type	*YesNo-Opinion*
Answer 1	[Yes] Because wisdom teeth are difficult to clean, they are more prone to oral problems than normal teeth, so doctors will suggest removing.
Answer 2	[Depend] Wisdom teeth do not necessarily have to be unplugged. We usually only unplug symptomatic wisdom teeth, such as which often cause inflammation.
Document 1	Why we remove wisdom teeth? My wisdom teeth are healthy, why do doctors want me to pull out? Mainly because wisdom teeth are hard to clean...
...	
Document 5	According to my clinical experience of many years, wisdom teeth do not have to be pulled out. There are many kinds of wisdom teeth impactions.

4.2 Model initialization

At the beginning of the training, the training data is pre-processed, a dictionary of data sets is generated, and a word with a size of 300 is randomly initialized, and the hidden of Bi-LSTM is set to 150. The network uses the Adam algorithm to train the model with an initial learning rate of 0.001 and a batch size of 32.

4.3 Experimental results

Model performance is determined by two indicators, BLUE and ROUGE. BLUE is essentially a calculation factor of the frequency of the co-occurrence words in a sentence. ROUGE is a similarity measure based on recall rate. Compared with BLUE, it is similar. There is no Fmeans evaluation function, which mainly investigates the sufficiency of sentences and cannot evaluate the fluency of sentences. It calculates the collinear probability of the N-gram prediction answer and the labeling answer. Roug-L is based on the longest shared clause co-occurrence accuracy and recall rate Fmeasure statistics.

We compare the results of the model test with the baseline provided by Dureader, which are shown in Table 2. The assessment of the model BLUE4 in this paper is 34.76%, and the Rouge-L% score is 39.5%, which is improved compared with other methods.

Table 2 Experimental result

Model	BLUE4%	Rouge-L%
Selected Paragraph	16.4	30.2
Match-LSTM	31,9	39.2
BIDAF	31.8	39.0
Our Models	**34.76**	**39.6**

4.4 Comparative experiment

From the change of the attention weight curve in Figure. 2, it can be seen that when the model is added to the self-attention layer, the weights of the core words such as "material," "bottle" and "best" is increased. Therefore, the self-attention network model can better assign attention weights, increase core word weights, and reduce non-core word weights.

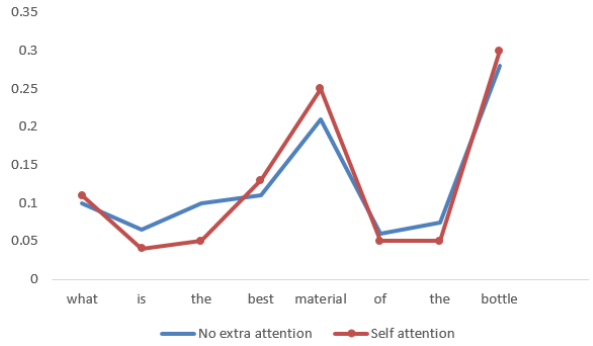

Figure 2 Attention weight change

Table 3 is about the network with or not Add Attention answer prediction results. From the table, the question is “How to make delicious salted fish.” The correct answer should be about how to make delicious salted fish. But the model without Add Attention answers how to make salted fish. Therefore, by comparison, Add Attention can make the model better learn the connection between the article and the problem to get the correct answer.

Table 3 Prediction results

Question	How do you make dry salted fish good to eat?
No extra attention	1. Put the clean fish into a clean container. Wipe the fish well with salt (5 times the usual cooking), cooking wine (can be added some more), ginger powder, and aniseed (a little). 2. Compacted the fish, the fish can be pressed of some weight above and pickled for four or five days. 4. Take out and hanging on the balcony. It’s best to bask in the sun for a few days (this link is to increase the aroma of salted fish), the longer the basking, the better the taste.
Attention	1. One dried salted fish. 2. Wash and cut into small pieces. 3. Put them into boiling water, so that salty taste can also be removed. 4. Small pieces of fish have then washed once again in cold water, put into the plate for use. 5. Stir-fry the frying pan with oil, add the pepper, aniseed, shredded dried pepper, scallion, ginger, and garlic, flatten out the fragrance and pour the fish pieces and stir-fry. 6. Add the cooking wine to remove the smell, use the big fire to boil, then use small fire slowly simmer for 1 hours, the longer the time the aroma the fish is. 7. No more salt, add monosodium glutamate before out of the pot, a dish of spicy and fragrant fish is completed. Because the dried fish tastes chewy, eating with rice is very fragrant.

5 Conclusion

The BiDMF designed in this paper is a multi-stage hierarchical model, to make the model better understand the meaning of the context and reduce the training process of the information of the loss. This paper proposes to let the network learn context information better by introducing a self-attention mechanism in the model. At the same time, an additional attention mechanism is added to the bidirectional attention mechanism to obtain additional information to reduce the information loss during the learning process. The experimental results show that the BiDMF model has better reasoning ability. At the same time, the circular attention machine is beneficial to the model to understand the context information, and also meets people's reading habits.

At present, machine reading comprehension is still in a shallow understanding, and the model of this generally only extracts the words contained in the context to predict the answer. To solve this problem, in the future, the work will incorporate the reasoning mechanism into the model, so that the model has certain reasoning ability and can generate its answers.

References

1. Weston, Jason, S. Chopra, and A. Bordes. "Memory Networks." *Eprint Arxiv* (2014).
2. M. Richardson, C. J. Burges, and E. Renshaw. MC Test: A Challenge Dataset for the Open Domain Machine Comprehension of Text. *In EMNLP,* (2013).
3. Hill, Felix, et al. "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations." *Computer Science* (2015).
4. He, Wei, et al. "DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications." (2017).
5. Hermann, Karl Moritz, et al. "Teaching machines to read and comprehend." 1693-1701.(2015).
6. Kadlec, Rudolf, et al. "Text Understanding with the Attention Sum Reader Network." 908-918.(2016).
7. Trischler, Adam, et al. "Natural Language Comprehension with the EpiReader." (2016).
8. Chen, Danqi, J. Bolton, and C. D. Manning. "A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task." (2016).
9. Cui, Yiming, et al. "Attention-over-Attention Neural Networks for Reading Comprehension." (2016).
10. Wang, Shuohang, and J. Jiang. "Machine Comprehension Using Match-LSTM and Answer Pointer." (2016).
11. Yin, Wenpeng, S. Ebert, and H. Schütze. "Attention-Based Convolutional Neural Network for Machine Comprehension." (2016).
12. Xiong, Caiming, S. Merity, and R. Socher. "Dynamic Memory Networks for Visual and Textual Question Answering." (2016).
13. Yang, Zhilin, et al. "Words or Characters? Fine-grained Gating for Reading Comprehension." (2017).
14. Cheng, Jianpeng, L. Dong, and M. Lapata. "Long Short-Term Memory-Networks for Machine Reading." (2016).
15. Seo, Minjoon, et al. "Bidirectional Attention Flow for Machine Comprehension." (2016).
16. He, Kaiming, et al. "Deep Residual Learning for Image Recognition." 770-778.(2015).
17. Vaswani, Ashish, et al. "Attention Is All You Need." (2017).