IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 8, 2020, accepted May 4, 2020, date of publication May 11, 2020, date of current version May 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993875

Multi-Layer Transformer Aggregation Encoder for Answer Generation

SHENGJIE SHANG, JIN LIU, (Member, IEEE), AND YIHE YANG
College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China
Corresponding author: Jin Liu (jinliu@shmtu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61872231 and Grant 61701297.

ABSTRACT Answer generation is one of the most important tasks in natural language processing, and deep learning-based methods have shown their strength over traditional machine learning based methods. However, most previous deep learning-based answer generation models were built on traditional recurrent neural networks or convolutional neural networks. The former model cannot well exploit contextual correlation preserved in paragraphs due to their inherent computation complexity. For the latter, since the size of the convolutional kernel is fixed, the model cannot extract complete semantic information features. In order to alleviate this problem, based on multi-layer Transformer aggregation coder, we propose an end-to-end answer generation model (AG-MTA). AG-MTA consists of a multi-layer attention Transformer unit and a multi-layer attention Transformer aggregation encoder (MTA). It can focus on information representation at different positions and aggregate nodes at same layer to combine the context information. Thereby, it fuses semantic information from base layer to top layer, enhancing the information representation of the encoder. Furthermore, based on trigonometric function, a novel position encoding method is also proposed. Experiments are conducted on public datasets SQuAD. AG-MTA reaches the state-of-the-art performance, EM score achieves 71.1 and F1 score achieves 80.3.

INDEX TERMS Question answering system, natural language processing, self-attention mechanism, transformer coding structure.

I. INTRODUCTION

Question answering(Q&A) system is built on the basis of understanding of the questions. It generates answers by searching existing knowledge bases such as knowledge graph, databases, or even internet, making knowledge acquirement more direct, efficient, and accurate. With the continuous development of question answering system, many novel methods have been developed. The most notable work is the Match-LSTM [1] framework. Then, the QANet [2] improves the speed and accuracy of answer generation by combining Convolutional Neural Network (CNN) and LSTM, and achieved reliable results on the SQuAD [3] dataset. Recent method [4] combines the CNN network and attention mechanism for Chinese question classification, which boosts the effect of the answer generation.

Most of the current research work is based on typical neural networks to deal with tasks such as intent classification and answer generation. However, these methods have disadvantage in utilizing contextual correlation. In this paper, in order to enhancing the relevance of contextual information, we propose a novel multi-layer attention Transformer aggregation encoder (MTA), and a novel answer generation network based on MTA encoder (AG-MTA). The main contributions of this paper are as follows:

1. A multi-layer attention Transformer aggregation encoder (MTA) is proposed to utilize contextual information at different layers to model the sequences.

2. Multi-layer attention and feedforward layer are designed to pay attention to different subspaces' information based on the Transformer unit structure.

3. A novel position encoding method that make use of the absolute position and relative position information by encoding the position of each word.

4. Multi-layer attention transformer units are proposed to enhance the context representation and solves the problem of information loss.

The related works are discussed in section 2. Section 3 presents AG-MTA model. Section 4 presents evaluation of AG-MTA based answer generation system and discussion of

The associate editor coordinating the review of this manuscript and approving it for publication was Arianna DUlizia.

the experiment results. Finally, we draw some conclusion in section 5.

II. RELATED WORK

Along with recent advancement in question answering method, much progress has been made on answer generation. Yu *et al.* [5] proposed a method which matches questions and answers by considering semantic coding of problems. At the same time, the application of LSTM has made much progress in the Q&A system. Tan *et al.* [6] enhance the composite representation of the model by connecting the LSTM network with the convolutional neural network. Liu *et al.* [7] proposed a method that applied dynamic LSTM networks to solve the problem of long-range dependence of RNN. Lende and Raghuwanshi [8] proposed a closed domain Q&A system for processing documents about the education acts, and improves the accuracy of retrieval answers by using NLP techniques. In particular, the remarkable improvement for reading comprehension in the long text has also led to the improvement of answer generation methods. Relying on efficient neural network models, these methods perform well in the answer generation task.

Wang and Nyberg [9] proposed a method to solve the answer selection problem. This method mainly uses bidirectional Long-Short Term Memory network, without any external knowledge resources. However, this model requires long-time training and may result in loss of information. Wang and Jiang [1] proposed a network structure called MATCH-LSTM, which is mainly used to answer the question that need to find continuous words in the article. However, this method is difficult to predict longer answers.

Recently, attention mechanism has also been introduced to answer generation. Seo *et al.* [10] proposed a complex network model based on Bi-Directional Attention Flow (Bi-DAF). The model contains the Query2Context module, similar to Context2Query, which can perform attention calculation on query by context information. Dhingra *et al.* [11] proposed a new attention model Gate-Attention Reader, which utilized attention mechanisms to connect query and paragraph information, thereby enhancing the information representation of each dimension in word embedding. Vaswani *et al.* [12] proposed a new self-attention encoder and decoder model, replacing LSTM and CNN models. The experiment results prove the effectiveness of the method which can provide new ideas and solutions for NLP field.

In addition, other studies also proposed different machine learning methods and different answer generation architectures. Aiming at the problem of gradient explosion when neural network updating a larger number of word vectors, Liu *et al.* [13] proposed an algorithm for accelerating neural network parameter convergence based on stochastic conjugate gradients. Wang *et al.* [14] proposed an end-to-end model called R^3 that uses the reinforcement learning framework to combine phrase sorting method and answer generation module, while traditional approach sorts the document first and then generate the answer. Yang *et al.* [15] uses semi-supervised learning method to generate questions based on the unlabeled text. This method not only increases the amount of training data but also achieves satisfactory results. Ghaeini *et al.* [16] designed a question answering network based on the gated. To improve the accuracy of the answer, this method established the interdependence between documents and queries. In order to make full use of various types of knowledge, Zhong *et al.* [17] proposed a graph algorithm to enhance the accuracy of the question answering system.

As presented above, the existing encoder models use only the top layer output information of the network, losing information available in other layers. Related research shows that different network layers can capture different levels of semantic information in the sequence. Therefore, it is necessary to add some useful sequence information of the base layer into the coding result [18]–[20].

On the other hand, some methods use convolutional neural network models or simple attention mechanisms to extract text information, which can significantly shorten the training time of the model, and the performance of these models is roughly the same as that of the RNN network. Bell and Penchas [21] proposed a method that can capture the local dependencies well, it replaced the RNN in the reading comprehension model with fully convolutional network. Zhou *et al.* [22] proposed a method to capture both semantic information and semantic correlations between questions and answers. In addition, Tay *et al.* [23] designed multi-cast attention networks to improve the training performance, which can be used in many tasks in the Q&A field. Dong *et al.* [24] proposed the multi-column convolutional neural networks, which can extract features between questions and answers at different layers and captures information well. He and Golub [25] show that the character-level encoder-decoder framework can be applied to the Q&A system.

In summary, most of the above methods use LSTM and CNN networks to generate answers directly, and fail to exploit the context information and the relations existing among the whole article and queries. To solve this problem, the AG-MTA model is proposed, which combines the context information with the different levels of semantic information.

III. METHOD

A. PROBLEM DEFINITION

For the answer generation task in the Q&A system, we describe the formal problem definition as follows. A context material paragraph can be defined as CTX:

$$CTX = \{ctx_1, ctx_2, \ldots, ctx_n\} \tag{1}$$

where n is the number of words in the paragraph; and we define the question as:

$$Q = \{q_1, q_2, \ldots, q_m\} \tag{2}$$

where m is the number of words in the question. The model outputs a sub-sequence S from paragraph CTX according to the question Q, the sequence S is the sequence of answers

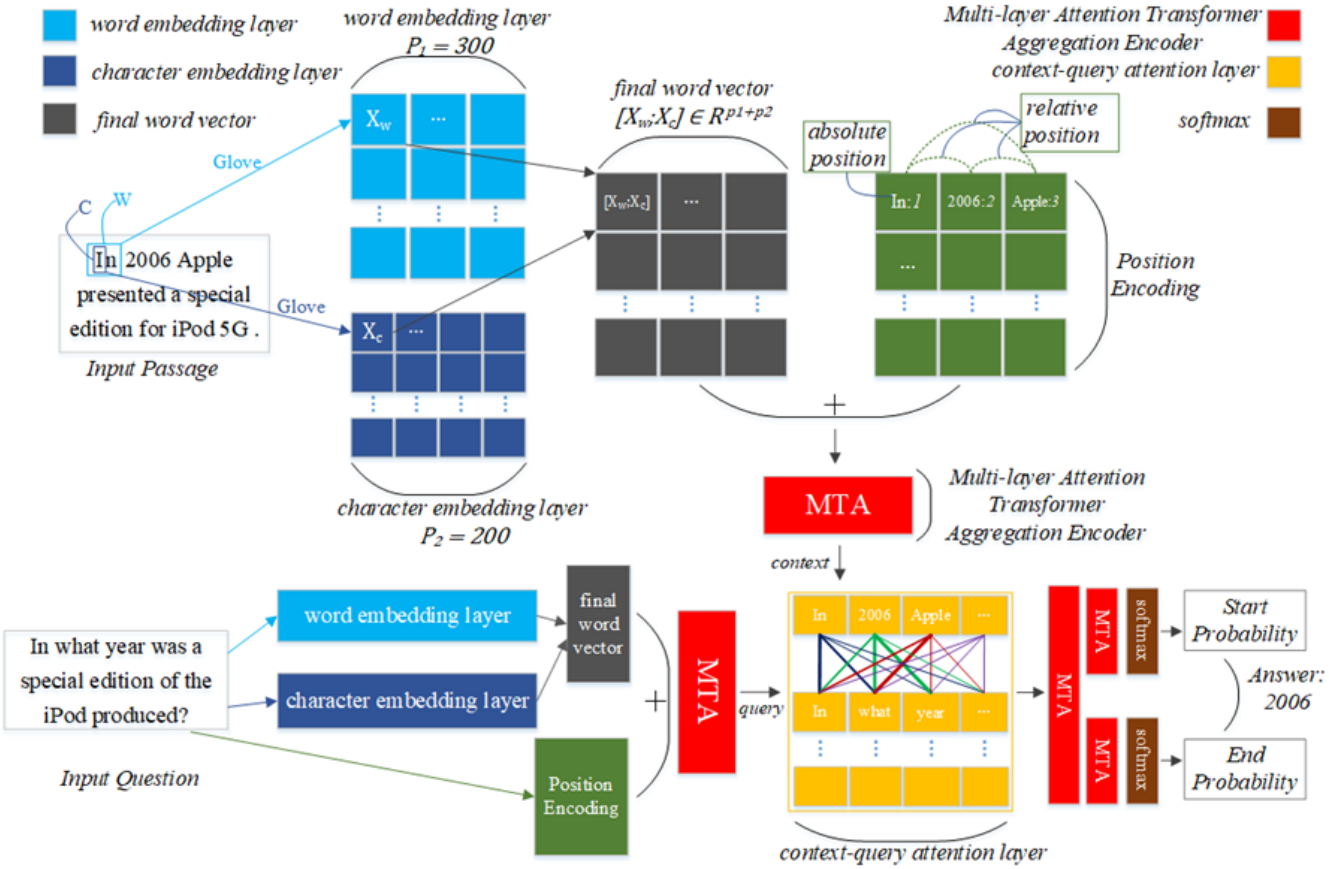

FIGURE 1. AG-MTA architecture. In the first part, we encode input passage and input questions respectively through the character embedding layer and word embedding layer. Where the dimensions of the character embedding layer is set to $p_2 = 200$ and the word embedding layer with a number of dimension $p_1 = 300$. Then we add the positional encoding, so that the attention mechanism can take into account the positional order information. In the second part, the model obtains the abstract semantic information by MTA and then learn the connection between context and query through the context-query attention layer. The model sends the information sequence into the coding layer which consists of three layers of MTAs to learn semantic information from the base layer to the top layer. Finally, the model obtains the start and end positions of the answer in the text through the softmax function.

generated by the model based on the question and the paragraph. So we define the answer S as:

$$S = \{ctx_i, ctx_{i+1}, \ldots, ctx_{i+k}\} \tag{3}$$

where i, k represent the starting position and ending position of the answer in the paragraph. In Fig. 2, we describe the process of answer generation.

B. ANSWER GENERATION NETWORK BASED ON MTA

The architecture of AG-MTA is shown in Fig. 1. It mainly consists of positional encoding, embedding layer, multiple transformers aggregation encoder, and context-query attention modules.

As shown in Fig. 1, The first part is to convert article information into a corresponding relationship matrix through the character embedding layer and the word embedding layer. The word embedding layer uses a pre-trained Glove [26] word vector with number of dimension p_1, and the dimensions of the character embedding layer is set to p_2. The word vector corresponding to a word w is x_w, and each character vector is recorded as x_c. Then we randomly initialize the character vector x_c and add it to the model.

In the meantime, each word can be seen as a connection to each character vector. We fixed the length of each word to a constant j. Thus, the word w can also be represented as a matrix of $p_2 * j$, which is the combination of the character vectors. Therefore, the final word vector $[x_w; x_c] \in R^{p_1+p_2}$ for the word w can be obtained by concatenating x_w and x_c.

Finally, the method adds the result to the positional encoding vector to obtain the final input sequence information.

Location information is especially important for attention mechanisms. For example, the words "Tom broke the vase on the table" and "The vase broke the Tom on the table" are almost same for attention mechanism. But the meaning of these two sentences are entirely different. Therefore, we introduce a new mechanism, a novel position encoding, to number the position of each word. By using the parity of trigonometric function, position information is introduced for each word by combining the position vector and the word vector. Therefore, by utilizing its information, the attention mechanism can distinguish words at different positions.

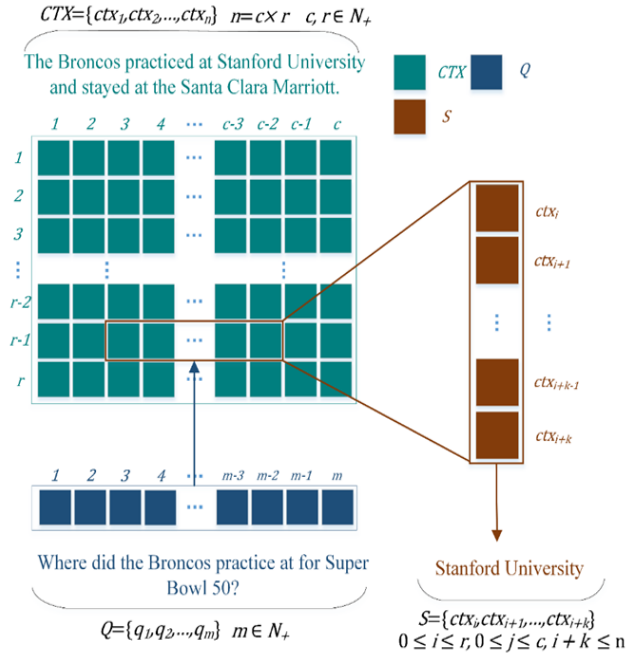

FIGURE 2. The process of answer generation, the answer S is a sub-sequence from paragraph CTX according to the question Q.

The calculation method for position encoding vector PE express as follows:

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d}\right) \tag{4}$$

$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d}\right) \tag{5}$$

where pos represents the position of the word, i represents the dimension of the i-th word, and d represents the dimension of the word vector. In addition to being able to express the absolute position of the sequence, the above equation can also express relative position relationships. We can explain the relationship by the following equation.

$$\sin(\alpha+\beta) = \sin\alpha^* \cos\beta + \cos\alpha^* \sin\beta \tag{6}$$

$$\cos(\alpha+\beta) = \cos\alpha^* \cos\beta - \sin\alpha^* \sin\beta \tag{7}$$

we set the position vector p and q, where $q = p + k$, and k is the distance from p to q. According to formula (6), the $\sin(q) = \sin(p + k)$. Therefore, position vector q can be expressed as the linear change of position vector p, thus representing the relative position information.

In the second part, questions Q, final word vectors $[x_w; x_c]$ and position vectors PE are used as inputs to the MTA module. By using the multi-layer attention to learn different layer information and capture the semantic information of different types, the model can obtain the high-level semantic information of the whole sequence. After that, we send the result of the question code Q(query) and the article context code C (context) which obtained by the MTA module to the context-query attention layer for learning the question and answer information. Inspired by QANet [2], this module can learn the associations between context and query effectively, and obtain keywords that describe the relationship between the query and the context. The module contains two calculation schemes: context-to-query attention A and query-to-context attention B. By using the above two calculation schemes, we can obtain the similarity matrix of query and context, which can enhance the relevance of query and context. The formal expression is as follows:

$$A = \text{softmax}\,(SM, axis = row) \cdot Q^T \tag{8}$$

$$B = A \cdot \text{softmax}(SM, axis = column)^T \cdot C^T \tag{9}$$

where $SM(n * m)$ is the similarity matrix function between context and query, n is the length of context, and m is the length of query. The function SM can be described as follows:

$$SM_{i,j} = f(Q, C) = W_0[Q, C, Q \odot C] \tag{10}$$

where W_0 is a trainable variable and $\odot$ is the element-wise product.

Then, we send the result to the coding layer which consists of 3 MTA modules to learn the relationship between context and query from a global perspective. The three MTAs output M_0, M_1 and M_2 respectively. Finally, the result will be sent to two softmax functions to get the start position and end position of the target answer in the article paragraph. The formal express as follows:

$$pos_{start} = \text{softmax}\,(W_{start}\,[M_0; M_1]) \tag{11}$$

$$pos_{end} = \text{softmax}\,(W_{end}\,[M_0; M_2]) \tag{12}$$

The model's loss function can be expressed as:

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N}\left[\log\left(p_{y_i^{start}}^{start}\right) + \log\left(p_{y_i^{end}}^{end}\right)\right] \tag{13}$$

where y_i^{start}, y_i^{end} represent the start and end positions of the answer in the context.

C. MULTI-LAYER ATTENTION TRANSFORMER UNIT

In order to understand and make full use of the output information of each layer of the network, we add a multi-layer attention method based on Transformer [12]. As shown in Fig. 3, the architecture uses Transformer structure as a base network. It uses a combination of multi-head attention mechanism and feedforward neural network to model sequences. Our method can use the multi-layer attention to learn different layer information and capture the semantic information of different levels in the sequence.

For the basic Transformer building blocks that contain a set of self-attention mechanisms and feedforward networks, we have the following definitions:

$$M^l = LayerNorm\left(Attention\left(Q^{l-1}, K^{l-1}, V^{l-1}\right) + T^{l-1}\right) \tag{14}$$

$$T^l = LayerNorm\left(FFN\left(M^l\right) + M^l\right) \tag{15}$$

where $LayerNorm()$ is a layer normalization function, $Attention()$ is a self-attention calculation function, and

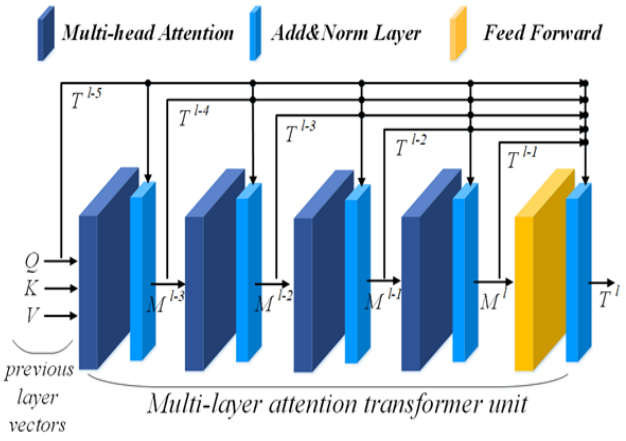

FIGURE 3. Multi-layer attention transformer unit. We changed the single-layer self-attention to a multi-layer attention in transformer and connected the layers in a fully connected manner. We also add the sequence information output from each layer to the next attention layer which strengthened the utilization of the network at different layers and reduces the loss of information.

$FFN()$ is a feedforward neural network with a ReLU function as an activation function. Also, $Q^{l-1}, K^{l-1}, V^{l-1}$ are the query, key and value vectors transformed from the previous layer T^{l-1}, and they are also initialization parameters of $Attention()$.

We first reconstruct the basic Transformer unit structure. In order to obtain key information of query and context, we changed the single-layer self-attention mechanism to a multi-layer attention mechanism, and fully interconnects all layers. The data processing in the multi-layer attention mechanism can be formally defined as follows:

$$\begin{aligned} A^l_{-1} &= Attention\left(Q^{l-1}, K^{l-1}, V^{l-1}\right) \\ A^l_{-2} &= Attention\left(Q^{l-2}, K^{l-2}, V^{l-2}\right) \\ &\cdots \quad \cdots \\ A^l_{-k} &= Attention\left(Q^{l-k}, K^{l-k}, V^{l-k}\right) \\ A^l &= Aggregation\left(A^l_{-1}, A^l_{-2}, \cdots, A^l_{-k}\right) \end{aligned} \tag{16}$$

where A^l_{-k} is the result calculated by the attention function of the l-k layer, and $Aggregation()$ is an aggregate function that unifies the results of each layer. The calculation method is as follows:

$$\begin{aligned} &Aggregation\left(x_1, x_2, \ldots, x_k\right) \\ &= LayerNorm\left(FFN\left([x_1; x_2; \ldots; x_k]\right) + \sum_{i=1}^{k} x_i\right) \end{aligned} \tag{17}$$

we first concatenate $x_1, x_2, \ldots, x_k$, then send them to the feedforward neural network with sigmoid as the activation function, and accumulate all the inputs. Finally, we use the layer normalization function to get the result.

The reason why we use a fully connected layer for the multi-layered attention layer instead of a residual connection is as follows:

1. Use fully connected layer can spread loss directly to the base layer for easy training.
2. The coding information of each layer is an aggregation of all the previous layers, and retains key information of all layers.
3. The final coding result relies on representations from all layers, including both sophisticated and simple features.

By using the Multi-head Attention mechanism, the model can pay attention to the representation information of different subspaces from different locations. The specific calculation method is as follows:

$$MultiHead(Q, K, V) = Concat\left(\text{head}_1, \cdots, \text{head}_h\right) W^o \tag{18}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{19}$$

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{20}$$

where W_i^Q, W_i^K, W_i^V, W^O are the training parameters in the model.

Based on the Transformer structure, we changed the previous multi-head attention layer to the combination of multiple multi-head attention layers. As shown in Fig. 1, The model aggregates the information of each attention layer and sends it to the next layer to make full use of the information of each layer.

D. MULTI-LAYER ATTENTION TRANSFORMER AGGREGATION ENCODER

Based on the Transformer structural model, we use layer aggregation techniques to integrate the information of each layer better. The structure of the MTA is shown in Figure 4.

By aggregating nodes, we can better utilize the information between each unit to analyze the sequence information from multiple aspects and ensure the efficient utilization of information.

The multi-layer attention transformer units are aggregated according to the following formula:

$$\hat{T}^i = \begin{cases} Aggregation\left(T^{2i-1}, T^{2i}\right) & i = 1 \\ Aggregation\left(T^{2i-1}, T^{2i}, \hat{T}^{i-1}\right) & i > 1 \end{cases} \tag{21}$$

The aggregate function $aggregation()$ is the one as formula (17). We aggregate the nodes of the same layer into one node, then send the result back to the linear backbone network as the input of the next layer. All the aggregation steps replace the layering combine operation by an addition operation so that the computational complexity can be reduced while maintaining the size of each layer.

IV. EXPERIMENTS

A. DATASET

In experiments, we used the SQuAD [4] data set proposed by Rajpurkar *et al.* It contains a total of 107,785 questions, as well as 536 pieces of material that contain the target

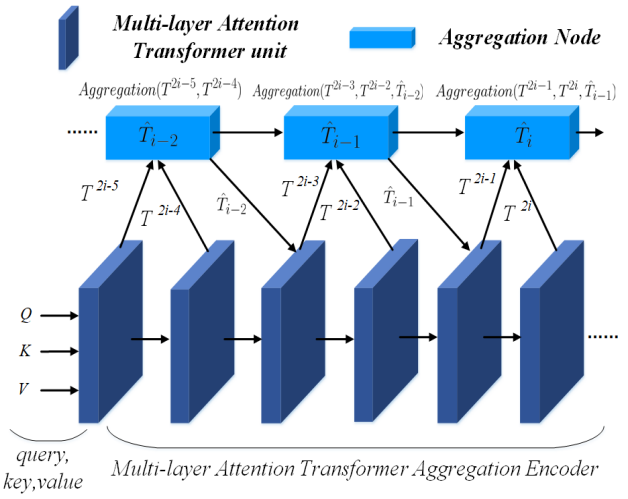

FIGURE 4. Multi-layer attention transformer aggregation encoder. We aggregate each multi-layer attention transformer unit between aggregation nodes and transmit the aggregated information to the backbone network to further enhance the utilization of information. In addition, since each layer transmits information in parallel, the computational efficiency of the model is improved.

answers. Table 1 shows an example of the SQuAD data set. SQuAD [4] has extracted more than 100,000 question-answer pairs from hundreds of articles on Wikipedia through crowdsourcing. Compared to other datasets like MCTest [27], Algebra [28], Science [29] and WiKiQA [30], the reason we chose the SQuAD dataset is that the number of questions in SQuAD is far greater than them. On the other hand, the number of questions in CNN/Daily [31] Mail and CBT [32] data sets are relatively large, but these are both cloze-style datasets, rather than a real question answering data.

B. NETWORK PARAMETER SETTINGS

Some of the hyper-parameters used in the neural network are shown in Table 2. We use the ADAM optimization algorithm [33] to train the model. Where $\beta_1 = 0.8, \beta_2 = 0.999, \epsilon = 10^{-7}$. For the setting of the learning rate lr, we use the warm-up scheme to gradually increase from 0.0 to 0.001 in the first 2000 steps of the model training, and then maintain a steady rate for training.

C. EXPERIMENTAL RESULTS AND ANALYSIS

In the answer generation task, we mainly evaluate the performance of the model with EM and F1 scores. EM is a score for complete match, which requires the model's prediction be exactly the same as the answer in the data set. F1 score is used to measure the degree of fuzzy matching between the model's prediction and the answer, it takes into account both the accuracy and recall of the model, so the evaluation results are more objective.

As shown in the plot(a) of Fig.5, the loss rate of the model is relatively large during the training process. When attention heads is set to 1 and attention layers is set to 3, the network loses a lot of information during the feedforward process and it is difficult to capture key information, so the model is difficult to converge. In plot(b), we change training steps from 30000 to 50000, attention dimension from 96 to 128, and increase the attention layers from 3 to 4. The EM score increased from 67.3 to 68.9 and the F1 score increased from 76.2 to 78. With the increment of the number of training rounds and attention layers, the model can capture and take full advantage of semantic information of the sentence. In the plot (c), we set the number of attention heads to 8, by using the multi-head attention mechanism, the model can find the key words in the sentence, so that the meaning of the context can be clearly expressed. The EM score and the F1 score are

TABLE 1. The example of squad dataset.

Paragraph	Quantity	Answer
The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.	At what university's facility did the Panthers practice?	San Jose State
	At what university's facility did the Broncos practice?	Stanford University
	In what city's Marriott did the Panthers stay?	San Jose
	In what city's Marriott did the Broncos stay?	Santa Clara
	Where did the Broncos practice at for Super Bowl 50?	Stanford University
One of the oldest depictions of civil disobedience is in Sophocles' play Antigone, in which Antigone, one of the daughters of former King of Thebes, Oedipus	Sophocles demonstrated civil disobedience in a play that was called?	Antigone
	Who is Antigone's father in the play?	Oedipus
	What character in the play portrays civil disobedience?	Oedipus
	Antigone was a play made by whom?	Sophocles

TABLE 2. Experimental super parameter configuration.

Parameter	Value
Glove dimension	300
L2_regularizer λ	3*10-7
Dropout	0.1
Attention dimension	128
Attention heads	8
Train steps	80000
Batch size	32

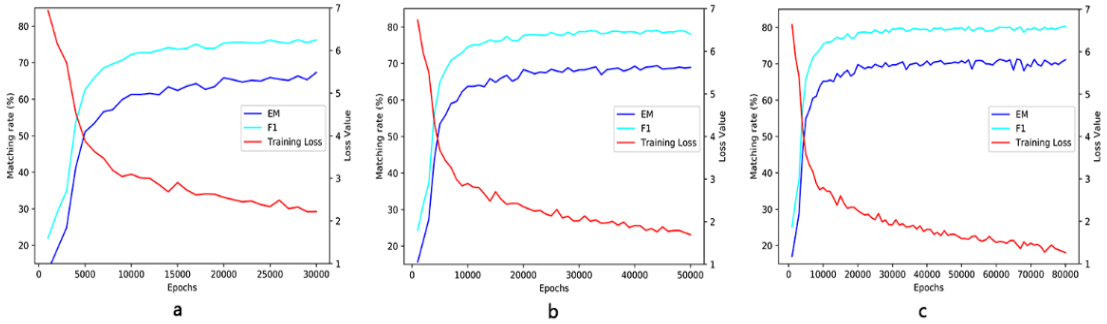

FIGURE 5. The performance breakdown with different configuration in training process. Plot (a) shows the trend of loss, EM and F1 scores in training steps = 30000, attention dimension = 96, attention heads = 1 and attention layers = 3; In the plot(b),the training steps = 50000, attention dimension = 128, attention layers = 4; plot (c) shows the trend in training steps = 80000, attention dimension = 128, attention Heads = 8 and attention layers = 4. The model successfully converges in the 80,000th round with EM and F1 reaching the maximum value (EM = 71.1, F1 = 80.3).

TABLE 3. The test results of the answer generation.

Paragraph	Quantity	Answer
In 1664, Peter Stuyvesant, the Director-General of the colony of New Netherland, surrendered New Amsterdam to the English without bloodshed. The English promptly renamed the fledgling city "New York" after the Duke of York (later King James II).	What did the English call New Amsterdam after its capture?	New York
	What was the regnal name of the Duke of York?	King James II
	In what year did the English take over New Amsterdam?	1664

TABLE 4. Model training effect under multiple parameter combinations.

Model	Train Steps	Attention Dimension	Attention Heads	Attention Layers	Unit Numbers	EM	F1
1	30000	96	1	3	4	67.3	76.2
2	30000	96	1	4	6	67.5	76.6
3	50000	96	1	3	4	68.1	77.4
4	50000	128	1	4	6	68.9	78.0
5	80000	128	8	3	4	70.7	79.6
6	**80000**	**128**	**8**	**4**	**6**	**71.1**	**80.3**

reached 71.1 and 80.3, respectively. It can be seen that the accuracy of our model has increased significantly by using the multi-head attention mechanism and MTA module.

In order to test the validity of the model, we use the test set to test the accuracy of the model and the ability to generate answers. We conducted three sets of experiments separately, and the experimental results are shown in Table 3. We choose three typical paragraphs and questions, where the colored words in the paragraphs are the answers to the questions. As can be seen from the table, our model shows quite good performance.

D. ABLATION STUDIES AND COMPARISONS WITH PRIOR METHODS

To demonstrate whether AG-MTA can effectively generate the correct answer, Table 4 shows the EM and F1 scores of different networks. Training steps indicates the number of training steps. Attention Dimension indicates the hidden layer dimension of the attention network. Attention Heads indicates the number of attention heads, Attention Layers indicates the number of layers of attention, and Unit Numbers indicates the number of layers of the multi-layer attention Transformer unit.

As shown in Table 4, by comparing Model 1 and Model 3, it can be seen that the more training steps, the better the model fits. In addition, by comparing Model 1 and Model 2, or Model 5 and Model 6, it can be seen that as the number of Attention Layers and Unit Number increases, the model can obtain more information representations at different positions. Through the experiments of multiple sets of different parameters, we obtained several sets of EM and F1 scores respectively. By comparing, we can get the setting of each parameter value of the model under the best effect. Moreover, EM and F1 achieved the highest scores of 71.1 and 80.3.

In order to measure the performance of our model, we compared it to other representative methods. As shown in Table 5, Dev represents model's test score under the development set, and Test represents model's test score under the test set. Compared with other methods, whether in Dev or Test, AG-MTA has a greater improvement in performance. Compared with models that only use the LSTM network (such as LR Base-Line [4]) or attention mechanism (such as BiDAF [10]). AG-MTA combines the context information and extracts key semantic information by using MTA module, position encoding, and multi-head attention mechanism. Most importantly, since the coding part of our model uses the pure attention mechanism scheme and data-parallel computing, with more data, the model can get better performance.

In addition, to help qualitatively evaluate our MTA module, position encoding and multi-head attention mechanism

TABLE 5. The comparison experiment of different answer generation model.

Models	Dev		Test	
	EM	F1	EM	F1
LR Baseline [4]	39.8	51.0	40.4	51.0
Match-LSTM with Ans-ptr [34]	54.4	68.2	59.5	70.3
Dynamic Chunk Reader [35]	62.4	71.2	62.5	71.0
Multi-Perspective Matching [34]	66.1	75.8	65.5	75.1
FastQA [36]	67.8	76.3	68.4	77.1
BiDAF [10]	67.7	77.3	68.0	77.3
Document Reader	69.5	78.8	70.0	79.0
JNet [37]	68.7	77.4	70.6	79.8
SEDT [38]	68.1	77.5	69.6	79.7
FastQAExt [39]	69.2	78.1	70.8	78.9
AG-MTA(our)	70.7	81.4	71.1	80.3

TABLE 6. Ablation experiments with different module combinations.

Models	Multi-head attention	Position encoding	MTA	EM	F1
(A)	√	√	√	**71.1**	**80.3**
(B)		√	√	67.2	76.0
(C)	√		√	64.6	72.1
(D)	√	√		60.2	69.5

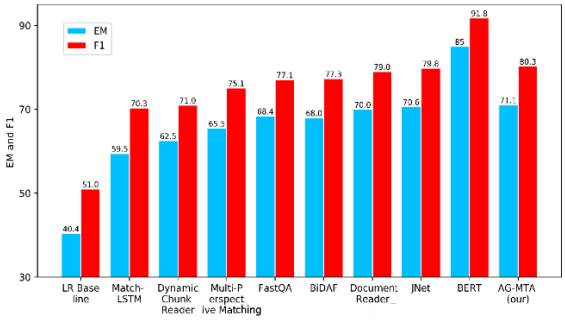

FIGURE 6. Performance of different models in the answer generation experiment.

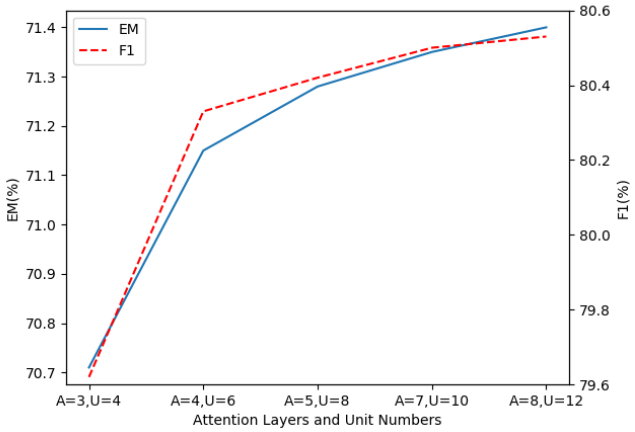

FIGURE 7. The EM score and F1 score on AG-MTA in function of Attention layers and unit number.

methodology, we conduct extensive ablation experiments. As shown in Table 6, the experimental results show that the model (A) with all modules obtained the best experimental performance. By comparing the experimental results of (A), (B) and (C), we observe that with help of the position encoding and the multi-head attention mechanism, model (A) can use logical semantic information to express the relationship among words. By comparing the experimental results of (A) and (D), it can be seen that the MTA module can significantly improve the performance of the model (A) by fusing semantic information in different locations from base layer to top layer.

We also made a formal comparison of the results, as shown in Figure 6. It can be seen that with the continuous improvement of the network, the performance of the network is getting better and better, and the EM and F1 values of our model have achieved 71.1 and 80.3 respectively.

E. DISCUSSION

Compared to the performance of other answer generation methods, our answer generation model produces significant improvement. By applying MTA module testing for AG-MTA with different parameters, we can improve EM from 67.3% to 71.1% and F1 from 76.2 % to 80.3%.

We have also studied the impact of the number of attention layers and the number of multi-layer attention Transformer unit module on the EM score and F1 score. Base on the model 6 in Table 4, we tune some parameters, so that Train Steps = 80000, Attention Dimension = 128 and Attention Heads = 8; and accordingly adjust the number of Attention Layers and Unit Numbers. The experimental result is shown in Figure 7, where A is Attention Layers and U is Unit Numbers. It can be observed in Figure 7, as the number of Attention Layers and Unit Numbers increases, the growth rate of EM score and F1 score gradually decreases. However, the computing resources consumed by the model have increased exponentially. As the model's complexity is increasing, it is difficult to fit. Therefore, we set Attention Layers = 4 and Unit Numbers = 6 to avoid consuming too many computing resources.

As shown in Table 5, the experimental results indicate that our AG-MTA can well exploit contextual correlation preserved in paragraphs. Nevertheless, the lack of similarity matching between questions and paragraphs will lead to poor logical reasoning ability. Compared with BERT [40], the AG-MTA is not achieving the best performance. But the

BERT also has high demand for hardware. Therefore, in terms of practicality, our model has significant performance with general applicability.

V. CONCLUSION

In this paper, we propose an end-to-end model for answer generation based on multi-layer Transformer aggregation coder. The model enhances the contextual correlation and improves the accuracy of the answer generation. We propose MTA to focus on information representation at different levels and aggregate the nodes of the same layer to combine the context information. Furthermore, a novel position encoding method that make full use of absolute position and relative position information of the word is designed to enhance the relationship of each word. Experiments on the SQuAD dataset verified that our model has a significant improvement over the state-of-the-art method. Moreover, ablation study on multi-head attention mechanism and position encoding have been done to prove the effectiveness of each component. In the experiments, we found that time cost increases with the complexity increase of the network structure. Hence, in our future work, we will pay more attention to the experiment on network architecture optimization tasks, and the reduction of model parameters to make these methods have greater applicability.

REFERENCES

[1] S. Wang and J. Jiang, "Machine comprehension using match-LSTM and answer pointer," 2016, *arXiv:1608.07905*. [Online]. Available: http://arxiv.org/abs/1608.07905

[2] A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "QANet: Combining local convolution with global self-attention for reading comprehension," 2018, *arXiv:1804.09541*. [Online]. Available: https://arxiv.org/abs/1804.09541

[3] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2383–2392.

[4] J. Liu, Y. Yang, S. Lv, J. Wang, and H. Chen, "Attention-based BiGRU-CNN for Chinese question classification," *J. Ambient Intell. Hum. Comput.*, Jun. 2019, doi: 10.1007/s12652-019-01344-9.

[5] L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman, "Deep learning for answer sentence selection," 2014, *arXiv:1412.1632*. [Online]. Available: http://arxiv.org/abs/1412.1632

[6] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "LSTM-based deep learning models for non-factoid answer selection," 2015, *arXiv:1511.04108*. [Online]. Available: http://arxiv.org/abs/1511.04108

[7] J. Liu, H. Ren, M. Wu, J. Wang, and H.-J. Kim, "Multiple relations extraction among multiple entities in unstructured text," *Soft Comput.*, vol. 22, no. 13, pp. 4295–4305, Jul. 2018.

[8] S. P. Lende and M. M. Raghuwanshi, "Question answering system on education acts using NLP techniques," in *Proc. World Conf. Futuristic Trends Res. Innov. Social Welfare (Startup Conclave)*, Feb. 2016, pp. 1–6.

[9] D. Wang and E. Nyberg, "A long short-term memory model for answer sentence selection in question answering," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Short Papers)*, vol. 2, Jul. 2015, pp. 707–712.

[10] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," 2016, *arXiv:1611.01603*. [Online]. Available: https://arxiv.org/abs/1611.01603

[11] B. Dhingra, H. Liu, Z. Yang, W. Cohen, and R. Salakhutdinov, "Gated-attention readers for text comprehension," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, Jul. 2017, pp. 1832–1846.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[13] J. Liu, L. Lin, H. Ren, M. Gu, J. Wang, G. Youn, and J.-U. Kim, "Building neural network language model with POS-based negative sampling and stochastic conjugate gradient descent," *Soft Comput.*, vol. 22, no. 20, pp. 6705–6717, Oct. 2018.

[14] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, and J. Jiang, "R^3: Reinforced reader-ranker for open-domain question answering," 2017, *arXiv:1709.00023*. [Online]. Available: http://arxiv.org/abs/1709.00023

[15] Z. Yang, J. Hu, R. Salakhutdinov, and W. W. Cohen, "Semi-supervised QA with generative domain-adaptive nets," 2017, *arXiv:1702.02206*. [Online]. Available: http://arxiv.org/abs/1702.02206

[16] R. Ghaeini, X. Z. Fern, H. Shahbazi, and P. Tadepalli, "Dependent gated reading for cloze-style question answering," 2018, *arXiv:1805.10528*. [Online]. Available: http://arxiv.org/abs/1805.10528

[17] W. Zhong, D. Tang, N. Duan, M. Zhou, J. Wang, and J. Yin, "A heterogeneous graph with factual, temporal and logical knowledge for question answering over dynamic contexts," 2020, *arXiv:2004.12057*. [Online]. Available: http://arxiv.org/abs/2004.12057

[18] X. Shi, I. Padhi, and K. Knight, "Does string-based neural MT learn source syntax?" in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1526–1534.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365

[20] A. Anastasopoulos and D. Chiang, "Tied multitask learning for neural speech translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol., (Long Papers)*, vol. 1, Jun. 2018, pp. 82–91.

[21] T. Bell and B. Penchas, "Lightweight convolutional approaches to reading comprehension on SQuAD," 2018, *arXiv:1810.08680*. [Online]. Available: http://arxiv.org/abs/1810.08680

[22] X. Zhou, B. Hu, Q. Chen, and X. Wang, "Recurrent convolutional neural network for answer selection in community question answering," *Neurocomputing*, vol. 274, pp. 8–18, Jan. 2018.

[23] Y. Tay, L. A. Tuan, and S. C. Hui, "Multi-cast attention networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2299–2308.

[24] L. Dong, F. Wei, M. Zhou, and K. Xu, "Question answering over freebase with multi-column convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. (Long Papers)*, vol. 1, Jul. 2015, pp. 260–269.

[25] X. He and D. Golub, "Character-level question answering with attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1598–1607.

[26] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[27] M. Richardson, C. J. Burges, and E. Renshaw, "MCTest: A challenge dataset for the open-domain machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Oct. 2013, pp. 193–203.

[28] N. Kushman, Y. Artzi, L. Zettlemoyer, and R. Barzilay, "Learning to automatically solve algebra word problems," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, Jun. 2014, pp. 271–281.

[29] P. Clark and O. Etzioni, "My computer is an honor student—But how intelligent is it? Standardized tests as a measure of AI," *AI Mag.*, vol. 37, no. 1, p. 5, 2016.

[30] Y. Yang, W.-T. Yih, and C. Meek, "WikiQA: A challenge dataset for open-domain question answering," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2013–2018.

[31] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading Children's books with explicit memory representations," 2015, *arXiv:1511.02301*. [Online]. Available: http://arxiv.org/abs/1511.02301

[32] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[34] Z. Wang, H. Mi, W. Hamza, and R. Florian, "Multi-perspective context matching for machine comprehension," 2016, *arXiv:1612.04211*. [Online]. Available: http://arxiv.org/abs/1612.04211

[35] Y. Yu, W. Zhang, K. Hasan, M. Yu, B. Xiang, and B. Zhou, "End-to-end answer chunk extraction and ranking for reading comprehension," 2016, *arXiv:1610.09996*. [Online]. Available: http://arxiv.org/abs/1610.09996

[36] D. Weissenborn, G. Wiese, and L. Seiffe, "FastQA: A simple and efficient neural architecture for question answering," 2017, *arXiv:1703.04816*. [Online]. Available: http://arxiv.org/abs/1703.04816

[37] J. Zhang, X. Zhu, Q. Chen, L. Dai, S. Wei, and H. Jiang, "Exploring question understanding and adaptation in neural-network-based question answering," 2017, *arXiv:1703.04617*. [Online]. Available: http://arxiv.org/abs/1703.04617

[38] K. Lee, S. Salant, T. Kwiatkowski, A. Parikh, D. Das, and J. Berant, "Learning recurrent span representations for extractive question answering," 2016, *arXiv:1611.01436*. [Online]. Available: http://arxiv.org/abs/1611.01436

[39] D. Weissenborn, G. Wiese, and L. Seiffe, "Making neural QA as simple as possible but not simpler," in *Proc. 21st Conf. Comput. Natural Lang. Learn. (CoNLL)*, 2017, p. 271.

[40] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

SHENGJIE SHANG received the B.E. degree from Liaocheng University, in 2018. He is currently pursuing the M.E. degree with Shanghai Maritime University.

His current research interests are computer vision, natural language processing, and machine learning.

JIN LIU (Member, IEEE) received the B.S. degree from Lanzhou University, the M.S. degree from the University of Electrical Science and Technology of China, and the Ph.D. degree from Washington State University. He is currently a Professor with Shanghai Maritime University. His research interests include deep learning, nature language processing, and computer vision. He is a member of CAAI and CCF.

YIHE YANG received the B.S. degree in software engineering from Shanghai Maritime University, Shanghai, in 2018, where he is currently pursuing the M.E. degree.

His current research interests are data mining, natural language processing, and machine learning.