Multi-Level Composite Neural Networks for Medical Question Answer Matching

Dong Ye[1†], Sheng Zhang[1], Hui Wang[1*], Jiajun Cheng[1], Xin Zhang[1], Zhaoyun Ding[1], Pei Li[1]
[1] College of Systems Engineering
National University of Defense Technology,
Changsha, P. R. China
[†] Email: kingod@foxmail.com
[*] Hui Wang is the corresponding author, email: huiwang@nudt.edu.cn

***Abstract*— The online medical question answering community where patients can ask medical questions becomes quite popular in recent years. Deep learning techniques have been widely used in medical care field and a large number of Natural Language Processing tasks, which makes it possible to answer the medical question automatically. The challenge in extracting high-level semantic information from the question and the answer is the key issue in question answer matching. This paper proposes a multi-level composite convolutional neural networks framework to alleviate the issue in question answer matching. The model does not just stack multiple convolutional neural networks together, but extracts information from each layer and then concatenates the features at the end of the framework. As a consequence, the framework is able to better capture high-level semantics and reach a new state-of-the-art performance on cMedQA dataset.**

I. Introduction

Recent advances in deep learning stimulate the development of a wide variety of areas, such as Education, Economy, Health Care, etc. In recent year, humans are paying more attention to their health condition and the online medical question answering community becomes quite active. The community provides the place where the qualified doctors answer the medical question proposed by patients online, which may alleviate the limited medical resources.

Question answering (QA) has always been the key issue in artificial intelligence. The text of Medical QA is highly professional related, which poses a great challenge for feature extracted algorithms. Currently used deep neural networks provide an approach to extract deep semantic information from the text. Zhang et.al.[1] proposed the end-to-end deep neural network frameworks to solve the problem. However, the framework used in their paper is just one level, which can not effectively extract semantic information.

The medical question answer investigated in this paper is a type of Community Question Answering (cQA), the form of cQA is as follows: Given a question q and a answers set A, where $A = \{a_1, a_2, ..., a_k\}$. Given a candidate answers set C, where $C = \{c_1, c_2, ..., c_n\}$, $k \ll n$, and $A \subset C$. The goal of cQA is to design a specific model and give the best candidate answer $c_i \in C$ to a given question q, which satisfies $c_i \in A$.

In this paper, multi-level composite deep neural networks framework is proposed to better capture the high dimensional information. The composite framework does not just stack different neural networks together, but extracts features from each level and combines them at the end of the framework. In this way, this composite structure can enrich the final feature representation, have a better understanding of the semantics of the question and answer, and therefore reach the new state-of-the-art performance.

The paper is constructed as follow, Section 2 illustrates related work, Section 3 presents several multi-level models and our proposed multi-level composite CNN networks, Section 4 shows the experimental results on a Medical QA dataset and provides the analysis, and Section 5 draws the conclusion.

II. Related Work

Question answering has always been the hot but challenging topic in Natural Language Processing these years. Some related work has explored in medical question answering problem.

Dodiya et.al.[2] first studied the classification of questions in Medical QA, which provided a basis for the following work.

Jan et.al.[3] proposed a rule-based Medical QA system. However, the patients are different in expressing the problems, and the rule-based system can not well-cover all language rules. The model cannot handle the question if it does not exist in the rule template.

Ben et.al.[4][5] used the conditional random field, support vector machine and other methods to extract the semantic relationships in medical questions for answer searching. They converted the question into database query format language (SPARQL), but the process is quite inefficient and manpower wasteful.

Terol et.al.[6] transformed questions into a logical structure, and solved the problem through this structure. They used the Unified Medical Language System (UMLS) to identify medical terms in the transformation to logical structure, and determined the terms in the open domain through WordNet. Delbecque et.al.[7] used UMLS to solve the Medical QA, but this method is low in efficiency and poor in applicability when handle problems that are not included in UMLS.

978-1-5386-4210-8/18/$31.00 ©2018 IEEE
DOI 10.1109/DSC.2018.00028

Deep learning techniques accelerate the development of question answering, since deep neural networks are capable of extracting deep semantic information from the text. The semantic information extracted is often used to calculate similarity between the question and the answer.

Hu et.al.[8] proposed a model for sentence similarity matching, which consists of a word vector layer, a convolution neural network layer and a pooling layer. Because the convolution layer performs convolution operation on local information, it can accurately extract the semantic information from local context.

Feng et.al.[9] proposed a deep learning framework based on convolution neural networks. They compared feature extraction capabilities of different CNN networks in terms of their parameter setup. The results show that the CNNs which share the same parameters when extracting features from the question and answer reach the highest accuracy score.

Tan et.al.[10] used recurrent neural network instead of convolution neural network to make full use of the advantage on sequence problems. At the same time, they used bi-directional recurrent neural network to extract semantic information from the two directions of the sequence to improve the matching accuracy.

Zhang et.al.[1] proposed a framework that uses character embeddings and multi-scale CNNs to extract the semantic information of Chinese medical text. The experiment on cMedQA, a dataset they collect, shows their models' superiority over other methods that use word character embeddings and single-scale CNN.

III. Methodology

A. CNN+biLSTMs Model

Shown in Figure 1 is the deep neural network model based on Convolutional Neural Network and bi-directional Long Short Term Memory networks (bi-LSTMs). Given a question sentence and a answer sentence, they are represented as embeddings and then fed to convolutional neural network, where the local contextual information was extracted. After that, the biLSTMs are used to capture the information extracted by CNN, and then the pooling layer is applied. Finally, cosine is used to measure the similarity between the question and the answer.

Specifically, let $D_q = [c_1, c_2, \cdots, c_{l_D}]$ and $D_a = [c_1, c_2, \cdots, c_{l_D}]$ be the character embeddings of the question and the answer respectively, where $D_q \in \mathbb{R}^{d_w \times l_D}$ and $D_a \in \mathbb{R}^{d_w \times l_D}$ are two-dimension embedding matrices, with the same shape. They are first fed to CNN model, shown as:

$$O_q = \text{CNN}(D_q) \tag{1}$$

and

$$O_a = \text{CNN}(D_a). \tag{2}$$

After that, biLSTMs are used to extract information:

$$H_q = \text{biLSTMs}(O_q) \tag{3}$$

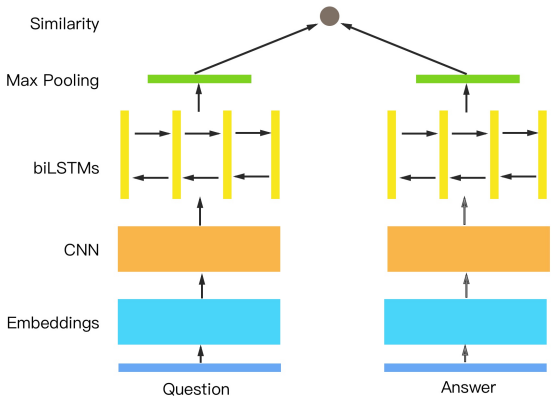

Fig. 1. This figure illustrates CNN+biLSTMs model.

and

$$H_a = \text{biLSTMs}(O_a). \tag{4}$$

Next, the pooling layer is used to extracted useful information and simplify the representation:

$$q = \text{Pool}_{\max}(H_q) \tag{5}$$

and

$$a = \text{Pool}_{\max}(H_a). \tag{6}$$

Finally, cosine is used to measure the similarity between the question and the answer:

$$cos(q,a) = \frac{q^T a}{||q|| \cdot ||a||}. \tag{7}$$

B. BiLSTM+CNN Model

Shown in Figure 2 is the deep neural network model based on bi-directional LSTMs and Convolutional Neural Network. Given a question sentence and a answer sentence, they are represented as character embeddings and then fed to biLSTMs, where the sequence information was extracted. After that, the convolutional neural network is used to capture the information extracted by last layer, and then the pooling layer is applied on the dimension of the convolution kernel. Finally, cosine is used to measure the similarity between the question and the answer.

Specifically, let $D_q = [c_1, c_2, \cdots, c_{l_D}]$ and $D_a = [c_1, c_2, \cdots, c_{l_D}]$ be the character embeddings of the question and the answer respectively, where $D_q \in \mathbb{R}^{d_w \times l_D}$ and $D_a \in \mathbb{R}^{d_w \times l_D}$ are two-dimension embedding matrices, with the same shape. They are first fed to biLSTMs model, shown as:

$$H_q = \text{biLSTMs}(D_q) \tag{8}$$

and

$$H_a = \text{biLSTMs}(D_a). \tag{9}$$

After that, CNN is used to extract information:

$$O_q = \text{biLSTMs}(H_q) \tag{10}$$

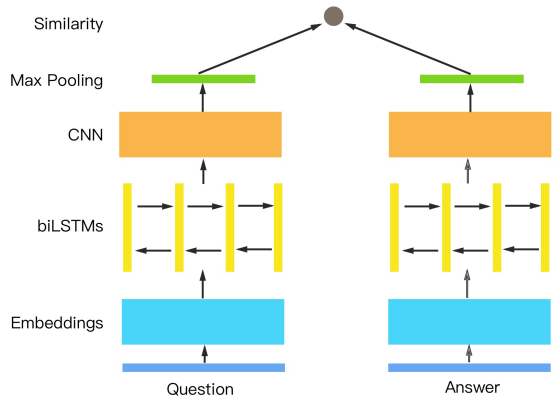

Fig. 2. This figure illustrates biLSTMs+CNN model.

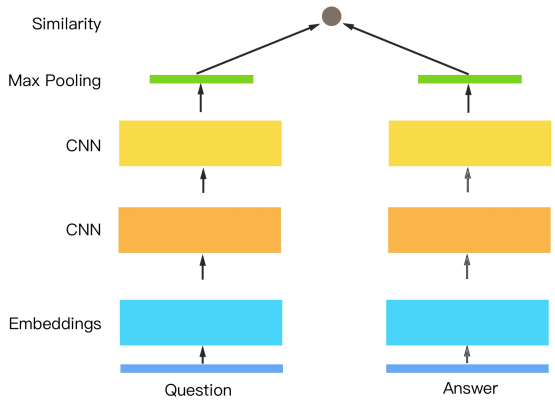

Fig. 3. This figure illustrates CNN+CNN model.

and

$$O_a = \text{biLSTMs}(H_a). \tag{11}$$

Next, the pooling layer is used to extracted useful information and simplify the representation:

$$q = \text{Pool}_{\max}(O_q) \tag{12}$$

and

$$a = \text{Pool}_{\max}(O_a). \tag{13}$$

Finally, cosine is used to measure the similarity between the question and the answer.

C. CNN+CNN

Shown in Figure 3 is the deep neural network model based on Convolutional Neural Network and Convolutional Neural Network. Given a question sentence and a answer sentence, they are represented as character embeddings and then fed to the first convolutional neural network, where the local contextual information was extracted. After that, the second convolutional neural network is used to capture the information extracted by last layer, and then the pooling layer is applied on the dimension of the convolution kernel. Finally, cosine is used to measure the similarity between the question and the answer.

Specifically, let $D_q = [c_1, c_2, \cdots, c_{l_D}]$ and $D_a = [c_1, c_2, \cdots, c_{l_D}]$ be the character embeddings of the question and the answer respectively, where $D_q \in \mathbb{R}^{d_w \times l_D}$ and $D_a \in \mathbb{R}^{d_w \times l_D}$ are two-dimension embedding matrices, with the same shape. They are fed to the first CNN model, shown as:

$$O_q^{(1)} = \text{CNN}_1(D_q) \tag{14}$$

and

$$O_a^{(1)} = \text{CNN}_1(D_a). \tag{15}$$

After that, the second CNN is used to extract information:

$$O_q^{(2)} = \text{CNN}_2(O_q^{(1)}) \tag{16}$$

and

$$O_a^{(2)} = \text{CNN}_2(O_a^{(1)}). \tag{17}$$

Next, the pooling layer is used to extracted useful information and simplify the representation:

$$q = \text{Pool}_{\max}(O_q^{(2)}) \tag{18}$$

and

$$a = \text{Pool}_{\max}(O_a^{(2)}). \tag{19}$$

Finally, cosine is used to measure the similarity between the question and the answer.

D. Multi-Level Composite CNNs

The models described above just stack different deep neural networks, and choose the output of the last neural networks as the output of the whole model. However, during the process of training, each model may play an important role and different levels may extract different types of information. Therefore, it is necessary to extract information from each level of the model and use them as the last representation.

In this paper, we propose a Multi-Level Composite Convolutional Neural Networks framework, which extracts information from each level and concatenate them as a representation vector instead of just staking the neural networks.

Figure 4 illustrates the Multi-Level Composite Convolutional Neural Networks framework. Given a question sentence and a answer sentence, they are represented as character embeddings and then fed to the Multi-Level Composite Convolutional Neural Networks. On the one hand, each output of CNN layer is used as the input of the next CNN layer and on the other hand, the pooling layer is applied to the output of the each neural network. After that, we can concatenate the output of each pooling layer, and get the vectors of the question and the answer. Finally, cosine is used to measure the similarity between the question and the answer. The model uses multi-level structure, and the extraction of semantics is deeper than multi-scale CNNs

model[1]. Therefore, it can extract higher dimensional semantic features and have better understanding of semantic information.

The modules with the same color in our framework share the same parameter. The neural network with the same parameters can effectively improve the accuracy of the model. At the same time, sharing the same parameter can reduce the number of parameters of the whole model and speed up the training procedure of the model. The same level of convolutional neural networks share the same parameters, and different levels of the networks use different parameters.

Specifically, the embeddings are fed to the first CNN layer and we get:

$$O_q^{(1)} = \mathrm{CNN}_1(D_q) \tag{20}$$

and

$$O_a^{(1)} = \mathrm{CNN}_1(D_a). \tag{21}$$

At the same time, pool from the output of the neural network,

$$q^{(1)} = \mathrm{Pool}_{max}(O_q^{(1)}) \tag{22}$$

and

$$a^{(1)} = \mathrm{Pool}_{max}(O_a^{(1)}). \tag{23}$$

After that multi-level CNNs are applied chronologically. The input of each CNN layer is the output of last CNN layer, and we can get the output of present CNN layer, namely:

$$O_q^{(i)} = \mathrm{CNN}_1(O_q^{(i-1)}) \tag{24}$$

and

$$O_a^{(i)} = \mathrm{CNN}_1(O_a^{(i-1)}). \tag{25}$$

After that, we apply pooling layer to extract information:

$$q^{(i)} = \mathrm{Pool}_{max}(O_q^{(i)}) \tag{26}$$

and

$$a^{(i)} = \mathrm{Pool}_{max}(O_a^{(i)}). \tag{27}$$

After that, we can concatenate the output of each pooling layer, and get the vectors of the question and the answer, which are $q = [q^{(1)}, q^{(2)}, \cdots, q^{(i)}]$ and $a = [a^{(1)}, a^{(2)}, \cdots, a^{(i)}]$. Finally, cosine is used to measure the similarity between the question and the answer.

In this paper, in order to simplify the training process and find out the best hyper-parameters, we adopt same-sized filters of CNNs, which means that the width of filter is fixed. Actually, higher layer CNN is able to extract wider information than lower CNN, since the former perform the convolutional operation on the later.

E. Objective Function

When training the deep neural networks, a question, a ground truth answer and a randomly chose negative answer are fed together to the network. By this mechanism, we do negative sampling for each question q_i and produce multiple (q_i, a_i^+, a_i^-) triples.

In this paper, the Max Margin Loss Function is used as objective function to train the neural networks. The formula is shown as follows:

$$L = \frac{1}{N}\sum_{i=1}^{N} max\{0, M - k(q_i, a_i^+) + k(q_i, a_i^-)\}, \tag{28}$$

M is the margin value, and N is the size of the training dataset. In the training, if $k(q_i, a_i^+) - k(q_i, a_i^-) > M$ means that the correct answer is much higher than the wrong answer, the function value is 0. The parameter list of the model is Θ, the strategy of updating parameters by the traditional random gradient descent method is:

$$\Theta \leftarrow \Theta - \eta \cdot \nabla_\Theta L(\Theta; x^{(i)}; y^{(i)}), \tag{29}$$

η is the step and η is a constant value. And $L(\Theta; x^{(i)}; y^{(i)})$ is the Max Margin Loss Function which calculated by data set$(x^{(i)}; y^{(i)})$.

But the rate is not easy to be determined. If choose the large learning rate, it is very easy to cross the optimal solution. But if choose the small learning rate, it's very easy to fail into the local optimal solution. In order to alleviate these problems, the Adagrad[11] gradient algorithm improves the random gradient descent algorithm. Dean et.al.[12] found that the Adagrad algorithm can greatly improve the robustness of the random gradient descent algorithm and apply to the large-scale neural network. Pennington et.al.[13] used Adagrad to train Glove word vectors, because the training of high-frequency words in text is faster than low-frequency words, so using Adagrad can efficiently train word vectors.

Set $g_i = \nabla_\Theta L(\Theta; x^{(i)}; y^{(i)})$, the learning rate of the Adagrad gradient algorithm is:

$$\eta = \frac{1}{\sqrt{\sum_{t=1}^{i} g_t^2 + \varepsilon}}, \tag{30}$$

ε is a small constant to avoid denominator to be 0. From the formula, it can be seen that with the training of the model, the learning rate will gradually decrease, thus to avoid the problems caused by the traditional gradient descent algorithm.

IV. Experiment and Results

To evaluate the models proposed in this paper, this section describes the experiment on a medical question answer dataset and analyses the parameters of our models.

A. Experimental Setup

All questions and answers use the gensim to construct the character embeddings and the word embeddings. In this experiment, the dataset called cMedQA[1] was applied. This paper chooses the ICTCLAS word segmentation tool to

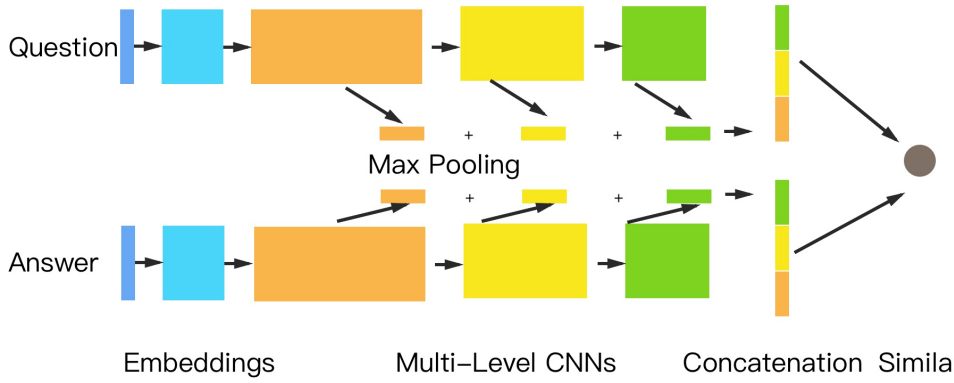

Fig. 4. This figure illustrates Multi-Level Composite CNNs model.

segment Chinese text. And the dimension of the embeddings is set to 200. In this experiment, the length of the question and answer sentence is fixed at 200 words or 400 characters.

For the multi-level convolutional neural network, the convolution scale is 3 and the number of each convolution kernel is 400. For biLSTMs, its outputs have 100 dimensions in each direction.

In this experiment, during the training process, we use 30 triples for each question, and the Adagrad gradient optimization algorithm is used to train the neural network. The initial training rate is set to 0.01. The max-margin constant is set to 0.05.

B. Metrics

The Metrics used in this experiment is top-1 accuracy, noted as ACC@1. The ACC@k accuracy is the metrics that often used in Information Retrieval, and the definition is:

$$ACC@k = \frac{1}{N}\sum_{i=1}^{N} 1[a_i \in C_i^k], \tag{31}$$

where a_i is the ground truth answer of question q_i, and C_i^k is the candidate answer set with the k highest similarity scores that the algorithm chooses. $1[\cdot] \rightarrow \{0,1\}$ is the indicate function, and the value is 1 if and of if the logical expression in the bracket is true.

C. Results

Table I illustrates the results of different types of deep neural networks on cMedQA dataset. The first column is the index, the second is vectors used in the model, including word embeddings and character embeddings, the third is the feature extraction model, and the last two column are the results of top-1 accuracy on development set and test set respectively.

Rows 1 to 3 present the results of single-level neural network using word embeddings. BiLSTMs model shows it superiority in capturing semantic information from word embeddings. Rows 4 to 8 describe the results of a multi-level neural networks using word embeddings. The best performance was achieved when we use 2-level Composite CNNs model and "biLSTMs+CNN" model. And compared with the composite model, the accuracy of the non composite model is about 3% lower. The 2-level composite CNNs model is better than the 3-level composite CNNs model, which shows that the 2-level composite CNNs model has been able to extract the semantic information of the word vector well.

Rows 9 to 11 illustrate the results of single-level neural networks using character embeddings. Zhang et.al.[1] has proved that multi-scale CNNs perform the best in single-level neural networks. Rows 12 to 16 show the results of multi-level neural network using character embeddings. The 3-level Composite CNNs model has the highest ACC@1 accuracy rate, followed by the 2-level Composite CNNs model. Rows 12 and 13 show the biLSTMs model at a low level can dig the semantic information of the text better. Compared with the non composite model, the composite model can greatly improve the accuracy of the model.

When the CNN layer is at a high level position in the model, it's good at extracting features. However, when it is at a low level, the ability to transmit information to the next level is poor if the composite method is not used. "biLSTMs+CNN" model is better than "CNN+ biLSTMs" model and "CNN+CNN" model both on word embeddings and character embeddings. The biLSTMs layer has a strong ability to transmit information from the low level layer to high level layer, because the output of the previous state in the sequence of LSTM is the hidden input of the next state. Therefore, it is better to use the biLSTMs layer as a low level model than the CNN model.

The multi-level composite model has better performance than single layer model. It shows that the multi-level model can extract the semantic information of the text better. In fact, multi-level models not only extract semantic information of

TABLE I
THE RESULTS OF DIFFERENT TYPES OF DEEP NEURAL NETWORK MODELS

Index	Vector	Model	Dev(%)	Test(%)
1	Word Embeddings	CNN	53.25	53.75
2		biLSTMs	56.15	57.65
3		Multi-scale CNNs	54.05	55.40
4		CNN+biLSTMs	55.60	57.35
5		biLSTMs+CNN	**56.95**	58.25
6		CNN+CNN	53.20	55.45
7		2-level Composite CNNs	56.30	**58.30**
8		3-level Composite CNNs	55.70	58.05
9	Character Embeddings	CNN	64.05	64.05
10		biLSTMs	61.65	63.20
11		Multi-scale CNNs	65.35	64.75
12		CNN+biLSTMs	62.30	62.70
13		biLSTMs+CNN	64.60	65.05
14		CNN+CNN	62.00	62.90
15		2-level Composite CNNs	65.35	65.80
16		3-level Composite CNNs	**65.60**	**66.15**

texts from depth, but also extract semantic information from different scales when extracting semantic information.

D. Experimental Analysis

In this subsection, we will analyze the number of the convolutional layers in different Multi-level Composite Convolutional Neural Networks. The model with more layers may be more accurate but it has more parameters which are very expensive to be trained. This subsection will find the most suitable hierarchical structure to achieve the balance between the number of parameters and the accuracy of the model.

The input of a convolutional layer depends on the output of its last convolutional layer, and the output of a convolutional layer is determined by the number of filters. If the number of filters is too large, the neural network may be over-fit. On the contrary, the neural network may be under-fit if it has small number of filters.

Table II illustrates the results of different convolution filters of Multi-Level Composite Convolutional Neural Network. The first column is multi-level composite convolutional neural networks with different levels and filters, and the lowest level is on the left. The scales of the convolutional filters listed in the table are all 3. The second and third columns are the results of top-1 accuracy on development set and test set respectively.

TABLE II
THE RESULTS OF MULTI-LEVEL COMPOSITE CNNS WITH DIFFERENT LEVELS AND FILTERS

Multi-level Composite CNNs	Dev(%)	Test(%)
(800,400)	65.35	65.80
(400,400)	64.45	65.45
(800,800)	65.25	65.15
(800,400,100)	**65.65**	65.05
(800,400,200)	65.05	64.25
(400,400,400)	65.60	**66.15**
(400,400,400,400)	64.80	65.55

As shown in Table II, the 3-level composite CNNs models have the highest ACC@1 accuracy rate. (400,400,400) model has the best performance on the test set. The best performance on the development set was achieved when we used (800,400,100) model. These models have good results both on the development set and the test set, which shows that the model is robust. The 4-level composite CNNs model is slightly worse than the 3-level composite models. Therefore, 3-level composite model reach the state-of-the-art when using the character embeddings.

It is noticed that even though multi-level composite CNNs framework has the same scale of the convolution of each level, the high-level neural network extracts the wider window of semantic information when extracting information from low level neural network. Therefore, the multi-level composite CNNs model contains the idea of multi-scale CNNs model. Therefore, the multi-level composite convolutional neural networks framework has showed its great ability in extracting the deep semantic information of the question and the answer.

V. CONCLUSION

The Multi-Level Composite Convolutional Neural Networks framework uses a multi-level structure can extract high-level dimensional semantic features. The multi-level architecture improve the feature extraction ability compared to single-level neural networks. Also, the composite mechanism make full use of the information from each layer and enrich the semantic representation of the question and the answer. The experiment illustrated in this paper shows the superiority of our model.

ACKNOWLEDGMENT

The work in this paper is nancially supported by National Natural Science Foundation of China under grant (No. 71331008) and (No. 61105124).

REFERENCES

[1] S. Zhang, X. Zhang, H. Wang, J. Cheng, P. Li, and Z. Ding, "Chinese medical question answer matching using end-to-end character-level multi-scale cnns," *Applied Sciences*, vol. 7, no. 8, 2017.

[2] T. Dodiya and S. Jain, "Question classification for medical domain question answering system," in *IEEE International Wie Conference on Electrical and Computer Engineering*, 2016, pp. 204–207.

[3] S. Jain and T. Dodiya, *Rule Based Architecture for Medical Question Answering System*. Springer India, 2014.
[4] A. B. Abacha and P. Zweigenbaum, "Medical question answering: translating medical questions into sparql queries," in *ACM Sighit International Health Informatics Symposium*, 2012, pp. 41–50.
[5] ——, "Means: A medical question-answering system combining nlp techniques and semantic web technologies," *Information Processing & Management*, vol. 51, no. 5, pp. 570–594, 2015.
[6] R. M. Terol and M. Palomar, "A knowledge based method for the medical question answering problem," *Computers in Biology & Medicine*, vol. 37, no. 10, pp. 1511–1521, 2007.
[7] T. Delbecque, P. Jacquemart, and P. Zweigenbaum, "Indexing umls semantic types for medical question-answering." *Stud Health Technol Inform*, vol. 116, pp. 805–810, 2005.
[8] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," vol. 3, pp. 2042–2050, 2015.
[9] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," in *Automatic Speech Recognition and Understanding*, 2016, pp. 813–820.
[10] M. Tan, C. D. Santos, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *Computer Science*, 2015.
[11] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 7, pp. 257–269, 2011.
[12] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, and P. Tucker, "Large scale distributed deep networks," in *International Conference on Neural Information Processing Systems*, 2012, pp. 1223–1231.
[13] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.