Hierarchical Multi-dimensional Attention Model for Answer Selection

Wei Liu*, Lei Zhang†, Longxuan Ma*, Pengfei Wang* and Feng Zhang‡
*School of Computer Science
Beijing University of Posts and Telecommunications, Beijing, China
Email: {lwabei,malongxuan,wangpengfei}@bupt.edu.cn
†Graduate School
Beijing University of Posts and Telecommunications, Beijing, China
Email: zlei@bupt.edu.cn
‡Information Science Academy
China Electronics Technology Group Corporation, Beijing, China
Email: feng3982315@163.com

***Abstract*—Answer selection is an important subtask of the question answering domain in natural language processing(NLP) applications. In this task, attention mechanism is a widely used technique which focuses on the context information and interrelationship between different words in the sentences to allocate different weight and enhance feature. However, the natural characteristics of words themselves are not fully excavated, thus the performance may be limited to a certain extent. In this paper, we propose a novel Hierarchical Multi-dimensional Attention (HMDA) model to address this issue. Especially, HMDA proposes a new kind of attention mechanism, word-attention, a true individual attention which can enhance the implied meaning of the word itself to extract features from word level which are more unique. Then HMDA uses global co-attention to better utilize word-attention and capture more common similar features. In order to utilize this attention-based semantic information on different granularities differently, HMDA designs a multi-layer structure which makes full use of all attention mechanisms by embedding attention features to model hierarchically. HMDA obtains various fine-grained information between question and candidate answers and avoids information loss. Empirically, we demonstrate that our proposed model can consistently outperform the state-of-the-art baselines under different evaluation metrics on all TrecQA, WikiQA and InsuranceQA datasets.**

I. Introduction

Answer selection task is to sort potential answer candidates according to relevance between question and answer. We give an example of a question with a positive answer and two negative answers extracted from TrecQA dataset in Table I. Acquiring the ability to rank is versatile and essential for many texts matching tasks, and serves as core function to more complex and sophisticated systems.

In recent years, the attention mechanism is widely applied in almost all aspects of NLP tasks. It significantly improves the performance by extracting only the most relevant information which is useful for the task [1] [2] [3]. Neural networks have achieved great success for texts matching systems [4] [5] [6] [7], and a wide assortment of neural ranking architectures have been proposed [5] [8] [9]. Improved attention models such as self-attention model [10] and co-attention mechanisms [11] [12] show efficiency in many tasks. Self-attention places weight on sequence itself, co-attention learns joint information with respect to both document and query, then distribute weight to both sides. These attention mechanisms focus on exploring the interrelationship between words in different contexts, thus assigning different weights to each word. They are essentially looking for the relationship between words and all other words in the sentences. However, there still lacks a kind of attention mechanism which focuses on word meaning itself. For example, verbs and nouns usually have more influence on sentence meaning, while conjunctions and prepositions have less influence on meaning. This information has nothing to do with the other words in the sentence, only from the word itself. Therefore, we think that the features extracted by the previous attention mechanism are insufficient, not exploit these potential information of words themselves fully.

TABLE I
Example of answer selection in TrecQA dataset.

Question	What is the primary symptom of a cataract?
Positive	Cataracts, a clouding of the lens in the eye, is the leading cause of blindness in China now.
Negative 1	Although two million cataract patients had their eyesight restored through surgery in the last decade, medical services are still greatly needed by patients.
Negative 2	”We have planned to provide 1.75 million cataracts surgeries in the 1996-2000 period,” said Chen yuan of the federation’s rehabilitation department.

We conduct experiments over three datasets. The empirical results demonstrate the effectiveness of our approach as compared with the state-of-the-art baseline methods. The main contributions of our work are as follows:

- We put forward a novel attention mechanism, denoted by word-attention, which is able to enhance the unique meaning of the word itself. Word-attention is capable of reflecting multiple features of word level, without any preprocessing additional feature statistics, such as part-

978-1-7281-2009-6/$31.00 ©2019 IEEE paper N-20008.pdf

of-speech tagging, named entity recognition, stem tagging and so on.

- We propose a novel hierarchical structure which can gradually focus the multiple attention mechanisms and extracts features from different granularities for better resolving answer selection task. We resort to co-attention as weight allocation method to capture the similarity between question and answer, adopt word-attention as feature augmenting tool to enhance the dissimilarity between words. The code for our model is available online[1].
- Experimental results on all TrecQA, WikiQA, InsuranceQA datasets outperform current work, which proves that HMDA achieves state-of-the-art performance without any external resource such as syntactic parser tree and additional lexicon features. We give an in-depth analysis of HMDA, explaining why word-attention effectively enhance the features of the word itself and illustrate the method to combine it with other attention mechanisms.

II. RELATED WORK

Answer selection is to rank the candidate answer list and choose the most relevant one. The dominant state-of-the-art models for learning to rank today are mostly neural network based models. Neural network models, such as convolutional neural networks (CNN) [13] [6] [14] [15], recurrent neural networks (RNN) [16] [17] [18] or recursive neural networks [19] are used for learning document representations. A parameterized function such as multi-layered perceptrons [5], tensor layers [20] or holographic layers [14] then learns a similarity score between document pairs.

Recent advances in neural ranking models go beyond independent representation learning. There are several main architectural paradigms that invoke interactions between document pairs which intuitively improve performance due to matching at a deeper and finer granularity. The first can be thought as extracting features from a constructed word-by-word similarity matrix [21] [22]. The second invokes matching across multiple views and perspectives [23] [12]. The third involves learning pairwise attention weights (i.e., co-attention). In these models, the similarity matrix is used to learn attention weights, learning to attend to each document based on its partner.

Attentive Pooling Networks [24] and Attentive Interactive Networks [25] are models that are grounded in this paradigm, utilizing extractive max-pooling to learn the relative importance of a word based on its maximum importance to all words in other documents. The Compare-Aggregate model [26] uses a co-attention model for matching and then a convolutional feature extractor for aggregating features. [27] firstly introduces attention mechanism into question answering under an RNN architecture. [28] proposes an attention-based model for word embedding, which calculates an attention weight for each word at each possible position in the context window. MAN [9] model uses multiple steps of attention. Different from those models and attention mechanisms, our proposed method uses a new word-attention mechanism focusing on extracting various potential features of word vector-representations and use it to perform an interactive operation with the co-attention result.

Notably, other related problem domains such as machine comprehension [29] [30] [11] and review-based recommendation [31] also extensively make use of co-attention mechanisms. In addition, learning sequence alignments via attention have been popularized by models in related problem domains such as natural language inference [32] [33] [34]. MCAN [35] can be viewed as an extension of the CAFE model proposed in [34] for natural language inference.

[1]https://github.com/malongxuan/MatchingSentencePair

III. OUR PROPOSED MODEL

The overall structure of HMDA is shown in Figure 1. The inputs to our model are two text sequences which denoted as question Q and answer A. Next we will portray this model in detail. For simplicity, we only describe the answer part of HMDA. Question part is exactly the same except the roles of A and Q exchange.

A. Embedding Layers

Firstly, we apply pre-trained 300 dimension Glove embedding [36] as input, denoted by $Q \in \mathbb{R}^{q*d}$ and $A \in \mathbb{R}^{a*d}$. d represents the 300 dimension, q and a represent the question and answer length which is fixed for batch data training.

B. Local Attention Layers

Previous methods to extract word level information are either not based on attention [37] [38] or still concerned with inter-sentence interaction information then concatenated to word level embedding [39]. The word-attention proposed in this paper is a real attention mechanism which only focuses on the information of words themselves. The calculation matrices perform operations on itself, only reserve the part which calculated on the same word embedding. The initial embedding is provided as the input of word-attention to collect local word level message. From different perspectives, we propose three methods to calculate word-attention, which are calculated as follows.

The first method will introduce an external parameter matrix which directly maps the results to a specified dimension. Where $W_1 \in \mathbb{R}^{d*a}$ is a parameter matrix to be learned. Every word embedding in A will be multiplied by a different column parameters in W_1. $diag$ means taking the diagonal element and changing the matrix of the $a * a$ dimension to the $a * 1$ dimension. $\cdot$ means the matrix multiplication.

$$O_1^A = diag(A \cdot W_1) \tag{1}$$

The second method is to directly calculate without introducing any external parameters. In this way, we only focus on the original embedded information.

$$O_2^A = diag(A \cdot A^T) \tag{2}$$

The third method also introduces external parameter matrix $W_2 \in \mathbb{R}^{d*d}$ help learning.

$$O_3^A = diag(A \cdot W_2 \cdot A^T) \tag{3}$$

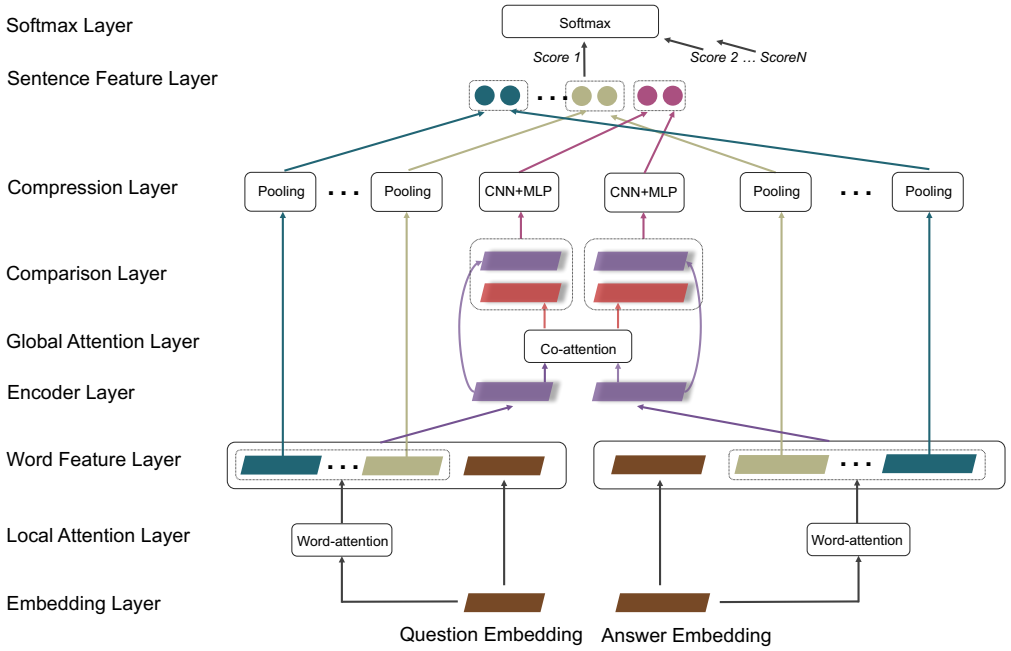

Fig. 1. Illustration of our proposed hierarchical multi-dimensional attention model. The local attention layer and the global attention layer extract the information of word level and sentence level respectively, and represent them in the word feature layer and the sentence feature layer.

Word attention is more focused and captures more unique characteristics of words themselves. Then, we adopt traditional softmax function to acquire attention weight matrix O^A, which merge word level information.

$$O^A = Softmax(O^A) \in \mathbb{R}^{a*1} \tag{4}$$

Where Softmax(.) is the Softmax operator.

C. Word Feature Layer

Next the following calculation is to match the result of word-attention with its original version to get new word feature embedding.

$$V^A = (O^A \otimes E) \odot A \in \mathbb{R}^{a*d} \tag{5}$$

Note that the outer product $O^A \otimes E$ is repeating the linearly transformed for d times. $\odot$ is the element-wise multiplication.

We use all three word-attention methods and express the results as V_1, V_2 and V_3. In order to fuse new word features and original embedding, here we can concatenate them either on the first dimension called $HMDA_{horizontal}$ or the second dimension called $HMDA_{vertical}$. Then we get the enhanced answer embedding $N^A \in \mathbb{R}^{4a*d}$ or $N^A \in \mathbb{R}^{a*4d}$.

$$N^A = concat[A, V_1^A, V_2^A, V_3^A] \tag{6}$$
$$N^Q = concat[Q, V_1^Q, V_2^Q, V_3^Q] \tag{7}$$

D. Encoder Layer

We use nonlinear functions to encode the enhanced embedding, denoted by H^A and H^Q. This encoding method has fewer parameters than usual method such as long short-term memory(LSTM) [40] and experiments show that they have comparable performance. Where σ is Sigmoid, Δ is Tanh.

$$H^Q = \sigma(N^Q) \odot \Delta(N^Q) \tag{8}$$
$$H^A = \sigma(N^A) \odot \Delta(N^A) \tag{9}$$

E. Global Attention Layer

After encoding the enhanced embedding, we have prepared input data for the global attention level to capture similar information between question and answer. We calculate co-attention matrices C^A and use it to weight the original embedding. We get the global sentence feature as follow:

$$C^A = H^A \cdot H^{Q^T} \tag{10}$$
$$R^A = Softmax(C^A) \cdot H^Q \tag{11}$$

F. Comparison Layer

Then we input $[H^A, R^A]$ into compare function which have several different forms [26]. The goal of the comparison layer is to match each word with its weighted version.

$$M^A = H^A \odot R^A \tag{12}$$

G. Compression, Sentence Feature Embed, Softmax Layers

As shown in Figure 1, we now input M^A to CNN+MLP aggregate, denote by $Z^A \in \mathbb{R}^{1*2d}$. Z^A represents the global semantic information of the entire sentence level. Then we

TABLE II
STATISTICS OF WIKIQA, TREC-QA AND INSURANCEQA DATASETS.

Dataset	Questions	Sentences	#Avg question	#Avg sentence	QA pairs	Avg candidate
InsuranceQA V1	17487	24981	7.16	49.5	1.36m	500
WikiQA	1242	29258	7.22	24.82	2.8k	9
TREC-QA(clean)	1295	8478	9.34	26.97	18.63 k	38

apply mean-max-pooling to V^A to compress the local word level feature. The final score is:

$$V_k^A = concat[mean(V_k^A), max(V_k^A)](k = 1, 2, 3) \quad (13)$$

$$V^A = concat[V_1^A, V_2^A, V_3^A] \in \mathbb{R}^{1*6d} \quad (14)$$

$$Score = concat[V^A, V^Q, Z^A, Z^Q] \cdot W_f \quad (15)$$

Where mean and max function extract maximum and average values along sentence length, W_f is a parameter matrix to be learned. We feed related answer set $[a_1, a_2, \ldots, a_N]$, target label set $[y_1, y_2, \ldots, y_N]$ along with one Q into the model:

$$Score_i = model[Q, a_i] \quad (16)$$

Lastly, we take KL-divergence loss to discriminatively train our framework. KL-loss is calculated as follow:

$$S_i = \frac{exp(Score_i)}{\sum_{t=1}^{N} exp(Score_t)} \quad (17)$$

$$Loss = \frac{1}{N} \sum_{1}^{N} KL(S \parallel Y) \quad (18)$$

When training with KL-divergence loss, we select all positive answers to this question, denote by p, then randomly select $N - p$ negative answers from the answer pool.

IV. EXPERIMENT

In this section, we introduce the WikiQA, TrecQA, InsuranceQA datasets, our common setup, and our experimental results.

A. Dataset

- WikiQA - This is a popular benchmark dataset for open-domain, factoid question answering. It is constructed by crowd-sourcing through sentences extraction from Wikipedia and Bing search logs proposed by [41]. We follow the same preprocessing steps as [41], where questions with no correct candidate answers are excluded. In total, we end up with 873 training questions, 126 development questions, and 243 test questions. Since negative answers for each question in WikiQA is not enough, we extend it by randomly selecting a set of negative candidates from the answer pool.
- TrecQA - This is a well-known benchmark dataset provided by [15]. This dataset contains a set of factoid questions, where candidate answers are limited to a single sentence. This dataset was collected from TrecQA tracks 8-13 and is comprised of factoid based questions which mainly answer the 'who', 'what', 'how', 'where', 'when' and 'why' types of questions. There are clean and raw versions [42]. For clean version, all questions have positive and negative answers simultaneously. In total, there are 1162 training questions, 65 development questions, and 68 test questions.
- InsuranceQA - This is an exclusive domain, non-factoid answering dataset proposed by [43], collected from a community question answering website which contains two versions. In this work, we use the V1 version of the dataset. This dataset is already divided into one training set, one validation set, and two testing sets, in which a question may have multiple correct answers and normally the questions are much shorter than the answers. The average length of questions and answers in tokens are 7.16 and 49.5 respectively. Such difference imposes additional challenges for the ranking task. For each question in the validate and test sets, there are 500 candidate answers, which include the ground-truth answers and randomly selected negative answers.

We illustrate the statistics of WikiQA, TrecQA and InsuranceQA V1 dataset in TableII. $\#Avg$ means the average length.

B. Compared Baselines

In this section, we introduce the baselines used for each dataset separately. The compared data in Table III are extracted from the original reference paper.

- WikiQA - Recently proposed MAN [9] apply multihop-sequential-LSTM to achieve step by step learning. The Pairwise Ranking MPCNN from [44] is effective. Other strong baselines include AP-BiLSTM and AP-CNN which are attentive pooling improvements of the former [24]. HyperQA [8] uses hyperbolic space for similarity analysis, reduce time and resource consumption. IARNN [3] employs inner attention within GRU proved effective. Compare-aggregate framework MULT [26] is impressive with matching sentence pairs.
- TrecQA - The important competitors on the dataset are mainly Multi-Perspective CNN (MP-CNN) [23]. We also compare with the pairwise ranking adaption of MP-CNN [44]. Additionally and due to the long-standing nature of this dataset, there have been a huge number of works based on traditional feature engineering approaches [45] [46] [15] [47]. For the clean version of this dataset, we also compare with AP-CNN just mentioned. LDC [48], BiMPM [12] achieves good performance. HyperQA [8] also proves its capacity to reduce resources occupancy. IWAN [17] gets good MAP score.

TABLE III
PERFORMANCE FOR ANSWER SENTENCE SELECTION ON WIKIQA, TREC-QA AND INSURANCEQA DATASETS.

Models	WikiQA		TrecQA(clean)		InsuranceQA	
	MAP	MRR	MAP	MRR	Acc(Test1)	Acc(Test2)
MPCNN (He et al.)	0.693	0.709	0.777	0.836	-	-
HyperQA (Yi et al.)	0.712	0.727	0.784	0.865	-	-
MPCNN + NCE (Rao et al.)	0.701	0.718	0.801	0.877	-	-
LDC Model (Wang et al.)	0.706	0.723	0.771	0.845	-	-
BiMPM (Wang et al.)	0.718	0.731	0.802	0.875	-	-
IWAN (Shen et al.)	0.733	0.750	**0.822**	0.889	-	-
AP-BiLSTM (Santos et al.)	0.671	0.684	0.713	0.803	0.717	0.664
AP-CNN (Santos et al.)	0.688	0.696	0.753	0.851	0.678	0.603
IARNN-Occam (Wang et al.)	0.734	0.741	0.727	0.819	0.689	0.651
IARNN-Gate (Wang et al.)	0.726	0.739	0.737	0.821	0.701	0.628
MULT (Wang and Jiang)	**0.743**	**0.754**	-	-	0.752	**0.734**
SUBMULT+NN(Wang and Jiang)	0.733	0.747	-	-	**0.756**	0.723
MAN(Tran and Niederee)	0.722	0.738	0.813	**0.893**	0.705	0.669
$HMDA_{reduced}$	0.744	0.753	0.811	0.890	0.748	0.722
$HMDA_{horizontal}$	0.759	0.769	**0.836**	0.910	**0.770**	0.740
$HMDA_{vertical}$	**0.763**	**0.776**	0.833	**0.919**	0.769	**0.748**

- InsuranceQA - The initial competitors of this dataset is the CNN-based architecture by [43]. APBiLSTM [24] is attentive pooling improvements of the former and Inner attention based RNN [3]. Most recently, multihop attention networks, MAN [9], which uses multiple vectors focusing on different parts of the question is proposed. Overall, Wang and Jiang [26] keep the highest record on this dataset with SUBMULT + NN [26] model.

C. Evaluation Protocol

Answer selection task is to rank the candidate answers using the trained model. Hence, we resort to Mean Average Precision (MAP), Mean Reciprocal Rank (MRR) and accuracy (Precision@1), standard metrics in Information Retrieval and Question Answering to measure the experimental results. We performed significant tests using the paired t-test. Differences are considered statistically significant when the p-value is lower than 0.05. We use the pre-trained 100/300 dimensional Glove vectors[2] proposed by [36] to initialize our word embedding layer. We fix the word representations during training for a fair comparison.

For WikiQA, we use 300 dimension Glove vectors. For concatenating, the hidden layer dimension is set to 300. We train our model in mini-batch and set the batch size with 11. To training our model in mini-batch, we truncate the question to 10 words and the answer to 40 words. If the sentence is shorter than the specified length, we add tokens with 0 embedding at the end of the sentence. We resort to Adam algorithm as the optimization method and update the parameters of the network with the learning rate as 0.001. We set two dropouts at the embedding layer and compression layer, the value is 0.1. We set a total answers sample as 15 and add L2 penalty with the coefficient parameter λ as 10^{-5}.

For TrecQA, there is only one different experiment setting, that is the maximum number of tokens for questions and answers. They are set to 15 and 60.

[2]https://nlp.stanford.edu/projects/glove/

For InsuranceQA, we use 100 dimension Glove vectors, set hidden size 280 for a fair comparison to former work, apply dropout of 0.2 to the compression layer, truncate the question to 12 words. Other settings are the same as WikiQA.

D. Experimental Results

In this section, we will introduce our experimental results and give detailed analysis.

- WikiQA - Table III reports our experimental results on WikiQA. HMDA outperforms a myriad of complex neural architectures. Notably, we obtain a clear performance gain of 2.2% by $HMDA_{vertical}$ in terms of MRR against strong models such as MULT.
- TrecQA - We compare against multiple previously published works on this dataset. The competitive baselines for this task are AP-CNN, LDC, MPCNN, MPCNN+NCE, HyperQA, BiMPM, MAN and IWAN. Experimental Results Table III reports our results on the clean version of TrecQA. The $HMDA_{vertical}$ model outperforms MAN by 2.6% on MRR.
- InsuranceQA V1 - Table III show the experimental results on InsuranceQA V1. All of our model outperform even the strongest model SUBMULT+NN [26]. $HMDA_{horizontal}$ gains a promotion of 1.4% and 1.7% in term of accuracy in test1 and test2.

The results on WikiQA and TrecQA show that our model also suitable for factoid, short paragraphs answer selection task. The experiments on InsuranceQA V1 demonstrate that our proposed model can improve long paragraphs, non-factoid, multiple candidate question answering task. On all datasets, our model outperforms current models.

V. DISCUSSION AND ANALYSIS

In this chapter, we analyze the model in detail from multiple angles.

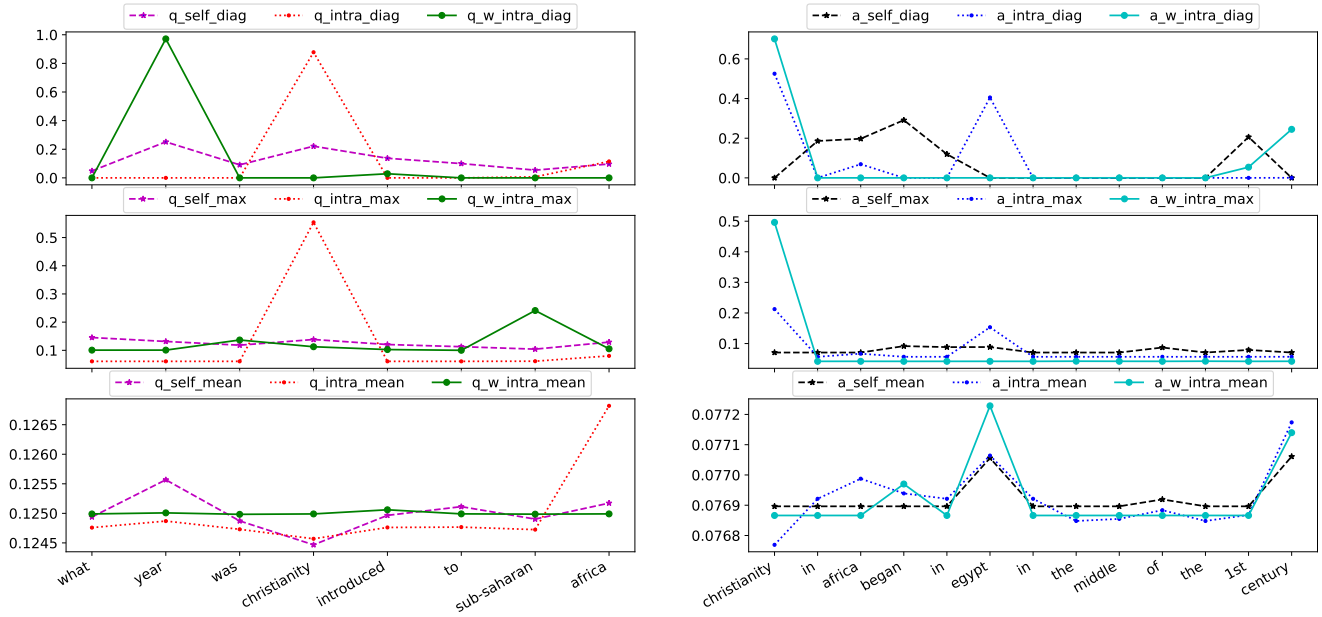

Fig. 2. The attention matrix in local attention layer is taken max pooling, mean pooling, and diagonal compared. Expression with "_self_" is the result calculated by the first method, while with "_intra_" suffix is the second and with "_w_intra_" is the third.

TABLE IV
ABLATION ANALYSIS ON TRECQA(CLEAN) TEST SET.

Setting	MAP	MRR
Original	0.833	0.919
(1)Without Encoder	0.790	0.861
(2)Without Global-attention	0.655	0.730
(3)Without Word-attention	0.811	0.890
(4)Without Comparison Layer	0.801	0.868

A. Ablation Analysis

This section shows the relative validity of the different components of our HMDA model. Table IV presents the results on the test set of the clean TrecQA dataset. From ablation analysis, we can easily observe the relative functions of various components to our model. We introduce four different model structures.

(1) We remove the encoder before global attention layer, input original embedding to global attention and comparison layer. The influence is significantly large, cause MAP drop by 4.3%. Due to the lack of contextual information of encoding process integration, the system greatly reduces efficiency.

(2) We abandon the global-attention layer before the comparison layer. As we expected, it is the core component of the entire system, and the lack of it would dramatically reduce the functionality of the entire system. The MAP and MRR reduce more than 17%.

(3) We remove word-attention before word feature layer. The performance of the model decrease obviously by 2.2% and 2.9% respectively which prove the efficiency of word-attention.

(4) We also propose a model without comparison layer. With performance decreasing obviously, it proves comparison layer is an indispensable component of the model.

B. In-depth Model Analysis

We use a question-answer pair in InsuranceQA dataset to perform the analysis. Word-attention mechanisms can be seen as taking the diagonal value of the original comparison matrix. We also compare the results of max-pooling and mean-pooling operation. From Figure 2, their general trend is similar, but the range of ordinate changes varies greatly. The mean-pooling operation pays attention to all the words, however, the features are not prominent enough. The weight distribution interval is less than 0.005. The max-pooling operation increases the weight assignment difference between features. Taking the diagonal operation, that is, word-attention, makes the keyword features more prominent. The difference interval has been expanded to 1. This means that some features are strongly enhanced. We can also see the difference among three word-attention calculation methods in this figure. The trend curve calculated by the first method is relatively flat. The second method is similar to the third method, and the calculated results highlight the importance of certain words. But because the third method additionally introduces an additional parameter to help with learning, in the first sentence, the second method emphasizes the "christianity" word, while the third method significantly increases the importance weight of the "year". It is easy to understand that these two words are both nouns and obviously have reasons for being valued. Therefore, as the model describes, we combine the calculations of different methods to capture as much information as possible.

Then we further compare the effects of enhanced and reduced input embedding on global feature representation. When we choose to concatenate the word feature embedding

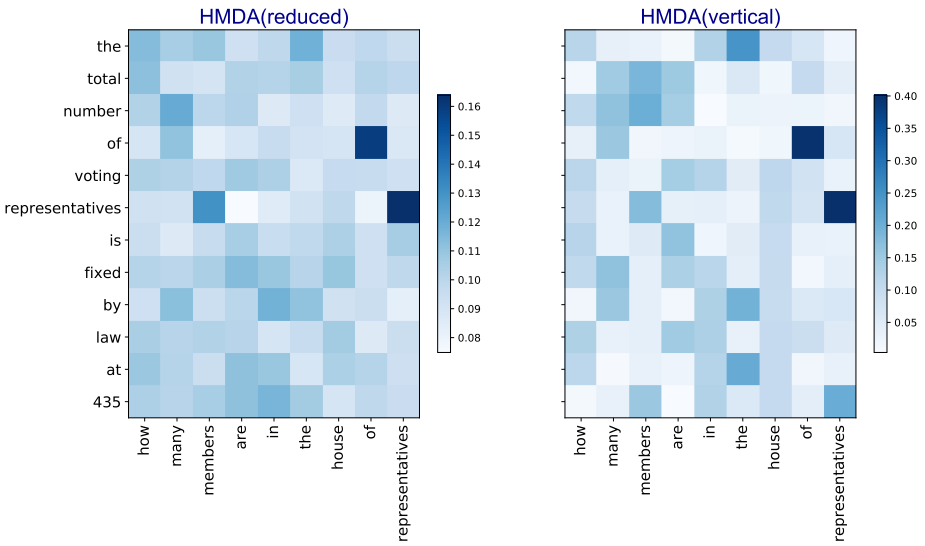

Fig. 3. Global attention weights with reduced and enhanced input embedding.

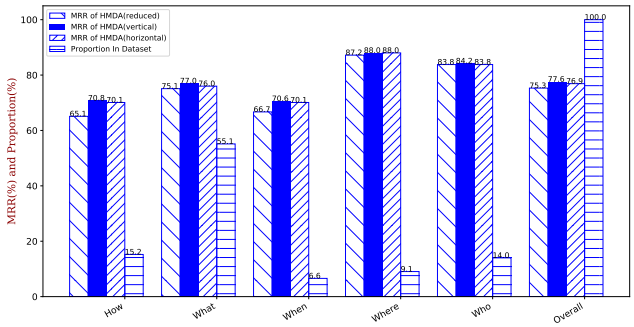

Fig. 4. Comparison between reduced and enhanced features on type of questions.

from the vertical direction, as shown in Figure 3, we use a heat map to visualize the global attention weights. The image on the left is the result of an original embedding input while the right one's input is the enhanced embedding with feature enhanced by word-attention as depicted in HMDA. For example, we can clearly see that the interaction weight of "435" with "members" and "representations" is more focused. The weight gap becomes larger among words.

Finally, we analyze the performance of the model for different types of problems. Figure 4 demonstrates all five types of questions in WikiQA test set. The histogram represents the MRR metric and the proportion of each type of questions in dataset respectively. We use $HMDA_{reduced}$, $HMDA_{vertical}$ and $HMDA_{horizontal}$ to compare the difference. Because locations and characters are easier to retrieve, all models own better results for "Where" and "Who" questions. In $HMDA_{reduced}$ model, the MRR value of 87.2% and 83.8% are respectively achieved for "Where" and "Who" questions. While due to the proportion of 55.1% and 15.2%, the effect on "What" and "How" questions decide the overall performance of the model. In $HMDA_{reduced}$ model, the MRR value 75.1% and 65.1% are respectively achieved for "What" and "How" questions. After adding enhanced features by word-attention, we find that $HMDA_{vertical}$ model improves the MRR value by 1.9% on "What" questions, polish up the MRR value of How question by 5.7%. "What" and "How" problems have increased more than the "Where" and "Who" problems, which shows that the HMDA model successfully improves the comprehension ability of complex semantic relations.

VI. CONCLUSION

It is very crucial to improve the performance of answer selection for satisfying the demands of the industry. We propose a new attention mechanism, word-attention and a novel model HMDA for ranking tasks in question answering domains which takes all advantage of multiple attention mechanisms hierarchically. Word-attention focuses on enhancing the unique features implied in the word itself without too much contextual information. All possible valid word features are extracted through a simple layer of attention calculation without any additional data statistical preprocessing. HMDA integrates multiple attention application scenarios to allocate weight and augment feature on different granularities hierarchically. Although we demonstrate experimental results for answer selection task only, this method suit for much more diverse types of NLP tasks. Experimental results illustrate our model achieves state-of-the-art performance on multiple well-known benchmark datasets.

ACKNOWLEDGMENT

This research work was supported by the National Natural Science Foundation of China under Grant No.61802029, the fundamental Research for the Central Universities under Grant No.500418800. We also thank the reviewers for their valuable suggestions.

REFERENCES

[1] T. Rocktaschel, E. Grefenstette, K. M. Hermann, T. K. Isk, and P. Blunsom, "Reasoning about entailment with neural attention," *international conference on learning representations*, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *international conference on learning representations*, 2015.

[3] B. Wang, K. Liu, and J. Zhao, "Inner attention based recurrent neural networks for answer selection," in *ACL (1)*. The Association for Computer Linguistics, 2016.

[4] H. He and J. J. Lin, "Pairwise word interaction modeling with deep neural networks for semantic similarity measurement," in *HLT-NAACL*. The Association for Computational Linguistics, 2016, pp. 937–948.

[5] A. Severyn and A. Moschitti, "Learning to rank short text pairs with convolutional deep neural networks," in *SIGIR*. ACM, 2015, pp. 373–382.

[6] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *CIKM*. ACM, 2014, pp. 101–110.

[7] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "anmm: Ranking short answer texts with attention-based neural matching model," *conference on information and knowledge management*, pp. 287–296, 2016.

[8] Y. Tay, L. A. Tuan, and S. C. Hui, "Hyperbolic representation learning for fast and efficient neural question answering," in *WSDM*. ACM, 2018, pp. 583–591.

[9] N. K. Tran and C. Niederée, "Multihop attention networks for question answer matching," in *SIGIR*. ACM, 2018, pp. 325–334.

[10] A. Vaswani, N. Shazeer, N. Parmar, L. Jones, J. Uszkoreit, A. N. Gomez, and . Kaiser, "Attention is all you need," *neural information processing systems*, pp. 5998–6008, 2017.

[11] C. Xiong, V. Zhong, and R. Socher, "Dynamic coattention networks for question answering," *international conference on learning representations*, 2017.

[12] Z. Wang, W. Hamza, and R. Florian, "Bilateral multi-perspective matching for natural language sentences," in *IJCAI*. ijcai.org, 2017, pp. 4144–4150.

[13] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *neural information processing systems*, pp. 2042–2050, 2014.

[14] Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui, "Learning to rank question answer pairs with holographic dual lstm architecture," *international acm sigir conference on research and development in information retrieval*, pp. 695–704, 2017.

[15] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? A quasi-synchronous grammar for QA," in *EMNLP-CoNLL*. ACL, 2007, pp. 22–32.

[16] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *AAAI*. AAAI Press, 2016, pp. 2786–2792.

[17] G. Shen, Y. Yang, and Z. Deng, "Inter-weighted alignment network for sentence pair modeling," in *EMNLP*. Association for Computational Linguistics, 2017, pp. 1179–1189.

[18] Y. Wu, W. Wu, and Z. Li, "Knowledge enhanced hybrid neural network for text matching," *national conference on artificial intelligence*, 2018.

[19] S. Wan, Y. Lan, J. Xu, J. Guo, L. Pang, and X. Cheng, "Match-srnn: modeling the recursive matching structure with spatial rnn," *international joint conference on artificial intelligence*, pp. 2922–2928, 2016.

[20] X. Qiu and X. Huang, "Convolutional neural tensor network architecture for community-based question answering," pp. 1305–1311, 2015.

[21] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," *national conference on artificial intelligence*, pp. 2793–2799, 2016.

[22] S. Wan, Y. Lan, J. Guo, J. Xu, L. Pang, and X. Cheng, "A deep architecture for semantic matching with multiple positional sentence representations," *national conference on artificial intelligence*, pp. 2835–2841, 2016.

[23] H. He, K. Gimpel, and J. J. Lin, "Multi-perspective sentence similarity modeling with convolutional neural networks," pp. 1576–1586, 2015.

[24] C. N. dos Santos, M. Tan, B. Xiang, and B. Zhou, "Attentive pooling networks," *CoRR*, vol. abs/1602.03609, 2016.

[25] X. Zhang, S. Li, L. Sha, and H. Wang, "Attentive interactive neural networks for answer selection in community question answering," in *AAAI*. AAAI Press, 2017, pp. 3525–3531.

[26] S. Wang and J. Jiang, "A compare-aggregate model for matching text sequences," *international conference on learning representations*, p. 1, 2017.

[27] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *NIPS*, 2015, pp. 1693–1701.

[28] W. Ling, Y. Tsvetkov, S. Amir, R. Fermandez, C. Dyer, A. W. Black, I. Trancoso, and C. C. Lin, "Not all contexts are created equal: Better word representations with variable attention," in *Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1367–1372.

[29] S. Wang and J. Jiang, "Machine comprehension using match-lstm and answer pointer," *CoRR*, vol. abs/1608.07905, 2016.

[30] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *international conference on learning representations*, 2017.

[31] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," *knowledge discovery and data mining*, 2018.

[32] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, and D. Inkpen, "Enhanced lstm for natural language inference," *meeting of the association for computational linguistics*, vol. 1, pp. 1657–1668, 2017.

[33] A. P. Parikh, O. Tackstrom, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," *empirical methods in natural language processing*, pp. 2249–2255, 2016.

[34] Y. Tay, L. A. Tuan, and S. C. Hui, "A compare-propagate architecture with alignment factorization for natural language inference." *arXiv: Computation and Language*, 2018.

[35] T. Yi, T. Luu Anh, and H. Siu Cheung, "Multi-cast attention networks for retrieval-based question answering and response prediction," *CoRR*, vol. abs/1806.00778, 2018.

[36] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*. ACL, 2014, pp. 1532–1543.

[37] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 2017, pp. 1870–1879.

[38] M. Hu, Y. Peng, Z. Huang, X. Qiu, and M. Zhou, "Reinforced mnemonic reader for machine reading comprehension," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, 2018, pp. 4099–4106.

[39] H. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," *CoRR*, vol. abs/1711.07341, 2017.

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] Y. Yang, W. Yih, and C. Meek, "Wikiqa: A challenge dataset for open-domain question answering," in *EMNLP*. The Association for Computational Linguistics, 2015, pp. 2013–2018.

[42] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *EMNLP*. The Association for Computational Linguistics, 2015, pp. 379–389.

[43] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," *ieee automatic speech recognition and understanding workshop*, pp. 813–820, 2015.

[44] J. Rao, H. He, and J. J. Lin, "Noise-contrastive estimation for answer selection with deep neural networks," in *CIKM*. ACM, 2016, pp. 1913–1916.

[45] M. Heilman and N. A. Smith, "Tree edit models for recognizing textual entailments, paraphrases, and answers to questions," in *HLT-NAACL*. The Association for Computational Linguistics, 2010, pp. 1011–1019.

[46] A. Severyn, A. Moschitti, M. Tsagkias, R. Berendsen, and M. de Rijke, "A syntax-aware re-ranker for microblog retrieval," in *SIGIR*. ACM, 2014, pp. 1067–1070.

[47] X. Yao, B. V. Durme, C. Callison-Burch, and P. Clark, "Answer extraction as sequence tagging with tree edit distance," in *HLT-NAACL*. The Association for Computational Linguistics, 2013, pp. 858–867.

[48] Z. Wang, H. Mi, and A. Ittycheriah, "Sentence similarity learning by lexical decomposition and composition," in *COLING*. ACL, 2016, pp. 1340–1349.