Encoder-Decoder Network with Cross-Match Mechanism for Answer Selection

Zhengwen Xie, Xiao Yuan, Jiawei Wang and Shenggen Ju*

College of Computer Science, Sichuan University, Chengdu 610065, China
jsg@scu.edu.cn

Abstract. Answer selection (AS) is an important subtask of question answering (QA) that aims to choose the most suitable answer from a list of candidate answers. Existing AS models usually explored the single-scale sentence matching, whereas a sentence might contain semantic information at different scales, e.g. Word-level, Phrase-level, or the whole sentence. In addition, these models typically use fixed-size feature vectors to represent questions and answers, which may cause information loss when questions or answers are too long. To address these issues, we propose an Encoder-Decoder Network with Cross-Match Mechanism (EDCMN) where questions and answers that represented by feature vectors with fixed-size and dynamic-size are applied for multiple-perspective matching. In this model, Encoder layer is based on the "Siamese" network and Decoder layer is based on the "matching-aggregation" network. We evaluate our model on two tasks: Answer Selection and Textual Entailment. Experimental results show the effectiveness of our model, which achieves the state-of-the-art performance on WikiQA dataset.

Keywords: Answer Selection, Multi-Perspective, Cross-Match Mechanism.

1 Introduction

Answer selection is an important subtask of question answering (QA) that enables choosing the most suitable answer from a list of candidate answers in regards to the input question. In general, a good answer has two characteristics: First, the question affects the answer, therefore the good answer must be related to the question; Second, the good answer does not require strict word matching, but show better semantic relevance. These characteristics consequently make traditional feature engineering techniques less effective compared to neural models [5,13,16,26].

Previous neural models can be divided into two kinds of frameworks. The first one is the "Siamese" network [17,27]. It usually utilizes either Recurrent Neural Network (RNN) or Convolutional Neural Networks (CNN) to generates sentence representations, and then calculate similarity score solely based on the two sentence vectors for the final prediction. The second one is called "matching-aggregation" network [14,23], it matches two sentences at Word-level by the fixed-size feature vector, and then aggregates the matching results to generate a final vector for prediction. Studies on benchmark QA datasets show that the second one performs better [23].

Despite the second framework has made considerable success on the answer selection task, there are still some limitations. First, [14,24] only explored the single-scale matching, whereas a sentence might contain semantic information at different scales, e.g. Word-level, Phrase-level, or the whole sentence. To deal with this issue, [23] applied the multi-perspective matching technique to match two sentences representations at Word-level and Sentence-level, but ignored the Phrase-level information. Second, [18,19,23] just aggregate questions and answers which have different lengths into fixed-size feature vectors for matching, which may lose a large amount of rich information contained in sentence compared to matching with feature vectors with dynamic fixed-size.

To address these issues, we propose an Encoder-Decoder Network with Cross-Match Mechanism (EDCMN) for answer selection. Our model is a new framework where the Encoder layer is based on the "Siamese" network and the Decoder layer is based on the "matching-aggregation" network. We first obtain three semantic representations which capture coarse-to-fine information including Sentence-level, Phrase-level and Word-level in the Encoder layer. After that, to get the interaction between QA pairs, we utilize three feature augmentation methods to match both on feature vectors with dynamic-size and fixed-size. And then we obtain six matching vectors and have a concatenation on them. Finally, we compress the concatenation vector to generate a final vector for prediction. The Cross-Match Mechanism in our Encoder-Decoder framework enables capturing multiple perspective information which is suitable to identify complex relations between questions and the answers.

The main contributions of our paper can be summarized as follows:

- We propose an Encoder-Decoder Network with Cross-Match Mechanism which based on two classical frameworks. Our model is the first to apply Encoder-Decoder architecture on the answer selection task, and it requires no additional information and relies solely on the original text.
- The Cross-Match Mechanism in EDCMN captures information of sentence at different scales and matching on multiple perspectives.
- In comparison to other state-of-the-art representation learning approaches with attention, our approach achieves the best results and significantly outperforms various strong baselines.

2 Related work

Answer selection (AS) has been studied for many years. The previous work focused on designing hand-craft features to capture n-gram overlapping, word reordering and syntactic alignments phenomena [6,25]; This kind of method can work well on a specific task or dataset, but it's hard to generalize well to other tasks.

Recently, researchers started using deep neural networks for answer selection, the first kind of framework is based the "Siamese" architecture [4,11,17,20,27], In this framework, the same neural network encoder (e.g., a CNN or an RNN) is applied to two input sentences individually. However, there is no explicit interaction between the two sentences during the encoding procedure. The second kind of framework is called

"matching-aggregation" model [14,23]. Under this framework, smaller units of the two sentences are firstly matched, and then the matching results are aggregated into a vector to make the final decision. It captures interactive features between two sentences at Word-level, but ignored other granular matchings.

[11] uses self-attention to focus on the important parts of the sentence, but ignore the interaction between sentence pairs. [1,17] apply additive attention to solve the problem of lack of interaction between question and answer and this approach only gets the coarse-grained information. [12,24,27] employ models which form an interactive Word-level matrix of questions and answers to discover fine-grained alignment of two sentences, whereas it only used one matching method. [14,19] utilize multiple attentions to focus on different parts of the semantic representation, then compress them into fixed-size feature vectors for matching, which may lose rich information in dynamic-size feature vectors.

Our work is also inspired by the idea of Multi-Perspective Matching models [19,23] in NLP. We propose the EDCMN model which achieve step by step learning. We are the first to apply Encoder-Decoder architecture to the answer selection task. Our model not only employs the Multi-Perspective Matching model to identify complex relations between questions and answers but also captures information of sentence at different scales in Encoder which results in a more close match in the Decoder layer.

3 Model

We introduce EDCMN and its detailed implementation in this section. We first cover the model architecture and the Cross-Match Mechanism which is the core innovation in this paper, in Section **3.1.** We then introduce the Encoder layer and Decoder layer in Section **3.2** and **3.3**, respectively.

3.1 Cross-Match Mechanism

The Cross-Match Mechanism is inspired by ResNet [28] which reveals that the information at different scales can be extracted as the network get deeper, in addition, hierarchical information can be combined when it crosses different levels. As shown in Fig. 1, the cross-matching mechanism is mainly composed of an encoder layer and a decoder layer. The encoder layer obtains sentence representations at Word-level, Phrase-level and Sentence-level by LSTM, CNN and Self-attention components respectively. The decoder layer first applies Match functions to get six interaction representations about QA pairs, and then merge them to the final vector with the Compress function.

The biggest benefit of the Cross-Match Mechanism is that it realizes short-circuit connection through concatenate feature, which makes some of the features extracted from earlier layers may still be used directly by deeper layers, meanwhile, it can match vectors in multiple ways.

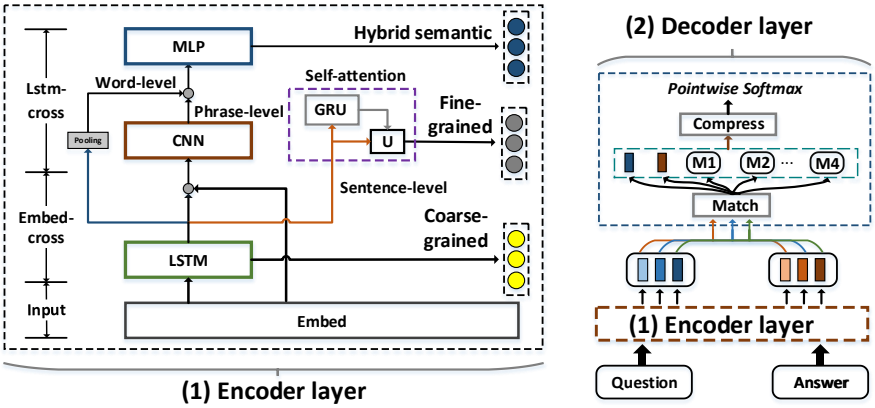

Fig. 1. The Architecture for Encoder-Decoder Network with Cross-Match Mechanism (EDCMN), The left-hand side shows the details about the Encoder layer.

3.2 Encoder layer

Embed Cross. With pre-trained d dimension word embedding, we can obtain sentence representations $H_q = [s_q^1, ..., s_q^m]$ and $H_a = [s_a^1, ..., s_a^n]$ where $s_q^i \in \mathbb{R}^d$ is the embedding of the i-th word in the sentence H_q. m and n are the lengths of H_q and H_a respectively. The model obtains the context information of the two vectors entered through the bidirectional Long Short-Term Memory (LSTM) [2] Network.

After the Bi-LSTM, the model obtains the sentence representation of the question $T_q = \{t_q^1, ..., t_q^m\}$ and $T_a = \{t_a^1, ..., t_a^n\}$ respectively. The coarse-grained semantic representations of QA pairs are obtained.

$$R_Q^1 = \{t_q^1, ..., t_q^m\} \tag{1}$$

$$R_A^1 = \{t_a^1, ..., t_a^n\} \tag{2}$$

Where $R_Q^1 \in \mathbb{R}^{m \times 2d}$, $R_A^1 \in \mathbb{R}^{n \times 2d}$, d is the dimension of pre-trained word embedding.

In order to make full use of the information of word embedding, the model has a concatenation on Bi-LSTM's output vectors and Embed Cross vectors. We apply this method to the question vector and then obtain a splicing vector $Cross_Q^1 \in \mathbb{R}^{m \times 3d}$. In the same way, the answer vector $Cross_A^1 \in \mathbb{R}^{n \times 3d}$ is got.

LSTM Cross. This component consists of two parts: Fine-grained representations and Hybrid semantic representations.

Fine-grained representations. We first create an encoding of the importance for each segment in the unpooled representation T which is the output of the Bi-LSTM by

applying an additional, separate Bi-GRU and obtain the concatenated output states $P \in \mathbb{R}^{l \times 2d}$ of this Bi-GRU where the *i*-th row in P. We then reduce each row P_i to a scalar v_i and apply *softmax* on the vector v to obtain scaled importance values :

$$P = BiGRU(T) \tag{3}$$

$$v_i = w^T P_i \tag{4}$$

$$\alpha = soft\max(v) \tag{5}$$

Where $w \in \mathbb{R}^{2d}$, are learned network parameters for the reduction operation, $v_i \in \mathbb{R}$ is the (unscaled) importance value of the *i*-th segment in P, and $\alpha \in \mathbb{R}^l$ is the attention vector. as mentioned before, self-attention [9,15] can solve the long-range dependency problem and choose the relevant information for sentence semantic. by utilizing this operation, the model can grasp the most relevant parts for inference relation precisely and make the correct decision. Unlike before, we retain the length of the sentence to get more fine-grained information. Finally, the sentence-level representation U according to our importance vector α is obtained:

$$u_{i,j} = \alpha P_{i,j} \tag{6}$$

$$R_Q^2 = SelfAttention_Q = U^Q \tag{7}$$

$$R_A^2 = SelfAttention_A = U^A \tag{8}$$

Hybrid semantic representations. From the perspective of a word, each word itself may have many meanings, but a phrase made up of many words makes the word less ambiguous. For example, it's difficult to know whether the word "apple" refers to a fruit or a company. If it is the phrase "apple company", we can clearly know the true meaning of apple. Inspired by this observation, the model utilizes 1-D convolutions (Conv1D) with multiple-windows to capture different Phrase-level information in sentences, then apply the max-pooling operation to select the most significant properties in the sentence.

$$CNN_{\max}^i = Conv1D_{Max-pooling}(Cross^1) \tag{9}$$

$$CNN_{feature} = [CNN_{\max}^1, ..., CNN_{\max}^k] \tag{10}$$

Therefore, we can obtain $CNN_Q \in \mathbb{R}^{ku}$ and $CNN_A \in \mathbb{R}^{ku}$, which extract the phrase structure information from questions and answers. k is the number of windows and u is the number of filters.

Furthermore, to take Word-level semantic into consideration, we first utilize the concatenation on the Conv1D feature $CNN_{feature} \in \mathbb{R}^{ku}$ and LSTM cross feature

$Cross^2 \in \mathbb{R}^{2d}$. Then, we apply a multi-layer perceptron (MLP) with *relu* activation function on concatenation vectors to get the hybrid semantic representations $R_Q^3 \in \mathbb{R}^d$ and $R_A^3 \in \mathbb{R}^d$ which denotes the hierarchical semantic fusion in questions and answers separately.

$$Cross^2 = Max(t^1, ..., t^l) \tag{11}$$

$$Hybird_{feature} = [Cross^2; CNN_{feature}] \tag{12}$$

$$R_Q^3 = R_A^3 = MLP_1(Hybird_{feature}) \tag{13}$$

Where $[.;.]$ represents the concatenation operation, l is the length of the sentence.

3.3 Decoder layer

Match & Interaction. After getting the multi-granularity feature vectors, we employ three different ways to explore the multi-scale matching.
Vectors with dynamic-size. For coarse-grained semantic representations and fine-grained semantic representations, their vectors both with dynamic-size. We construct a similarity matrix to further match the similarity word by word. Then, the similarity matrix representation is transformed into high-dimensional vector which hidden dimensions is $m \times n$. At last, we convert it into the compressed vector. This method can make the model more robust, because dimensions of the vector get higher as sentence gets longer. Let f denote a match function that matching two semantic representations in different ways, $M_1 \in \mathbb{R}^d$ and $M_2 \in \mathbb{R}^d$ are as follows:

$$M_1 = f(R_Q^1, R_A^1) = relu(W_1(R_Q^1 \otimes (R_A^1)^T) + b_1) \tag{14}$$

$$M_2 = f(R_Q^2, R_A^2) = relu(W_2(R_Q^2 \otimes (R_A^2)^T) + b_2) \tag{15}$$

Vectors with fixed-size. For the hybrid semantic vector, in order to measure the gap between the question representation and answer representation, a direct strategy is to compute the absolute value of their difference. $M_3 \in \mathbb{R}^d$ is approximating the Euclidean distance between the two vectors. Then, their cosine distance $M_4 \in \mathbb{R}^d$ is calculated by the element-wise product, $\odot$ means element-wise product:

$$M_3 = f(R_Q^3, R_A^3) = R_Q^3 - R_A^3 \tag{16}$$

$$M_4 = f(R_Q^3, R_A^3) = R_Q^3 \odot R_A^3 \tag{17}$$

The hybrid hierarchical semantic vectors R_Q^3 and R_A^3 without matching directly are also as outputs, which can preserve the semantic integrity to the greatest and make the network more robust.

Compression & Label Prediction. To be specific, we concatenate those six matching vectors by rows, then MLP with activation function are applied to them to calculate the probability distribution of the matching relation between the QA pair. The final compression vector $Score \in \mathbb{R}^{6d}$ and the output of this layers are as follows:

$$Score = MLP_2([R_Q^3; R_A^3; M_1; M_2; M_3; M_4]) \tag{18}$$

$$P(y \mid H_q, H_a) = Soft\max(Score) \tag{19}$$

We regard the answer selection task as the binary classification problem and the training objective is to minimize the negative loglikelihood in training stage:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \log P(y \mid H_q, H_a) + R \tag{20}$$

Where y_i is the one-hot representation for the true class of the *i*-th instance, N represents the number of training instances, and $R = \lambda \| \theta \|_2^2$ is the L2 regularization.

4 Experimental Setup

4.1 Datasets and Evaluation Metric

Table 1. The statistics of Answer Selection datasets and Textual Entailment datasets.

Dataset	WikiQA	TREC-QA	SCITAIL
# of questions/premise (train/dev/test)	873/126/243	1162/65/68	1542/121/171
Avg length of questions/premise	6	8	11
Avg length of answers/hypothesis	25	28	7
Avg # of candidate answers	9	38	-

Answer Selection task. For answer selection task, we experiment on two benchmark datasets. We evaluate models by mean average precision (MAP) and mean reciprocal rank (MRR). Statistical information of QA datasets is shown in Table 1.

WikiQA [10] is a recent popular benchmark dataset for open-domain question answering, based on factual questions from Wikipedia and Bing search logs. each question is selected from Wikipedia and used sentences in the summary paragraph as candidates.

TREC-QA [21] is a well-known benchmark dataset collected from the TREC Question Answering tracks. The dataset contains a set of factoid questions, where candidate answers are limited to a single sentence.

Textual Entailment task. For textual entailment task, we experiment on one new dataset that is created from a QA task rather than sentences authored specifically for the

entailment task. which is more challenging. Following previous works [7], we use accuracy (ACC) as evaluation metrics. Statistical information of the dataset is shown in Table 1.

SCITAIL [7] is the first entailment set that hypotheses from science questions and the corresponding answer candidates, and premises from relevant web sentences retrieved from a large corpus. combined with the high lexical similarity of premise and hypothesis for both entailed and non-entailed pairs, makes this new entailment task particularly difficult.

4.2 Implementation Details

We initialized word embedding with 300d-GloVe vectors pre-trained from the 840B Common Crawl corpus [8], while the word embeddings for the out-of-vocabulary words were initialized randomly.

WikiQA: To train our model in mini-batch, we truncate the question to 25 words, the answer to 90 words and batch size to 32. We add 0 at the end of the sentence if it is shorter than the specified length. We resort to Adam algorithm as the optimization method and update the parameters with the learning rate as 0.001. The CNN windows are [1, 2, 3, 4, 5, 6] and the number of CNN filters is 300. We set a dropout rate as 0.5 at the encoder layer and 0.7 at the decoder layer. We add the L2 penalty with the parameter λ as 10^{-5}.

TREC-QA: The experiment settings are the same as WikiQA.

SCITAIL: We truncate both questions and answers to 20 words and change the number of CNN filters to 100, The other experiment settings are the same as WikiQA.

4.3 Experimental Results

WikiQA and TREC-QA. Table 2 reports our experimental results and compared models on these two datasets. We selected 9 models for comparison. On WikiQA, our model achieves state-of-the-art performance. More specifically, Compared with MAN [19] which matching only on fixed-size vectors, our EDCMN model obtains a fine improvement (1.8%) by achieving 74% in MAP, and we outperform it by 1.4% on MRR. Both BiMPM [23] and our model benefit from Multi-Perspective Matching, however, EDCMN is better than BiMPM by 2.2% in terms of MAP and 2.1% in terms of MRR. The reason is that we take Phrase-level information into consideration, but BiMPM ignored it. On TREC-QA (clean), Our EDCMN model is better than most strong baselines such as ABCNN [26], LDC [23], MAN [18], which further indicates that our Cross-Match Mechanism is very effective for matching vectors.

SCITAIL. Table 3 presents the results of our models and the previous models on this dataset. We compare the results reported in the original paper [7]: Majority class, n-gram, decomposable attention, ESIM, and DGEM. Compared with ESIM [3] which had got a huge success on the NLI task, our EDCMN model outperforms 7.6% in test

accuracy. We also obtain a fine improvement by achieving 78.2% contrast to the DGEM in test accuracy. This ascertains the effectiveness of the EDCMN model.

Table 2. Performance for answer sentence selection on WikiQA and TREC-QA test set.

Model	WikiQA		TREC-QA (clean)	
	MAP	MRR	MAP	MRR
AP-BiLSTM [12]	0.671	0.684	0.713	0.803
ABCNN [27]	0.692	0.711	0.777	0.836
PWIM [5]	0.709	0.723	0.738	0.827
LDC [24]	0.706	0.723	0.771	0.845
IARNN [20]	0.734	0.742	-	-
BiMPM [23]	0.718	0.731	0.802	0.875
IWAN [14]	0.733	0.750	0.822	0.889
MCAN-SM [18]	-	-	0.827	0.880
MAN [19]	0.722	0.738	0.813	0.893
EDCMN (proposed)	**0.740**	**0.752**	**0.811**	**0.896**

4.4 Ablation Analysis

This section shows the impacts and contribution of the different components of our EDCMN model. Table 4 presents the results on the WikiQA test set. 'w/o' stands for without; 'fea' stands for feature; 'dyna' stands for Dynamic.

In the Encoder layer, we take four steps. First, we take away Embed-Cross feature and just let the Bi-LSTM's output as the input of Conv1D. The influence is small, causing a MAP to drop by 1% and MRR to drop by 0.9%, but it indicates that the effectiveness of the features reuse. Second, we remove Word-level feature, which is also called LSTM cross feature, The MAP drops by 1.5% and MRR drop by 1.9%. It shows that Word-level information is a useful supplement to the model. Third, we get rid of the Conv1D segment and find that the influence is huge, the MAP and MRR drop by 3.1% and 4.1% respectively which confirms the effectiveness of Phrase-level feature. At last, we abandon Sentence-level feature which is generated by self-attention, the influence of Sentence-level feature is close to Word-level feature, the MAP and MRR only drop by 1.6% and 1.7% respectively.

In the Decoder layer, we also take four ablation experiments. if we abandon the Match component and compress Encoder layer's output into the final vector directly for prediction, we find that our model has the worst performance on the dataset. Both MAP and MRR drop nearly by 8%. We can observe that Mat feature reduction had caused the most enormous influence (6%) in these match methods, indicating matching feature vectors with dynamic-size successfully acquires many rich characteristics of sentences.

Ablation analysis shows that different-scale information and dynamic-size feature vectors are important for matching, which confirms the effectiveness of our model.

Table 3. Performance for Textual Entailment on SCITAIL train and test set.

Models	Dev	Test
Majority class	63.3	60.3
DecompAtt	75.4	72.3
ESIM [3]	70.5	70.6
Ngram	65.0	70.6
DGEM [7]	79.6	77.3
EDCMN	**77.9**	**78.2**

Table 4. Ablation analysis for Answer Selection On WikiQA test set.

Model structure	MAP	MRR
Full Model	0.740	0.752
(1) w/o Embed-cross fea	0.731	0.739
(2) w/o Word-level fea	0.725	0.733
(3) w/o Phrase-level fea	0.709	0.711
(4) w/o Sentence-level fea	0.724	0.735
(5) w/o Sub fea (fixed-size)	0.720	0.733
(6) w/o Mul fea (fixed-size)	0.722	0.732
(7) w/o Mat-fea (dyna-size)	0.681	0.695
(8) w/o Match	0.672	0.675

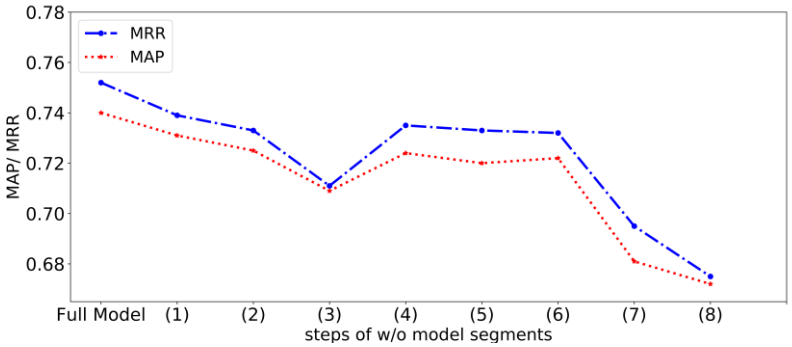

Fig. 2. Ablation Analysis about different components of model on WikiQA test set

5 Conclusion

In this paper, we propose an Encoder-Decoder Network with Cross-Match Mechanism, where the encoder layer is based on the "Siamese" network and decoder layer is based on the "matching-aggregation" network. The Cross-Match Mechanism which is the core innovation captures information of sentences at different scales including Sentence-level, Phrase-level, and Word-level. In addition, it explores sentence matching both on vectors with dynamic-size and fixed-size and it is more suitable for identifying complex relations between questions and the answers. In the experiments, we show that proposed model achieves state-of-the-art performance on the WikiQA dataset. In future work, we will incorporate external knowledge bases into our model to improve its performance. Furthermore, unlabeled data is much easier to obtain than labeled data. We will explore unsupervised methods for answer selection.

Acknowledgements

This research was partially supported by the Sichuan Science and Technology Program under Grant Nos. 2018GZ0182, 2018GZ0093 and 2018GZDZX0039.

References

1. Bachrach, Y. et al.: An attention mechanism for neural answer selection using a combined global and local view. Proc. - Int. Conf. Tools with Artif. Intell. ICTAI. 2017-Novem, 425–432 (2018). https://doi.org/10.1109/ICTAI.2017.00072.
2. Cascade-correlation, R., Chunking, N.S.: 1997Hochreiter_LSTM. 9, 8, 1–32 (1997).
3. Chen, Q. et al.: Enhanced LSTM for Natural Language Inference. 2008, 1657–1668 (2016).
4. Feng, M. et al.: Applying deep learning to answer selection: A study and an open task. 2015 IEEE Work. Autom. Speech Recognit. Understanding, ASRU 2015 - Proc. 813–820 (2016). https://doi.org/10.1109/ASRU.2015.7404872.
5. He, H., Lin, J.: Pairwise Word Interaction Modeling with Deep Neural Networks for Semantic Similarity Measurement. 937–948 (2016). https://doi.org/10.18653/v1/n16-1108.
6. Heilman, M., Smith, N.: Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. … Technol. 2010 Annu. Conf. …. June, 1011–1019 (2010).
7. Khot, T. et al.: SCITAIL: A Textual Entailment Dataset from Science Question Answering. AAAI Conf. Artif. Intell. (2018).
8. Lin, W.S. et al.: Utilizing different word representation methods for twitter data in adverse drug reactions extraction. TAAI 2015 - 2015 Conf. Technol. Appl. Artif. Intell. 260–265 (2016). https://doi.org/10.1109/TAAI.2015.7407070.
9. Lin, Z. et al.: A Structured Self-attentive Sentence Embedding. 1–15 (2017).
10. Meek, W.Y.C.: WIKI QA : A Challenge Dataset for Open-Domain Question Answering. September 2015, 2013–2018 (2018).
11. Rücklé, A., Gurevych, I.: Representation Learning for Answer Selection with LSTM-Based Importance Weighting. Proc. 12th Int. Conf. Comput. Semant. (IWCS 2017). (to appear) (2017).
12. Santos, C. dos et al.: Attentive Pooling Networks. Cv, (2016).
13. Severyn, A.: Rank with CNN. (2014). https://doi.org/10.1145/2766462.2767738.
14. Shen, G. et al.: Inter-Weighted Alignment Network for Sentence Pair Modeling. 1179–1189 (2018). https://doi.org/10.18653/v1/d17-1122.
15. Shen, T. et al.: DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. (2017).
16. Shen, Y. et al.: A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manag. - CIKM '14. 101–110 (2014). https://doi.org/10.1145/2661829.2661935.
17. Tan, M. et al.: LSTM-based Deep Learning Models for Non-factoid Answer Selection. 1, 1–11 (2015).
18. Tay, Y. et al.: Multi-Cast Attention Networks for Retrieval-based Question Answering and Response Prediction. (2018).

19. Tran, N.K., Niedereée, C.: Multihop Attention Networks for Question Answer Matching. 325–334 (2018). https://doi.org/10.1145/3209978.3210009.
20. Wang, B. et al.: Inner Attention based Recurrent Neural Networks for Answer Selection. 1288–1297 (2016). https://doi.org/10.18653/v1/p16-1122.
21. Wang, M. et al.: What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA. Proc. 2007 Jt. Conf. Empir. Methods Nat. Lang. Process. Comput. Nat. Lang. Learn. June, 22–32 (2007).
22. Wang, S., Jiang, J.: A Compare-Aggregate Model for Matching Text Sequences. 2016, 1–11 (2016).
23. Wang, Z. et al.: Bilateral multi-perspective matching for natural language sentences. IJCAI Int. Jt. Conf. Artif. Intell. 4144–4150 (2017).
24. Wang, Z. et al.: Sentence Similarity Learning by Lexical Decomposition and Composition. challenge 2, (2016).
25. Wang, Z., Ittycheriah, A.: FAQ-based Question Answering via Word Alignment. 1, (2015).
26. Yang, L. et al.: aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model. (2018).
27. Yin, W. et al.: ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. (2015).
28. He, K. et al.: Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-December, 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90.