

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/343183610>

Interpretable Duplicate Question Detection Models Based on Attention Mechanism

Article in Information Sciences · July 2020

DOI: 10.1016/j.ins.2020.07.048

CITATIONS

24

READS

282

3 authors, including:



[Qing Wang](#)

IBM Research

23 PUBLICATIONS 508 CITATIONS

[SEE PROFILE](#)

Interpretable Duplicate Question Detection Models Based on Attention Mechanism

Qifeng Zhou^{a,*}, Xiang Liu^a and Qing Wang^b

^aDepartment of Automation, Xiamen University, Xiamen, China

^bDepartment of Cognitive Service Foundations, IBM T.J. Watson Research Center, Yorktown Heights, NY, USA

ARTICLE INFO

Keywords:

Sentence pair matching

Interpretability

Attention Mechanism

ABSTRACT

Recently, there exist growing concerns about the interpretability of deep learning models. While few of these models have been applied to a duplicate question detection task, which aims at finding semantically equivalent question pairs of question answering forum. In this paper, based on an attention mechanism, we propose two modularized interpretable deep neural network models for such tasks. During the word precessing procedure, a filter operation is employed to enhance the relevant information contained in the pre-trained word embeddings. Regarding the word matching and sentence representation process, vanilla attention and structured attention mechanisms are utilized, respectively. Benefiting from the interpretability of attention techniques, our models can illustrate how the words match between sentence pairs and what aspects of the sentences are extracted to have an effect on the final decision. The attention visualization furnishes us with detailed representation at word and sentence level. And experimental results show that our models are comparable with other reported models.


1. Introduction

With the development of question answering community like Quora¹, Reddit² and Stack Overflow³, more and more individuals tend to post and answer questions in such forums. However, more and more newly posted questions cannot be answered quickly for the reason that fews have a proper recommended answer due to the lack of popularity. In practice, many newly posted questions have the similar or even the same answers as the stored ones³. Specifically, many questions, posted in different formulations, are of semantic equivalence, thus similar answers must be well matched to the corresponding ones. Therefore, to answer a newly posted question as quickly as possible, we can directly take advantage of the answers to the duplicate or nearly duplicate questions stored in the forums, which can bring a great convenience to question answering forum.

According to [3], two questions are regarded as duplicate question pair if they share common answers. This task is similar to the text similarity one. While the text similarity task has graded outputs indicating how similar they are, the output of duplicate question detection is yes/no. Thus this task can be viewed as a subtask of a text similarity identification that has a binary output and only contains the interrogative text.

For the text similarity task, many recently proposed deep learning models have achieved the state-of-the-art results [7, 23, 21, 8]. However, the existing methods have such limitations: (1) In many models, the final matching score is calculated through two individual sentence vectors. Considering each sentence pair as a single input, we argue that putting two sentences together to capture the integral information must be a more natural method. (2) For most of the sentence pair modeling models, the generated sentence vectors only contain single-aspect information. Nevertheless, multi-aspect information can be extracted from one sentence focusing on different part of it, which could definitely enrich the sentence representation. (3) Recently, the popularity of model interpretability is increasing [14, 4], which is very helpful for improving the quality of the online question answering forums. But many existing models proposed in [23, 3, 1] have no ability to tell us which words or phrases of two sentences contribute more to the final decision.

To address the aforementioned limitations, in this paper, two interpretable matching models based on attention mechanism are proposed to implement sentence pairs. Our models first match words of one sentence with words of

 zhouqf@xmu.edu.cn (Q. Zhou)

ORCID(s):

¹<https://www.quora.com>

²<https://www.reddit.com>

³<https://stackoverflow.com>

the other sentence, then perform sentence level interaction to generate similarity vectors (see Figure 1). The final decision will be made based on the vectors. Main contributions of this work are summarized as follows: (1) A filter operation is proposed to eliminate the redundant information of pre-trained word embeddings. (2) Based on structured attention [13], two models are proposed to enhance the sentence level information. (3) The proposed models are interpretable deep models, which will provide the semantic matching process illustrated.

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 illustrates and details the proposed model architecture. Section 4 and Section 5 show the datasets and the experimental results. Final conclusion and future work are described in Section 6.

2. Related Work

In natural language processing and information retrieval, there exist a large number of models for sentence pair modeling task, which includes the duplicate question detection task.

For the sentence pair modeling problem, many models apply the recurrent neural network (RNN) architecture. Due to the effectiveness of attention-based models in many NLP tasks [2, 19], some recent models try to utilize the attention mechanism to capture word interaction information to improve model performance.

Rocktäschel et al. [18] first apply a word-to-word attention mechanism to conduct word level interactions for a natural language inference task. Their method tries to attend over one sentence against the other sentence. Parikh et al. [16] propose a more effective attention-based model for that task. Through the attention mechanism, the sentence pair modeling problem is decomposed into many subproblems by their model. Based on the previous models and the attention mechanism, Chen et al. [5] tries enrich word level manipulations with the attention and simultaneously employ the bidirectional LSTM (BiLSTM) and the tree-LSTM [24] to capture more sentence information. Besides, Ghaeini et al. [7] present a strategy called *dependent reading*, which first employ a LSTM to encode one sentence, and utilize the last hidden state as the initial hidden state of the other LSTM encoding the other sentence.

Wang et al. [23] propose a bilateral multi-perspective matching (BiMPM) method to enrich the words interaction procedure. They conduct word level and granular level matchings in both directions. This matching operation is essentially very similar to the attention-based strategy.

Besides the RNN-based models, some convolutional neural network (CNN) models have recently been explored about their ability to model sentences or documents. And some work tries to encode sentences with RNN and then applies CNN pooling methods to get the final representation vectors.

Severyn and Moschitti [20] introduce a CNN architecture for reranking short text pairs. Without any manual feature engineering or external resources, their model shows the start-of-the-art performance on answer selection task. He and Lin [9] first calculate pairwise word interactions over the outputs of a BiLSTM layer and then apply a deep CNN to extract deep features for final classification. Conneau et al. [6] use the BiLSTM with max-pooling to encode the words and generate the sentence representation. Nie and Bansal [15] utilize a stacked BiLSTM with shortcut connections to encode the sentences. Their model concatenates all the hidden states of previous BiLSTM layers as the input of next BiLSTM layer. The final sentence vector is the max pooling over the hidden states of the last BiLSTM layer.

Bogdanova et al. [3] also design a CNN-based strategy to detect duplicate question pairs. Experimented on the datasets extracted from different online forums, their model outperforms the baseline ones by a noticeable margin. Antonio Rodrigues et al. [1] provide a deep discussion about different kinds of models. And their proposed novel conventional neural network model achieves the best performance on the duplicate question detection task. However, their proposed models, as well as the former CNN-based models, take no interpretability into consideration and thus show no words interaction.

3. The Interpretable Models

Let $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_m\}$ be the sentence pair, where $p_i, q_j \in R^d$ are pre-trained word embeddings, and n and m are the length of P and Q , respectively. Their relationship is represented by the label y . The overall architecture of the proposed model is depicted in Figure 1.

For the sake of simplicity, the dashed lines is utilized to illustrate the BiLSTM layer, the sequential processing units which will be formulated in detail. We propose different attention-based matching methods at different layers to identify the relationship between two sentences. As is shown in the figure, the proposed framework contains three major modules (separated by the BiLSTM layer). In the last module, two kinds of representation methods are proposed, and we will detail them in the corresponding layers.

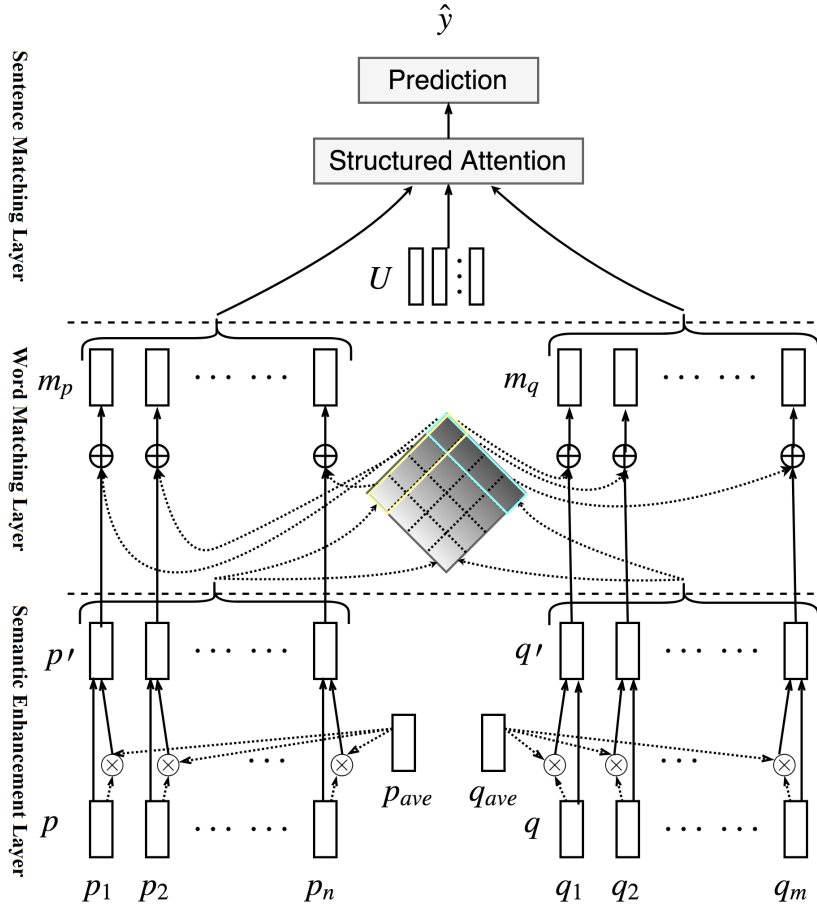


Figure 1: The proposed model architecture. \otimes indicates filter process in the semantic enhancement layer, and \oplus is the information aggregation operation of word matching layer.

3.1. Semantic Enhancement Layer

In this layer, we use the pre-trained word embedding as inputs. However, the pre-trained word embeddings are generally trained under some specific circumstances. We argue that using task-related representation may contribute more to our classification task. Hence we apply a *filter* operation to enhance relevant information.

Take one sentence as an example, we first calculate the non-sequential sentence vectors by merely averaging the pre-trained word embeddings:

$$p_{ave} = \frac{1}{n} \sum_{i=1}^n \tanh(W_f p_i), \quad (1)$$

where W_f is the filter weight. Then we compute the filter factor for each word as follows (denoted as \otimes in Figure 1):

$$f_i^p = \text{sigmoid}(p_i^T p_{ave}). \quad (2)$$

The larger filter factor denotes the word embedding contains more relevant information. The ultimate word representation is:

$$p'_i = [p_i; f_i^p p_i]. \quad (3)$$

The final word vector contains two parts: the pre-trained embedding and the task-related embedding. Additionally, such enhanced representation can be treated as the combination of the fixed word embedding and trainable word embedding, which, therefore, avoids discussing whether to update the word embeddings when training the models.

The enhanced word embeddings will be further inputted into a BiLSTM layer (depicted as the dashed line in Figure 1), which is a common method to generate contextual vectors.

$$\begin{aligned}\bar{p}_i &= \text{BiLSTM}(p'_i), i = 1, \dots, n, \\ \bar{q}_j &= \text{BiLSTM}(q'_j), j = 1, \dots, m,\end{aligned}\quad (4)$$

where $\bar{p}_i, \bar{q}_i \in R^h$ are concatenation of forward and backward outputs of BiLSTM.

3.2. Word Matching Layer

In this layer, for each word of one sentence, relevant information in the other sentence will be assembled through the attention approach. These collected information will be applied to enrich the word representation of the sentence.

Attentive Matching: Relevant information for one word is discovered via word-to-word attention weights. The attention weights (the grey matrix grid shown in Figure 1) between two words are computed as follows:

$$e_{ij} = \bar{p}_i^T \bar{q}_j, \quad (5)$$

where \bar{p}_i and \bar{q}_j are computed by the previous layers. Now for each word in one sentence, the relevant information is extracted and composed through normalized attention weights.

$$\hat{p}_i = \sum_{j=1}^m \frac{\exp(e_{ij})}{\sum_{k=1}^m \exp(e_{ik})} \bar{q}_j, \quad i = 1, \dots, n \quad (6)$$

$$\hat{q}_j = \sum_{i=1}^n \frac{\exp(e_{ij})}{\sum_{k=1}^n \exp(e_{kj})} \bar{p}_i, \quad j = 1, \dots, m \quad (7)$$

The normalized attention weights can be viewed as normalized matching score distribution of one word in one sentence against all words of the other sentence. Thus the information relevant to the word is collected by the weighted summation of the word vectors in the other sentences.

Information Aggregation: To enrich the word vectors, more relevant information needs to be collected. One typical aggregation approach is to concatenate the summation and the original representation. While such concatenation fails to extract the interactive information, thus we further employ the difference and element-wise product between the summation and the original vectors [5, 7]. Besides, we apply max pooling and average pooling to enhance the interactive information and obtain a more sophisticated information. The following equations are denoted as \oplus in Figure 1.

$$\begin{aligned}p_d &= \hat{p} - \bar{p}, \\ p_e &= \hat{p} \odot \bar{p}, \\ p_{max} &= \text{MaxPooling}(\hat{p}, \bar{p}), \\ p_{mean} &= \text{AvePooling}(\hat{p}, \bar{p}), \\ m_p &= [\bar{p}; \hat{p}; p_d; p_e; p_{max}; p_{mean}], \\ q_d &= \hat{q} - \bar{q}, \\ q_e &= \hat{q} \odot \bar{q}, \\ q_{max} &= \text{MaxPooling}(\hat{q}, \bar{q}), \\ q_{mean} &= \text{AvePooling}(\hat{q}, \bar{q}), \\ m_q &= [\bar{q}; \hat{q}; q_d; q_e; q_{max}; q_{mean}],\end{aligned}\quad (8)$$

where \odot means element-wise product, and m_p and m_q is the aggregated representation for \bar{p} and \bar{q} , respectively, both of which contain aggregated information.

In Equation 8 and 9, the aggregated representations m_p and m_q are very high-dimensional vectors. A feedforward layer with ReLU activation function is utilized to lower the dimension of the vectors and simultaneously capture deeper dependencies.

$$\begin{aligned} a &= \text{ReLU}(W_p m_p + b_p), \\ b &= \text{ReLU}(W_p m_q + b_p), \end{aligned} \quad (10)$$

where W_p and b_p are the trainable parameters of the feedforward layer.

Since there still exist sequential dependencies among the aggregated word representations of the sentence, we thus feed these representations into another BiLSTM to capture aggregated sequential information and generate deeper representations.

$$\begin{aligned} \bar{a}_i &= \text{BiLSTM}(a_i), \quad i = 1, \dots, n, \\ \bar{b}_j &= \text{BiLSTM}(b_j), \quad j = 1, \dots, m, \end{aligned} \quad (11)$$

where $\bar{a}_i, \bar{b}_j \in R^h$ are the outputs of BiLSTM. The parameters of BiLSTM here are different from that in Equation 4.

3.3. Sentence Matching Layer

In this layer, some models either employ max and average pooling over the outputs of BiLSTM or concatenates the last hidden states of BiLSTM to describe the overall sentence representation [5, 23]. And some use the dot product or cosine similarity of two sentence vectors as the final matching score [3, 1]. We argue that such an operation based on multi-aspect sentence vectors can capture as much semantics of the sentences as possible. Thus we apply the structured attention [13] to attempt to extract different aspects of semantics of the sentences. Based on the structured attention mechanism, multiple sentence vectors are extracted, resulting in multiple matching scores. Besides, a novel method of modeling two sentences is proposed, which could extract joint representation vectors as the final representation.

Structured Attention: For a specific sentence $\{\bar{a}_i | i = 1, \dots, n\}$, a parameters vector u (called a context vector), which has the same dimension as \bar{a}_i , is predefined. For one word of the sentence, the unnormalized attention weight will be:

$$\alpha_i = u^T \tanh(W_s \bar{a}_i + b_s), \quad (12)$$

where $W_s \in R^{h \times h}$ and $b_s \in R^h$ are the trainable weights and bias. After obtaining the attention weights for all word of the sentence, we use softmax function to normalize them to get a probability distribution α over all words. The vector (probability distribution) α shows which part of the sentence should be focused on given the context vector u . Under this setting, multiple context vectors will, according to Equation 12, generate multiple distributions focusing on multiple aspects of the sentence.

$$H = [\bar{a}_1; \bar{a}_2; \dots; \bar{a}_n], \quad (13)$$

$$A = \text{Softmax}(U^T \tanh(W_s H + b_s)), \quad (14)$$

$$Y_a = AH^T, \quad (15)$$

where $U \in R^{h \times r}$ is the parameter matrix, $A \in R^{r \times n}$ is the normalized attention weights, and r is the number of the context vectors. The final sentence representation is $Y_a \in R^{r \times h}$, of which each row represents the sentence of different aspects, or more specifically, each vector can represent the sentence, but different vector pays attention to different parts of the sentence [13].

Integration Representation: In the Siamese framework, two sentences are individually modeled, and the final decision is made by two sentence vectors. However, because two sentences are given as a pair, the whole sentence pair is observed simultaneously. It is natural to make a final judgment based on all observed information. Thus our intuitive idea is to integrate two sentences and then generate the integrated sentence pair representation. Figure 2 shows the integration process and Equation 16 to 19 detail the computation.

$$H_a = [\bar{a}_1; \bar{a}_2; \dots; \bar{a}_n], \quad (16)$$

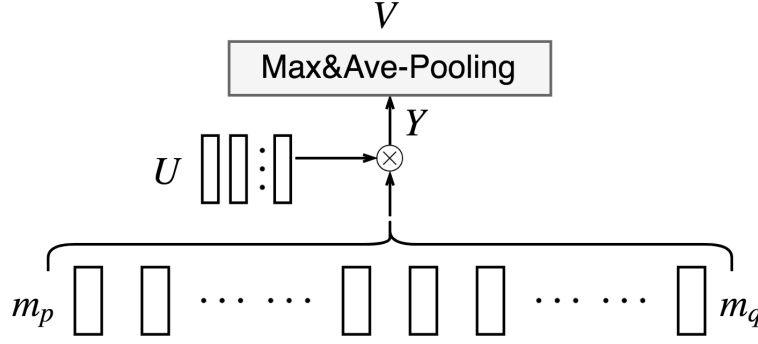


Figure 2: Integration Representation. Two sentences are integrately modeled by context vectors to generate multiple sentence vectors Y .

$$H_b = [\bar{b}_1; \bar{b}_2; \dots; \bar{b}_m], \quad (17)$$

$$A = \text{Softmax}(U^T \tanh(W_s[H_a; H_b] + b_s)), \quad (18)$$

$$Y = A[H_a; H_b]^T, \quad (19)$$

where the parameter matrix is $U \in R^{h \times r}$, $W_s \in R^{h \times h}$ is the weight matrix, and $b_s \in R^h$ is the bias. Different from Equation 14, the normalized attention weight here is $A \in R^{r \times (n+m)}$, each row of which is the integrated probability distribution over the joint two sentences. Each row tells us what parts of the two sentences to be focused on, and corresponding sentence vector will mostly determined by the focused parts of two sentences. Consequently, the structured attention based representation $Y \in R^{r \times h}$ contains r aspects of the integrated sentence pair.

The final integration representation is:

$$V = [\text{MaxPooling}(Y); \text{AvgPooling}(Y)] \quad (20)$$

We apply MaxPooling and AvgPooling to extract the most relevant information, meanwhile considering to reduce the computing time and memory usage.

Sentence Matching Representation: We also apply the most common approach to generate the final decision. However, different from the aforementioned methods, our multi-aspect sentence vectors produce multiple matching scores. The whole process is depicted in Figure 3 and Equation 21 to 24.

$$A_a = \text{Softmax}(U^T \tanh(W_s H_a + b_s)), \quad (21)$$

$$A_b = \text{Softmax}(U^T \tanh(W_s H_b + b_s)), \quad (22)$$

$$Y_a = A_a H_a^T, \quad (23)$$

$$Y_b = A_b H_b^T, \quad (24)$$

where the trainable parameters U , W_s , and b_s are shared by two sentences. $Y_a = [y_{a,1}, \dots, y_{a,r}]^T$ and $Y_b = [y_{b,1}, \dots, y_{b,r}]^T$ are multiple structured attention representations of two sentences. Each row of them can represent one sentence, respectively. Therefore, any row can be used to match with any the other rows.

Thus we obtain multiple matching scores. Here we apply the dot product as a matching approach, which is the most common method to compute the similarity and easy to be implemented. Take two sentences $Y_{a,i}$ and $Y_{b,i}$ as an example, the score is computed as follows:

$$s(Y_{a,i}, Y_{b,i}) = Y_{a,i}^T Y_{b,i}. \quad (25)$$

We also test the *bilinear* method to compute matching scores. Bilinear is designed to capture deeper interactions between two vectors [22]¹. But in our experiments, the performance of bilinear is inferior to dot product.

¹Bilinear is computed as follows:

$$s(Y_{a,i}, Y_{b,i}) = Y_{a,i}^T M Y_{b,i} + b,$$

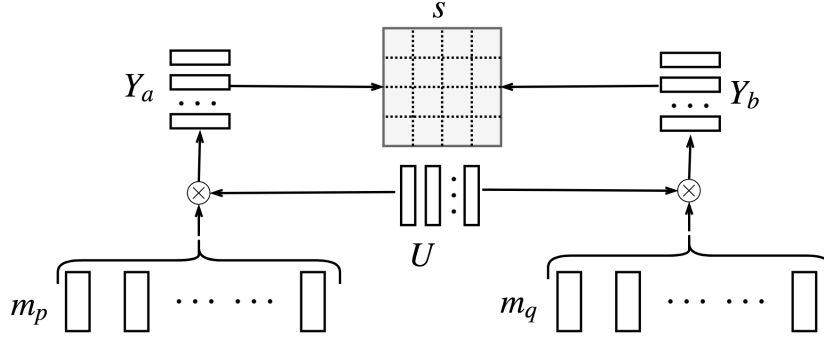


Figure 3: Sentence Matching Representation. Two sentences are individually modelled by context vectors, generating multiple vectors for two sentences.

The matching scores between any two representations of two sentences compose the similarity matrix, of which each row is computed by one aspect of one sentence against all aspects of the other sentence, and each column is just the opposite.

In the Siamese network, the notion of similarity or dissimilarity requires context, which means the network is sensitive to calibration [10]. Therefore, the predefined parameter matrix U can be deemed to be the calibration or context of the Siamese matching network. The generated representation Y_a and Y_b are calibrated or context based sentence representations.

3.4. Classification

Now we have two kinds of matching representations, the integration representation V and the sentence matching representation s (named **InteMatch** and **SenMatch** respectively). The representation will be fed into a two-layer feed-forward neural network with tanh activation function and a softmax output layer.

$$\text{InteMatch} : \hat{y} = \text{Softmax}(\tanh(WV + b)),$$

$$\text{SenMatch} : \hat{y} = \text{Softmax}(\tanh(Ws + b)),$$

where W and b are trainable parameters.

Then we adopt the cross-entropy loss to train the models:

$$\text{Loss} = \text{CE}(\hat{y}, y), \quad (26)$$

where CE computes the cross-entropy between the prediction vectors \hat{y} and the ground truth labels y .

The overall prediction and training procedure is described as follows:

- Step 1.** Enhance the word representations by the filter layer, and input them into a BiLSTM layer to generate the contextual vectors;
- Step 2.** Aggregate the interactive information by the word-to-word attention weights;
- Step 3.** InteMatch: generate the multiple vectors for the sentence pair and extract the integral representation;
SenMatch: generate the multiple vectors for each sentence and compute the match scores.
- Step 4.** Compute the loss between the prediction vectors and the truth label, and train the model with given back-propagation algorithm.

Note that the final loss function contains all the trainable parameters (the filter weights, the parameters of ReLU layer, the structured attention weights, and the parameters of all the BiLSTM layers). And the optimization algorithm we utilize in our models will update the parameters given the loss function.

where M and b is the trainable parameter matrix and bias.

Table 1
Quora Question Pair

Pair ID	Question Pair	Is Duplicate
447	What are natural numbers?	0
	What is least natural number?	
3272	How do you start a bakery?	1
	How can one start a bakery business?	

4. Datasets

The proposed models (**InteMatch** and **SenMatch**) are tested on the different datasets which are collected from different question answering forums.

Quora A brand new dataset released by Quora Question Answering Website². The dataset [23] contains more than 400K question pairs, of which over 380K are sampled as the training set, 10K as the validation set, and 10K as the test set. Table 1 shows some examples of different relationships from Quora dataset.

AskUbuntu The dataset [1] is extracted from Ask Ubuntu Community Questions and Answers site³. The dataset contains 24K training pairs, 1K validation pairs, and 6K test pairs, which is smaller than the Quora dataset. We used two datasets released by them: one contains both title and body (referred as AskUbuntuTB in [1]) and the other only contains the title (referred as AskUbuntuTO, no validation set).

Quora_few This dataset is a portion with 30K question pairs randomly extracted from the Quora dataset by [1]. It is released in the same manner as the AskUbuntuTO dataset.

Meta This dataset [1] is randomly extracted from Meta Stack Exchange⁴ data dump. It has disjoint 20K, 1K and 4K question pairs for training, validation, and testing. And it is prepared in the same manner as the AskUbuntu dataset.

5. Experiments

5.1. Parameter Settings

The word embeddings are initialized with the pre-trained 300-dimensional GloVe word vectors [17]. And the out-of-vocabulary (OOV) words are randomly initialized. During the training process, the word embeddings are set fixed. We employ the Adam optimization method [11] to minimize the cross-entropy loss and update trainable parameters. The hidden size is set as 300 for BiLSTM layers. We apply dropout for all feedforward layers, and the dropout ratio is set as 0.1. The batch size for Quora dataset is 256, and for other datasets is 64. The initial learning rate for Quora is set as 0.00009 and for other datasets is 0.0001. And the learning rate decreases $0.5 * 10^{-5}$ every ten epochs. For the structured attention representation layer, we set the number of context vectors as 20.

5.2. Experimental Results

Quora Table 2 shows the accuracy of models on test sets of Quora dataset. The baseline models, based on CNN or RNN, are designed by Wang et al. [23]. We list their reported results in our table. The baseline models are all implemented based on their architectures. The dataset is also tested by [1], but their data is only a small part of the whole Quora dataset. Thus we don't list their results here. Besides, Lan and Xu [12] replicate some attention based deep learning models, and test them on the Quora dataset. In order to compare our models with other attention based models, we list some replicated results in [12].

As Table 2 shows, the BiMPM has the best performance, and InferSent and SSE also have relatively high accuracy. Except that the tree-LSTM based ESIM performs worst, other attention based models (DecAtt, $ESIM_{seq+tree}$, $ESIM_{seq}$) have comparable performances. Our proposed SenMatch model achieves a relatively high accuracy. And the InteMatch model attains the best performance among all the attention based models, ranking the third in all the models, which verifies the efficiency of the proposed idea of modeling the integrated sentence pairs.

Compared with the reported accuracy of the BiMPM and SSE, the accuracy of our models drops. We believe that the main reason for this is that more complicated are carried out in BiMPM and SSE. Compared with our attention

²<https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs>

³<http://askubuntu.com/>

⁴<http://meta.stackexchange.com/>

Table 2
Performance on Quora dataset.

Models	Accuracy(%)
Siamese-CNN[23]	79.60
Multi-Perspective-CNN[23]	81.38
Siamese-LSTM[23]	82.58
Multi-Perspective-LSTM[23]	83.21
L.D.C.[23]	85.55
BiMPM [23]	88.17
ESIM _{tree} [5]	75.50
PWIM [9]	83.40
ESIM _{seq} [5]	85.00
ESIM _{seq+tree} [5]	85.40
DecAtt[16]	86.50
InferSent [6]	86.60
SSE [15]	<u>87.80</u>
InteMatch	<u>86.81</u>
SenMatch	86.45

based matching, more complicated operations are carried out in the word matching layer of the BiMPM model, which is called multi-perspective matching and extracts more information between two sentences. Besides, the BiMPM applies the highway network at each layer, which can further capture the deeper information from two sentences. The SSE (shortcut-Stacked Sentence Encoder [15]) employs the stacked BiLSTM. Their BiLSTM of one layer takes the outputs of all the previous layers as inputs (which is called shortcut connections). And the final sentence representation is obtained by max-pooling over the outputs of the last BiLSTM layer. Although our models attain relatively lower accuracy than the BiMPM and SSE, yet we utilize more simple and more intuitive operations to conduct word matching and model sequences. The final visualization of the attention weights of word matching and sentence representation gives us a better understanding of our models.

Quora_few Table 3 lists the experimental results of models from [1] and our models. For the Quora dataset, we set the batch size 256. However, The Quora_few dataset has smaller number of question pairs, thus we set its batch size 64. And in the rest of the datasets, we use almost the same parameter setting (including the learning rate and the batch size), and we don't run many epochs of the proposed models on the rest of the datasets.

Table 3 shows that our models perform much better than the listed baseline models on the Quora_few dataset, the reason for which is that the baselines are not well designed, like our models. However, compared with the accuracy of our models on the larger Quora dataset, the figure drops remarkably, which may result from the insufficient number of training data of the Quora_few dataset.

AskUbuntu Table 4 shows the experimental results on the AskUbuntu dataset. We use the same dataset as [1]. During the preprocessing of [1], they eliminate the tag "Possible duplicate: <title>" from the whole text, and this leads to an accuracy drop compared with experiments with the tag. To validate the effectiveness of the proposed models we also test the models on the dataset without the "Possible duplicate: <title>" tag (referred as InteMatch-w/o-dup and SenMatch-w/o-dup).

According to the experimental results, our proposed models outperform other models by a very large margin on both AskUbuntuTB (with and without "Possible duplicate: <title>" tag) and AskUbuntuTO datasets.

Under similar experimental settings, the reported accuracy in [3] is 92.9%. And the accuracy of 94.20% is obtained by [1] on the AskUbuntuTB dataset that keeps the "Possible duplicate <title>" phrase, which is still not comparable

Table 3
Performance on Quora_few dataset.

Models	Accuracy(%)
Jcrd	69.53
SVM-bas	64.93
SVM-adv	68.56
CNN	59.90
DNN	69.53
DCNN	71.48
InteMatch	73.03
SenMatch	75.82

Table 4
Performance on AskUbuntu dataset.

Models	Accuracy(%)	
	AskUbuntuTB	AskUbuntuTO
Jcrd	72.91	72.35
SVM-bas	70.25	68.88
SVM-adv	75.87	70.87
CNN	74.50	74.12
DNN	78.65	78.40
DCNN	79.00	79.67
InteMatch-w/o-dup	92.05	-
SenMatch-w/o-dup	92.00	-
InteMatch	95.55	83.80
SenMatch	96.08	83.53

with our models in the same setting.

Meta Table 5 shows us the experimental results on Meta dataset (w/o the “Possible duplicate <title>” tag).

From Table 5, we find that both models perform worse on the dataset with the “Possible duplicate <title>” tag. This mainly results from that many non-duplicate question pairs in the dataset with the “Possible duplicate <title>” tag contain the tag, which may affect the model performance.

Besides, compared with the reported testing accuracy of 92.68% in [3], the proposed models perform worse. The main reason is that we don’t employ the domain-specific word embeddings as [3] did and our models are not fine-tuned (former experimental results indicate that the proposed models perform well on small datasets, thus we use the same parameter setting for the small datasets).

From the testing accuracy of our models on different datasets, we can conclude that our models can get a better performance on domain-specific datasets. On the domain-specific datasets (Meta and AskUbuntu), our proposed models show the state-of-the-art performance, but our models only reach a relatively high accuracy on the Quora dataset. Besides, comparing the performance of our models on the Quora and Quora_few datasets, we can find that our deep models can be trained to achieve better performance with the dataset containing more data samples.

Table 5
Performance on Meta dataset.

Models	Accuracy(%)
InteMatch-w/o-dup	90.55
SenMatch-w/o-dup	90.05
InteMatch	88.83
SenMatch	87.80

5.3. Ablation Analysis

As is detailed in Section 3, our proposed models are highly modularized, and consist of several different semantic processing modules:

- *Filter Module* This module can be viewed as a processing step. Pre-trained word embeddings which maybe contain redundant information will be input into this layer. And after filtering redundant information out, the relevant information will be concatenated with the pre-trained word embeddings to enhanced word information.
- *Information Aggregation Module* Based on the attention mechanism, the relevant information in one sentence will be aggregated individually for each word of the other sentence. After some interactive operation, the final word representation will aggregate the origin information and other relevant interactive information.
- *Structured Attention Module* In this module, we try to extract multiple aspects of sentences by structured attention. The final matching scores are calculated by multi-aspect sentence vectors.

To evaluate the effectiveness of different modules, we construct three ablation experiments, each of which eliminates one module respectively:

- w/o Filter: eliminate filter layer of InteMatch and SenMatch models.
- w/o Aggregation: eliminate aggregation layer of InteMatch and SenMatch models.
- w/o Structured Attention: in InteMatch, we just eliminate Equation 18 and delete the attention weights in Equation 19; in SenMatch, after eliminating structured attention layer, we just conduct average pooling and max pooling on both sentences, and their concatenation will be input fed into the feed-forward neural network.

Table 6 shows the experimental accuracy on the test set. In the brackets are the dropped accuracy values.

Table 6
Ablation experiments on Quora dataset

Models	InteMatch(%)	SenMatch(%)
Full Model	86.81	86.45
w/o Filter	86.59(-0.22)	85.98(-0.47)
w/o Aggregation	82.87(-3.94)	83.44(-3.01)
w/o Structured Attention	84.16(-2.65)	83.15(-3.30)

We can clearly notice that the accuracy drops more or less without any of the proposed modules. Compared with the filter layer, the aggregation layer and structured attention layer play a much more important role in both models.

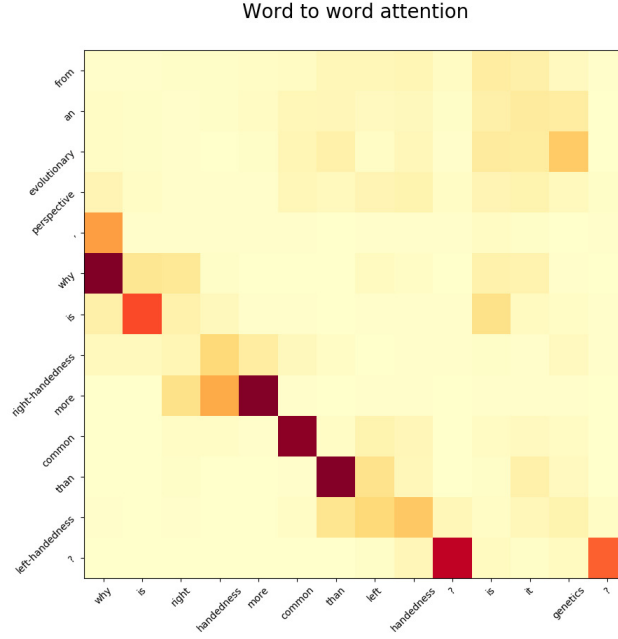


Figure 4: Word Level Attention

Besides, in different models, eliminating aggregation layer or structured attention layer has different influence on the final performance.

For InteMatch model, the aggregation layer plays a major role, without which the accuracy drops by 3.94%, larger than without structured attention. This fact seems to indicate that in the InteMatch the major sentence interaction is much more important. However, this may be caused by not adding penalization term [13], which, illustrated in the next subsection, could result in multiple similar integrated sentence representations. While in SenMatch model, both the aggregation module and structured attention module are of much importance, and the accuracy drop by a near value. This indicates that both modules are of great importance.

5.4. Interpretability Visualization

5.4.1. The Attention of InteMatch Model

Due to the interpretability of the attention mechanism, word-level attention and structured attention can interpret and visualize how the words in one sentence match with the words in the other sentence and which aspect of the sentence is identified.

We randomly select a sample from the test set, where one sentence is “*Why is right handedness more common than left handedness? Is it genetics?*”, and the other is “*From an evolutionary perspective, why is right-handedness more common than left-handedness?*”, and their label class is 1 (duplicate questions). Figure 4 shows the word attention weights (Equation 5) of the sentence pair, where the darker color denotes larger attention value.

From Figure 4, we find that the InteMatch model attends to some semantically related words or phrase pairs, such as {handedness, right-handedness} and {left handedness, left-handedness}. Besides, the proposed model detects the word pair {genetics, evolutionary}, which is much harder to be detected than the former ones. All the detected pairs are decisive parts to identify two sentences as a duplicate.

Figure 5 shows the integration attention weights calculated by Equation 18 at the sentence level. Each row of Figure 5 represents one attention distribution on two sentences, from which we can find that words mentioned before are attended again at the sentence level. However, most of attention distributions focus on the same words, which results from that we train the model without penalization term added on the integration attention weights [13].

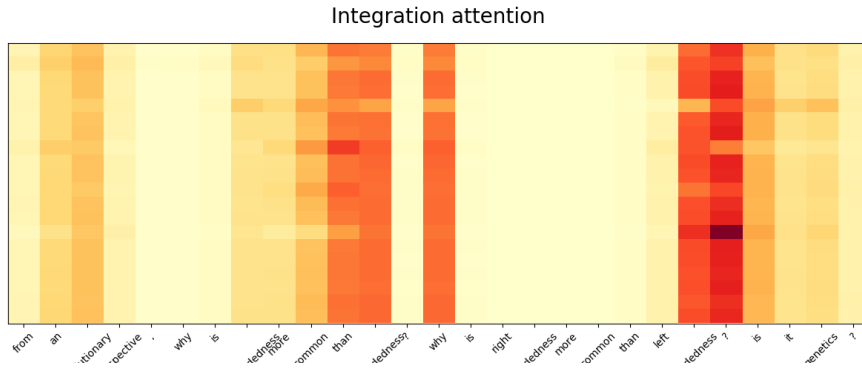


Figure 5: Integration Attention

5.4.2. Attention of SenMatch Model

Based on the same test sample, Figure 6 illustrates all kinds of attention weights (Equation.5, 21 and 22) and matching scores (Equation.25) of SenMatch model in four different parts.

The top right part shows the word level attention weights (Equation.5), which is very similar to InteMatch model in Figure 4. However, compared with InteMatch model, the word pair {genetics, evolutionary} seems not to be detected by SenMatch model.

The top left and bottom right parts show the structured attention weights of two sentences (Equation.21 and 22). In the top left part, each column is one aspect of attention distribution on a sentence, and each row represents one word (i.e., the top left part have the same vertical axis as the top right part). In the bottom right part, each row illustrates one attention distribution on the other sentence, and the columns represent different words (the horizontal axis of the top right part). The SenMatch model is also trained without a penalization term, but the structured attention distributions here attend on different words or phrases, which indeed extracts multiple aspects of two sentences.

Finally, the bottom left part illustrates the overall matching scores of different aspects of the two sentences. The darker color of this part means a higher value, which denotes that the corresponding attention distributions on two sentences attend to the semantically similar words or phrases. As is depicted in Figure 6, two sentences focus on the similar phrases {why is right-handedness} and {why is right handedness} (the ellipses), and therefore the corresponding color are darker.

6. Conclusion and Future Work

In this paper, we propose two interpretable sentence matching models based on attention mechanism. Compared with other methods (w/o the attention mechanism), the proposed models achieve comparable performance on the large-scale duplicate question detection dataset and best performance on almost the small-scale datasets.

In our models, we apply a filter operation to preprocess pre-trained word embeddings, which avoids discussing whether to update the word embeddings during training phrase. And our proposed integration sentence representation model achieves the best performance on AskUbuntuTO dataset, which verifies the efficiency of the integration representation. Besides, a traditional sentence matching representation is enriched by multiple aspects of sentence based on structured attention mechanism. Experimental results show that our models are easy to be trained on small-scale datasets.

Future work includes building a model without recurrent neural network units, to speed up our model training. Trying to build a theoretically interpretable sentence pair model is another interesting work. Besides, we can also find out how to apply external knowledge to avert misunderstanding typical words of the dataset. And most of our dataset sizes are not very large, how to extend our models to large size dataset is also worth studying.

Acknowledgments

This work is supported in part by the National Science Foundation of Fujian Province (China) under Grant No.2017J01118 and Shenzhen Science and Technology Planning Program under Grant No. JCYJ20170307141019252.

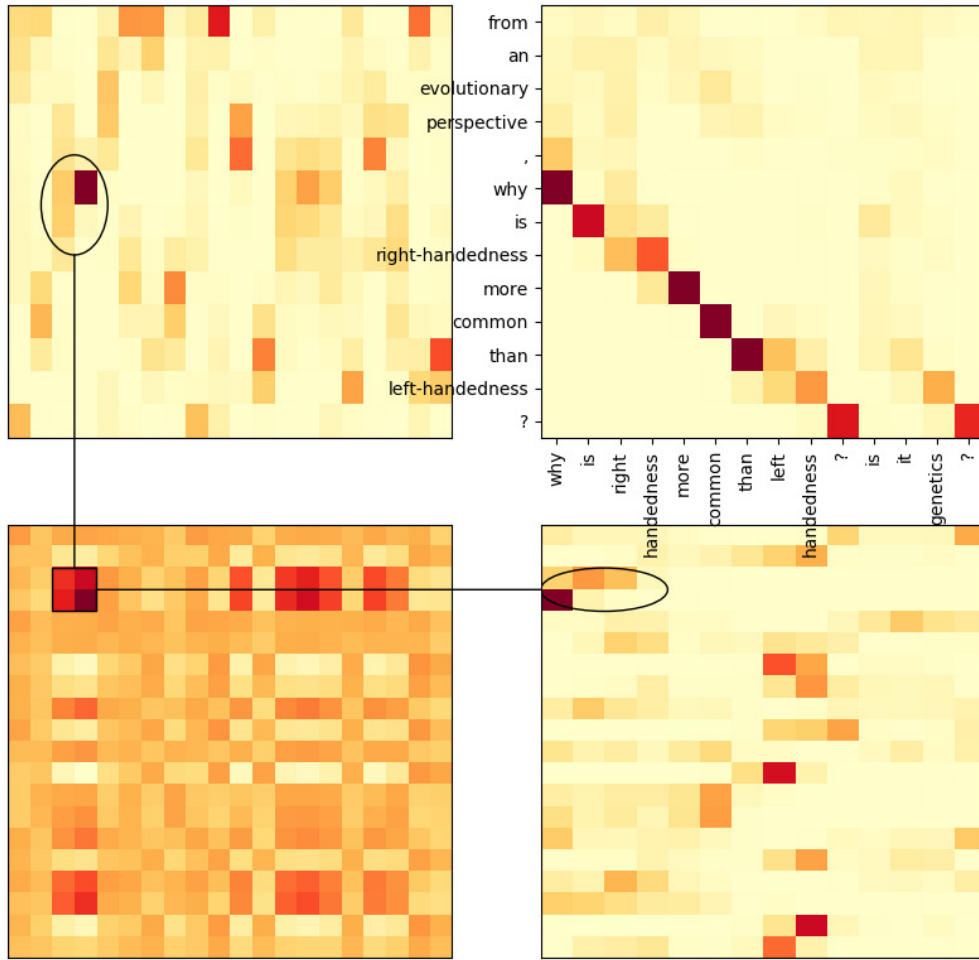


Figure 6: Attention of SenMatch Model

References

- [1] Antonio Rodrigues, J., Saedi, C., Maraev, V., Silva, J., Branco, A., 2017. Ways of asking and replying in duplicate question detection, in: Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017), Association for Computational Linguistics, Vancouver, Canada. pp. 262–270. URL: <http://www.aclweb.org/anthology/S17-1030>.
- [2] Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473. URL: <http://arxiv.org/abs/1409.0473>, arXiv:1409.0473.
- [3] Bogdanova, D., dos Santos, C.N., Barbosa, L., Zadrozny, B., 2015. Detecting semantically equivalent questions in online user forums, in: Alishahi, A., Moschitti, A. (Eds.), Proceedings of the 19th Conference on Computational Natural Language Learning, CoNLL 2015, Beijing, China, July 30-31, 2015, ACL. pp. 123–131. URL: <http://aclweb.org/anthology/K/K15/K15-1013.pdf>.
- [4] Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M.B., Preece, A.D., Julier, S., Rao, R.M., Kelley, T.D., Braines, D., Sensoy, M., Willis, C.J., Gurram, P., 2017. Interpretability of deep learning models: A survey of results, in: 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI 2017, San Francisco, CA, USA, August 4-8, 2017, IEEE. pp. 1–6. URL: <https://doi.org/10.1109/UIC-ATC.2017.8397411>, doi:10.1109/UIC-ATC.2017.8397411.
- [5] Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H., Inkpen, D., 2017. Enhanced LSTM for natural language inference, in: Barzilay, R., Kan, M. (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, Association for Computational Linguistics. pp. 1657–1668. URL: <https://doi.org/10.18653/v1/P17-1152>, doi:10.18653/v1/P17-1152.
- [6] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A., 2017. Supervised learning of universal sentence representations from natural language inference data, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark. pp. 670–680. URL: <https://www.aclweb.org/anthology/D17-1070>.

- [7] Ghaeini, R., Hasan, S.A., Datla, V., Liu, J., Lee, K., Qadir, A., Ling, Y., Prakash, A., Fern, X., Farri, O., 2018. Dr-bilstm: Dependent reading bidirectional lstm for natural language inference, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics. pp. 1460–1469. URL: <http://aclweb.org/anthology/N18-1132>, doi:10.18653/v1/N18-1132.
- [8] Gong, Y., Luo, H., Zhang, J., 2017. Natural language inference over interaction space. CoRR abs/1709.04348. URL: <http://arxiv.org/abs/1709.04348>, arXiv:1709.04348.
- [9] He, H., Lin, J., 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement, in: Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 937–948.
- [10] Hoffer, E., Ailon, N., 2015. Deep metric learning using triplet network, in: Feragen, A., Pelillo, M., Loog, M. (Eds.), Similarity-Based Pattern Recognition, Springer International Publishing, Cham. pp. 84–92.
- [11] Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980. URL: <http://arxiv.org/abs/1412.6980>, arXiv:1412.6980.
- [12] Lan, W., Xu, W., 2018. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering, in: Proceedings of the 27th International Conference on Computational Linguistics (COLING).
- [13] Lin, Z., Feng, M., dos Santos, C.N., Yu, M., Xiang, B., Zhou, B., Bengio, Y., 2017. A structured self-attentive sentence embedding. CoRR abs/1703.03130. URL: <http://arxiv.org/abs/1703.03130>, arXiv:1703.03130.
- [14] Lipton, Z.C., 2016. The mythos of model interpretability. CoRR abs/1606.03490. URL: <http://arxiv.org/abs/1606.03490>, arXiv:1606.03490.
- [15] Nie, Y., Bansal, M., 2017. Shortcut-stacked sentence encoders for multi-domain inference, in: Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP, Association for Computational Linguistics, Copenhagen, Denmark. pp. 41–45. URL: <http://www.aclweb.org/anthology/W17-5308>.
- [16] Parikh, A., Täckström, O., Das, D., Uszkoreit, J., 2016. A decomposable attention model for natural language inference, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas. pp. 2249–2255. URL: <https://aclweb.org/anthology/D16-1244>.
- [17] Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25–29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL. pp. 1532–1543. URL: <http://aclweb.org/anthology/D/D14/D14-1162.pdf>.
- [18] Rocktäschel, T., Grefenstette, E., Hermann, K.M., Kocisky, T., Blunsom, P., 2016. Reasoning about entailment with neural attention, in: International Conference on Learning Representations (ICLR).
- [19] Rush, A.M., Chopra, S., Weston, J., 2015. A neural attention model for abstractive sentence summarization, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal. pp. 379–389. URL: <http://aclweb.org/anthology/D15-1044>.
- [20] Severyn, A., Moschitti, A., 2015. Learning to rank short text pairs with convolutional deep neural networks, in: Baeza-Yates, R.A., Lalmas, M., Moffat, A., Ribeiro-Neto, B.A. (Eds.), Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9–13, 2015, ACM. pp. 373–382. URL: <https://doi.org/10.1145/2766462.2767738>, doi:10.1145/2766462.2767738.
- [21] Tay, Y., Luu, A.T., Hui, S.C., 2018. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics. pp. 1565–1575. URL: <http://aclweb.org/anthology/D18-1185>.
- [22] Wan, S., Lan, Y., Guo, J., Xu, J., Pang, L., Cheng, X., 2016. A deep architecture for semantic matching with multiple positional sentence representations, in: Schuurmans, D., Wellman, M.P. (Eds.), Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA., AAAI Press. pp. 2835–2841. URL: <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11897>.
- [23] Wang, Z., Hamza, W., Florian, R., 2017. Bilateral multi-perspective matching for natural language sentences, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 4144–4150. URL: <https://doi.org/10.24963/ijcai.2017/579>, doi:10.24963/ijcai.2017/579.
- [24] Zhu, X., Sobhani, P., Guo, H., 2015. Long short-term memory over recursive structures, in: Proceedings of International Conference on Machine Learning, pp. 1604–1612.