Sentence Matching With Deep Self-Attention and Co-Attention Features

No Author Given

No Institute Given

Abstract. Sentence matching refers to extracting the logical and semantic relation between two sentences which is widely applied in many natural language processing tasks such as natural language inference, paraphrase identification, and question answering. Many previous methods apply a siamese network to capture semantic features and calculate cosine similarity to represent sentences relation. However, these methods could be effective for overall rough sentence semantic but not sufficient for word-level matching information. In this paper, we propose a novel neural network based on attention mechanism which focuses on learning richer interactive features of two sentences. There are two complementary components in our model: semantic encoder and interactive encoder. Interactive encoder compares sentences semantic features which are encoded by semantic encoder. In addition, semantic encoder considers the output of interactive encoder as supplementary matching features. Experiments on three benchmark datasets prove that self-attention network and cross-attention network can efficiently learn the semantic and interactive features of sentences, which helps our method achieves state-of-the-art results.

Keywords: Sentence matching · Natural language processing · Neural network · Attention mechanism.

1 Introduction

Sentence matching requires a model to identify the logical relation between two sentences. It is a fundamental technology in natural language processing research area which has a wide range of practical applications such as natural language inference, question answering, paraphrase identification and so on. In natural language inference (also known as recognizing textual entailment) task [1], it is utilized to predict the reasoning relationship (entailment, contradiction, neutral) given premise sentence and hypothesis sentence. In paraphrase identification task [2], sentence matching needs to judge whether two sentences have the same meaning or not.

Recently, deep neural networks make progress in the field of natural language processing and become the most popular methods for sentence matching. There are two mainstream framework [3] in deep neural networks: sentences-encoding-based method and features-interaction-based method. The first method [4] is that

encodes each sentence to a fixed-length vector and uses the vectors to predict the relationship in a simple way such as cosine similarity or a feed-forward network . Another method [5] makes an improvement base on the first method and it could captures the interactive features while encoding the sentence. There is semantic gap [6] between the two sentences, which is a puzzle for determining the logical relationship without the interactive features.

Inspired by multi-head attention mechanism [7], we propose a model Deep Attention Matching Model (DAMM) for sentence matching task, which is constituted only by attention mechanism network, while many previous and powerful models [8–11] almost consist of deep convolutional neural network (CNN) or long short term memory (LSTM) network. Compared to convolutional neural network, attention mechanism network could extract the word order information, although convolutional neural network has achieved a huge success in compute vision field and it's widely utilized in natural language processing field recently. Compared to LSTM network, attention mechanism network has a stronger ability for long distance dependence because LSTM network has the multi-step multiply operation which may cause gradient vanishing. Base on the analysis, our model could have a more better result than the previous CNN-based or LSTM-based sentence matching models.

In DAMM, multi-head self-attention network is firstly employed for deep sentence semantic features. Then, multi-head cross-attention network is utilized for sentences interactive features with sentence semantic features as network's input. With semantic and interactive features, we design a alignment layer to integrate them by using feed-forward network, resnet[12] and layer-norm[13]. Furthermore, to achieve a better results, our model apply a stacked framework as shown in Figure 1. We will introduce more details of DAMM model in Section 3.

We evaluate the model on three sentence matching datasets: SNLI, SciTail, Quora Question Pairs (Quora). Experimental results show our model achieve the state-of-the-art performance.

In summary, our contributions are as follows:

- We only use attention mechanism in the encoder of the sentence matching model and achieve the state-of-the-art performance.
- Compared with previous interactive-based model, our model proposes multi-head cross-attention mechanism to capture more powerful interactive features.

2 Related Work

Early work of sentence matching mainly focus on conventional methods and small datasets, which works only on specific tasks [14]. Recently, many human annotated sentence pairs high quality datasets opened which make a big progress for sentence matching tasks. These datasets including SNLI [1], Quora Questions Pairs [2] and so on have contributed significantly to learning sentence semantic. In more details, SNLI is a dataset for natural language inference and Quora Questions Pairs is a dataset for paraphrase identification.

The development of deep learning algorithm makes natural language processing task to have more flexible and complex solving methods. As described in Section 1, sentences-encoding-based methods and features-interaction-based methods both are effective to sentence matching.

Sentences-encoding-based methods encode each sequence individually into a vector and then calculate cosine similarity or build a neural network classifier upon the two vectors. [4] proposes Deep Structured Semantic Models (DSSM) based on feed-forward neural networks. Compared to human-features-based methods, it is more automated and has a good performance. [15] and [11] apply recurrent networks and convolutional networks as their sequence encoder respectively which has a more powerful encoder than DSSM.

More recently, features-interaction-based methods consider that the interactive features could make a difference to the final prediction. [5] uses bidirectional LSTMs as encoders and employs a attention mechanism as interactive features collector. [2] interacts two sentences vectors from multi-perspective matching operation. [3] utilizes a deep convolutional network to extract interactive information.

3 Our Approach

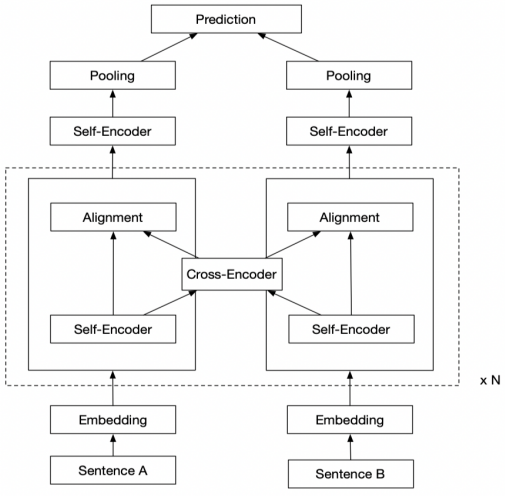


Fig. 1: Architecture of Deep Attention Matching Model (DAMM). Dashed frame including Self-Encoder, Cross-Encoder and Alignment could be repeated N times. Self-Encoder and Cross-Encoder respectively extract single sentence semantic information and interactive features between two sentences .

In this section, we introduce our proposed sentence matching neural networks Deep Attention Matching Model (DAMM) which is composed of the following major components: embedding layer, self-encoder, cross-encoder, alignment layer, pooling layer and prediction layer. Figure 1 shows the overall architecture of our model.

The input of model are two sentences as $A = (A_1, A_2, ..., A_I)$ with the length I and $B = (B_1, B_2, ..., B_J)$ with the length J where A_i is the i^{th} word of sentence A and B_j is the j^{th} word of sentence B. The sentence matching's goal is to give a label y to represent the relationship between sentence A and sentence B.

In DAMM, two sentences are firstly embedded by the embedding layer into a matrix. And then, N same-structured blocks encode the matrix where each block has a self-encoder, a cross-encoder and an alignment layer. The output of last block is fed into a self-encoder again to integrate the features and a pooling layer to get the final vector representation of the whole sentence. Finally, DAMM uses the two final vector representation as input and makes a prediction.

3.1 Embedding Layer

The goal of embedding layer is to represent each token of the sentence to a d-dimensional vector by using a pre-trained vector such as GloVe [16], Word2Vec [17] and Fasttext [18]. In our model, we use GloVe vector (840B Glove) to get the vector for sentence A and sentence B and the vector is fixed during training. Now, we have sentence A representation $E_a \in R^{l_a * d}$ and sentence B representation $E_b \in R^{l_b * d}$, where l_a refers to the max sequence length of sentence A, l_b refers to the max sequence length of sentence B and d refers to the dimension of word embedding.

3.2 Self-Encoder

The sentence A representation E_a and the sentence B representation E_b are fed into the Self-Encoder which is composed of a multi-head self-attention layer and a feed-forward layer to capture the richer semantic features of each sentence themselves.

Firstly, the multi-head self-attention network consists of query matrix Q_i, key matrix K_i and value matrix V_i. Each matrix respectively uses a linear transformation on the output of embedding layer representations E_a and E_b. Then, the scaled dot-product attention is employed to compute the self-attention output. Finally, we concatenate the multi-head self-attention outputs and feed the concatenate vector into a two layer feed-forward network with $gelu$ activative function. This process can be described by the following formulas and the formulas for H_b are similar and omitted here :

$$Q_i^s = A_i S_i^q \tag{1}$$

$$K_i^s = A_i S_i^k \tag{2}$$

$$V_i^s = A_i S_i^v \tag{3}$$

$$Att_i^s = softmax(\frac{Q_i^s (K_i^s)^T}{\sqrt{d_q}}) V_i^s \tag{4}$$

$$M_a = [Att_1^s; ...; Att_h^s] \tag{5}$$

$$H_a = gelu(M_a W_1^s) W_2^s \tag{6}$$

where h is number of the head of the multi-head self-attention network, i is an integer from 1 to h , d_q refers to the dimension of self-attention, $A_i \in R^{d_q * d_q}$ refers to hidden states, the projections are parameter matrices $S_i^q \in R^{d_q * d_q}$, $S_i^k \in R^{d_q * d_q}$, $S_i^v \in R^{d_q * d_q}$, $W_1^s \in R^{d * d'}$, $W_2^s \in R^{d' * d}$ where d' refers to intermediate hidden size, [...; ...]denotes the concatenation operation.

3.3 Cross-Encoder

In a sentence matching task, sentences interactive features could be important as same as the sentences semantic features generating by Self-Encoder above. Our model employs a Cross-Encoder to extract the sentences interactive features. The Cross-Encoder is the similar with the Self-Encoder but the key and value matrix are different. We calculate the interactive features from sentence A to sentence B as following, we omitted the another direction here:

$$Q_i^c = A_i C_i^q \tag{7}$$

$$K_i^c = B_i C_i^k \tag{8}$$

$$V_i^c = B_i C_i^v \tag{9}$$

$$CrossAtt_i^c = softmax(\frac{Q_i^c (K_i^c)^T}{\sqrt{d_q}}) V_i^c \tag{10}$$

$$M_{b2a} = [CrossAtt_1^c; ...; CrossAtt_h^c] \tag{11}$$

$$H_{b2a} = gelu(M_{b2a} W_1^c) W_2^c \tag{12}$$

where H_{b2a} denotes the interactive features from sentence A semantic features H_a to sentence B semantic features H_b, other parameters are similar with Self-Encoder.

3.4 Alignment Layer

After the Self-Encoder and Cross-Encoder, we have two features matrices H_a and H_{b2a} which respectively represents the sentence A semantic matrix and interactive matrix between two sentences. The two features both are significant components for a sentence matching task. We make an alignment for stacking operation and apply residual connection to avoid overfitting as following:

$$C_a = [H_a; H_{b2a}]W_a \tag{13}$$

$$H_a = H_a + E_a \tag{14}$$

where the projections are parameter matrices $W_a \in R^{2d*d}$.

3.5 Pooling Layer

The pooling layer's goal is to converts the matrices H_a and H_b to fixed-length vectors v_a and v_b which will be fed into prediction layer to classify. As we all know, both average and max pooling are useful strategies for sentence matching. Hence, we combine the max pooling strategy and mean pooling strategy in our model. Our experiments show that this leads to significantly better results. Formulations for v_b are similar and omitted here. This process is described by the following formulas:

$$v_a^{max} = \max_{i=1}^{l_a} H_{a,i} \tag{15}$$

$$v_a^{mean} = \sum_{i=1}^{l_a} \frac{H_{a,i}}{l_a} \tag{16}$$

$$v_a = [v_a^{max}; v_a^{mean}] \tag{17}$$

3.6 Prediction Layer

In our model, v_a and v_b are the sentence A and sentence B feature vectors from the output of the pooling layer. The prediction layer is to aggregate the vectors v_a and v_b in a proper way, and then predicts the label by using a feed-forward neural network.

Firstly, similarity and difference between two sentences are meaningful features for a symmetric task. Hence, we aggregates v_a and v_b in various ways as follows:

$$v = [v_a; v_b; v_a - v_b; v_a * v_b] \tag{18}$$

where $-$, $*$ are the element-wise subtraction and element-wise product.

Then, with the aggregated features v, we employ a two-layer feed-forward neural network for classification task and *gelu* activative function is adopted

after the first layer. Finally, we use multi-class cross-entropy loss function with Label Smooth Regularization (LSR) [19] to train our model.

$$\hat{y} = softmax(gelu(vW_1^o)W_2^o) \tag{19}$$

$$y = \hat{y}(1-\epsilon) + \frac{\epsilon}{C} \tag{20}$$

$$Loss = -\sum_{j=1}^{C} y_j log(\hat{y_j}) + \lambda \sum_{\theta \in \Theta} \theta^2 \tag{21}$$

where feature vector is $v \in R^{1*d}$, the projections are parameters $W_o^1 \in R^{d*d'}$, $W_o^2 \in R^{d'*C}$ and C is the number of label classes, y is the ground truth, hyper-parameter ϵ denotes the degree of smooth of LSR, θ denotes the parameters of DAMM.

4 Experiments

We conduct experiments on three sentence matching benchmark datasets: SNLI, Scitail, Quora Question Pairs (Quora).

SNLI (The Stanford Natural Language Inference corpus) is a popular benchmark dataset for natural language inference. It focuses on three basic relationships between a premise and a hypothesis: entailment(the premise entails the hypothesis), contradiction(the premise and the hypothesis contradict), neutral(the premise and the hypothesis are not related).

SciTail is a textual entailment dataset from science question answering. The premises and hypothesis in SciTail are different from existing entailment datasets. The hypothesis is generated from science questions and the corresponding answer candidates, and the premises are retrieved from a large corpus. The generated way of SciTail make it more challenging.

Quora Question Pairs is a dataset for paraphrase identification provided by Quora. This task is a binary classification task which need to determine whether one question is a paraphrase of another.

4.1 Implementation Details

In our experiments, word embedding vectors are initialized with 300d GloVe vectors pre-trained from the 840B Common Crawl corpus. Embeddings of out of the vocabulary of GloVe is initialized to zeros. All embeddings are fixed during the training. All other parameters are initialized with a normal distribution where *mean* is 0.0 and *standard deviation* is 0.02. Dropout with a keep probability of 0.8 is applied after the word embedding layer and every fully-connected layer. We also apply attention dropout with a keep probability of 0.8 after the attention operation of Self-Encoder and Cross-Encoder. The hidden size is 300 in all experiments. Activative functions in all feed-forward networks are *gelu*

function. After the residual connections and two-layer feed-forward networks, we use LayerNorm with a norm epsilon of $1e-12$ to accelerate training model. Adam optimizer with weight decay of 0.01 is employed in our model. Learning rate is tuned from 0.00001 to 0.0005 and an exponentially decaying learning rate with a linear warmup is applied for learning rate.

Table 1: Classification accuracy (%) on SNLI test set.

Model	**Acc. (%)**
BiMPM[2]	86.9
ESIM[5]	88.0
DIIN[3]	88.0
DRCN[10]	88.9
RE2[20]	**88.9**
DAMM(ours)	88.8
BiMPM(ensemble)	88.8
DIIN (ensemble)	88.9
DRCN (ensemble)	**90.1**
RE2(ensemble)	89.9
DAMM(ensemble)	**90.1**

4.2 Results on SNLI and SciTail

We evalutated our model on the natural language inference task over SNLI and SciTail datasets. Results on SNLI and SciTail are listed in Table 1 and 2. Our method obtains a performance which achieves state-of-the-art results. For SNLI dataset, our method get a accuracy score 88.8% in single model and 90.1% by ensemble in test dataset which obtains a state-of-the-art performance. We employ 8 different randomly initialized models with same hyper-parameters for our ensemble approach. For SciTail dataset, we obtain a result nearly the most highest performance. SciTail dataset is a more difficult and challenging task for natural language inference, because it has only 27k samples while SNLI has 570k samples.

4.3 Results on Quora Question Pairs

The results on Quora Qustion Pairs are shown in Table 3. Most methods such as BiMPM and DIIN, apply attention method for features alignment after bi-directional long short-term memory or convolutional neural network encoder. However, DAMM abandons complex encoder methods and uses a stacked structure based on simple attention mechanism. The performance of our model is on par with the state-of-the-art on this dataset.

Table 2: Classification accuracy (%) on Scitail test set.

Model	**Acc. (%)**
ESIM[5]	70.6
DecompAtt[8]	72.3
DGEM[24]	77.3
HCRN[21]	80.0
CAFE[22]	83.3
RE2[20]	**86.0**
DAMM(ours)	85.7

Table 3: Classification accuracy (%) on Quora test set.

Model	**Acc. (%)**
BiMPM[2]	88.2
DIIN[3]	89.1
MwAN[11]	89.1
CSRAN[23]	89.2
RE2[20]	89.2
DAMM(ours)	**89.4**

4.4 Analysis

Ablation Study We conducted an ablation study of our model for 7 ablation baselines: (1) replaced fix-embedding with trainable-embedding in embedding layer, (2) removed Cross-Encoder in every block, (3) replaced Self-Encoder with LSTM network in every block, (4) removed Cross-Encoder and replaced Self-Encoder with LSTM network in every block, (5) removed residual connections in alignment layer (Equtation 14), (6) removed the last Self-Encoder before pooling layer, (7) removed difference and similarity features between two sentences ($v_a - v_b$ and $v_a * v_b$ in Equtation 18). Ablation study is conducted on the test set of SNLI and alation experiments results are shown in Table 4.

We compared the difference of fix embedding and trainable embedding in embedding layer in ablation experiment (1). The result shows that fix embedding is more effective, and we think trainable embedding may be easier to overfit than fix embedding. In (2), we verified the effectiveness of interactive features which is captured by Cross-Encoder. Without Cross-Encoder, DAMM becomes a siamese network and its performance decreases significantly. In (3), we replaced Self-Encoder with LSTM network. It means that different encoding ways have a marked impact of the model. The result shows that attention mechanism works well for sentence matching task. The result of experiment (4) could be a supplement to experiments (2-3). The result of ablation experiment (5) demonstrate that residual connection is a key component of alignment layer. With the residual

connection, DAMM has more powerful capability to aggregate semantic features and interactive features. In (6), we applied Self-Encoder to integrate semantic and interactive features rather than the features are fed into pooling layer. The result shows the last Self-Encoder before pooling layer is necessary. The result of experiment (7) show that difference and similarity features are important for sentence matching task.

Table 4: Ablation study on the SNLI test set.

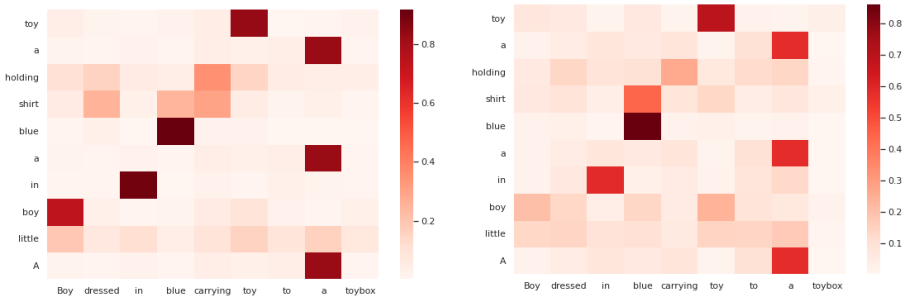
Model	**Acc. (%)**
DAMM	88.8
(1) − Fix. + Tr.	88.4
(2) − Cross.	86.3
(3) − Self. + LSTM	85.5
(4) − Self. + LSTM − Cross.	84.7
(5) − RES	87.6
(6) − Last Self.	87.2
(7) − Symmetry	88.3

Case Study In this section, we used a premise *"A little boy in a blue shirt holding a toy"* and a hypothesis *"Boy dressed in blue carrying toy to a toybox"* from SNLI test set as a case study. As show in Fig. 2, we visualized the attention weights in the first and last Cross-Encoder between the premise and hypothesis. There are multi-head cross-attention in Cross-Encoder, and multi head could obtain more information from different perspectives. Because each head in Cross-Encoder has its own attention weights, our attention weights in visualization are calculated by concatenating all head attention weights and the relation between words is represented by consine similarity.

From Fig. 2(a), we can see that the word **"blue"** of hypothesis is highly related to the phrase **"blue shirt"** of premise. In the first block of DAMM, our model mainly pays attention to the word-level interaction. But as in Fig. 2(b), the attention weights between the word **"blue"** of hypothesis and the phrase **"A little boy"** were increased obviously, which proves our model is able to take into consideration of the whole sentence-level semantic and the interaction between the premise and hypothesis.

5 Conclusions and Future Work

In this paper, we propose a novel attention-based network for semantic matching. We align the semantic features and interactive features which both are captured



(a) Attention weight results in the first block (b) Attention weight results in the last block

Fig. 2: A case study of the natural language inference task. The premise is "A little boy in a blue shirt holding a toy", and the hypothesis is "Boy dressed in blue carrying toy to a toybox".

from attention mechanism. The alignment features have sufficient context information towards the two sentences. Our model achieves the state-of-the-art performance on most of the datasets of highly challenging natural language tasks.

For future work, we will explore how to introduce external knowledge to improve performance.

References

1. Bowman, Samuel R and Angeli, Gabor and Potts, Christopher and Manning, Christopher D.: A large annotated corpus for learning natural language inference. 2015
2. Wang, Zhiguo and Hamza, Wael and Florian, Radu.: Bilateral multi-perspective matching for natural language sentences. 2017
3. Gong, Yichen and Luo, Heng and Zhang, Jian.: Natural language inference over interaction space. 2017
4. Huang, Po-Sen and He, Xiaodong and Gao, Jianfeng and Deng, Li and Acero, Alex and Heck, Larry.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp. 2333–2338. 2013
5. Chen, Qian and Zhu, Xiaodan and Ling, Zhenhua and Wei, Si and Jiang, Hui and Inkpen, Diana.: Enhanced lstm for natural language inference. 2016
6. Liu, Pengfei and Qiu, Xipeng and Chen, Jifan and Huang, Xuan-Jing.: Deep fusion lstms for text semantic matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1034–1043. 2016
7. Vaswani, Ashish and Shazeer, Noam and Parmar, Niki and Uszkoreit, Jakob and Jones, Llion and Gomez, Aidan N and Kaiser, Łukasz and Polosukhin, Illia.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008. 2017
8. Parikh, Ankur P and Täckström, Oscar and Das, Dipanjan and Uszkoreit, Jakob.: A decomposable attention model for natural language inference. 2016

9. Liu, Xiaodong and Duh, Kevin and Gao, Jianfeng.: Stochastic answer networks for natural language inference. 2018
10. Kim, Seonhoon and Kang, Inho and Kwak, Nojun.: Semantic sentence matching with densely-connected recurrent and co-attentive information. In: Proceedings of the AAAI conference on artificial intelligence, pp. 6586–6593. 2019
11. Tan, Chuanqi and Wei, Furu and Wang, Wenhui and Lv, Weifeng and Zhou, Ming.: Multiway Attention Networks for Modeling Sentence Pairs. In: IJCAI, pp. 4411–4417. 2018
12. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778. 2016
13. Ba, Jimmy Lei and Kiros, Jamie Ryan and Hinton, Geoffrey E.: Layer normalization. 2016
14. Romano, Lorenza and Kouylekov, Milen and Szpektor, Idan and Dagan, Ido and Lavelli, Alberto.: Investigating a generic paraphrase-based approach for relation extraction. In: 11th Conference of the European Chapter of the Association for Computational Linguistics. 2006
15. Conneau, Alexis and Kiela, Douwe and Schwenk, Holger and Barrault, Loic and Bordes, Antoine.: Supervised learning of universal sentence representations from natural language inference data. 2017
16. Pennington, Jeffrey and Socher, Richard and Manning, Christopher D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543. 2014
17. Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg S and Dean, Jeff.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems, pp. 3111–3119. 2013
18. Joulin, Armand and Grave, Edouard and Bojanowski, Piotr and Mikolov, Tomas.: Bag of tricks for efficient text classification. 2016
kim2014convolutional Kim, Yoon.: Convolutional neural networks for sentence classification. 2014
19. Szegedy, Christian and Vanhoucke, Vincent and Ioffe, Sergey and Shlens, Jon and Wojna, Zbigniew.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826. 2016
20. Yang, Runqi and Zhang, Jianhai and Gao, Xing and Ji, Feng and Chen, Haiqing.: Simple and effective text matching with richer alignment features. 2019
21. Tay, Yi and Luu, Anh Tuan and Hui, Siu Cheung.: Hermitian Co-Attention Networks for Text Matching in Asymmetrical Domains. In: IJCAI, pp. 4425–4431. 2018
22. Tay, Yi and Tuan, Luu Anh and Hui, Siu Cheung.: Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. 2017
23. Tay, Yi and Tuan, Luu Anh and Hui, Siu Cheung.: Co-stack residual affinity networks with multi-level attention refinement for matching text sequences. 2018
24. Khot, Tushar and Sabharwal, Ashish and Clark, Peter.: SciTaiL: A Textual Entailment Dataset from Science Question Answering. In: AAAI, pp. 41–42. 2018