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

1st Danfeng Yan, 2nd Zhipeng Wang
State Key Laboratory of Networking and Switching Technology
Beijing University of Posts and Telecommunications
Beijing, China
{yandf, wangzhipeng}@bupt.edu.cn

3rd Wenting Shi
Department of Electrical and Computer Engineering
University of California, San Diego
San Diego, United States
wentingshi10@gmail.com

***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. However, many previous methods simply use a siamese network to capture the semantic features and apply attention mechanism to align the semantic features of two sentences. In this paper, we propose a deep and effective neural network based on attention mechanism to learn richer semantic features and interactive features of two sentences. Each layer of our model include two sub-layers semantic encoder and interactive encoder of which one uses a self-attention network for the semantic features and another one uses a cross-attention network for the interactive feature. Experiments on three benchmark datasets prove that self-attention network and cross-attention network can efficiently learn the semantic and interactive features of two sentences, which helps our method achieves state-of-the-art results.**

***Index Terms*—sentence matching, natural language processing, attention mechanism**

I. INTRODUCTION

Sentence matching requires a model to identify the logical relationship between two sentences. It is a fundamental technology in natural language processing research area which have 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 some progress in the field of natural language processing and become the most favorite methods for sentence matching. There are two mainstream framework [**3**] in deep neural networks: sentences-encoding-based method, features-interaction-based method . The first method is to encode each sentence to a fixed-length vector and use the vectors to predict the relationship in a simple way [**4**] such as cosine similarity or two-layer feed-forward network . Another method makes an improvement base on the first method and it captures the interactive features while encoding the sentence [**5**]. There is semantic gap between two sentences, which is a puzzle for determining the logical relationship without the interactive features [**6**].

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. However, many powerful models [**8**]-[**11**] must consists of deep Convolutional Neural Network (CNN) or Long Short-Term Memory (LSTM) Network .

CNN has achieved a huge success in compute vision area, and it's widely utilized in natural language process recently. However, attention mechanism network could extract the word order information while CNN can't. Compared to LSTM network, attention mechanism network has a stronger ability for long distance dependence because LSTM 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 before 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 and LayNorm. Furthermore, to achieve a better results, our model apply a stacked framework as shown in Figure 1. We will introduce 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. For the convenience of researcher. We opened our source code on github. [1]

In summary, our contributions are as follows:

- To the best of our knowledge, we are the first to only use attention mechanism in the encoding process of the sentence matching model.
- Compared with interaction-based model, our model apply multi-head cross attention mechanism to capture more

Key Project of Chinese National Programs for Fundamental Research and Development (973 Program)

[1] https://github.com/2hip3ng/Deep-Attention-Matching-Model

interactive features and achieve the state-of-the-art per formance.

II. RELATED WORK

Early work of sentence matching mainly focus on con ventional methods and small datasets, which works only o specific tasks [**12**]. Recently, many human annotated sentenc pairs high quality datasets opened which make a big progres for sentence matching tasks. These datasets including SNL [**1**], Quora Questions Pairs [**2**] and so on have contribute significantly to learning semantics as well. In more details SNLI is a dataset for natural language inference and Quor Questions Pairs is a dataset for paraphrase identification.

The developments of deep learning algorithm make natura language process task to have more flexible and complex solv ing methods. As described in Section 1, sentences-encoding based method and features-interaction-based method both ar effective to sentence matching.

Sentences-encoding-based method encodes each sequenc individually into a vector and then or calculate cosine similar ity or build a neural network classifier upon the two vectors Huang et al. [**4**] propose a Deep Structured Semantic Model (DSSM) based on feed-forward neural networks. Compare to human-features-based methods, it is more automated an have a good performance. Alexis Conneau et al. [**13**] and Tan et al. [**11**] use recurrent networks and convolutional networks as their sequence encoder respectively which have a more powerful encoder than DSSM.

More recently, the second method features-interaction-based consider the cross features or interactive features which could make a difference to the final prediction. ESIM [**5**] uses bidirectional LSTMs as encoders and employs a similar attention mechanism as interactive features collector. BiMPM [**2**] interacts two sentences vectors from multi-perspective matching operation. DIIN [**3**] utilizes a deep convolutional network to extract alignment information.

III. OUR APPORACH

In this section, we introduce our proposed sentence matching networks Deep Attention Matching Model (DAMM) which are composed of the following major components: embedding layer, self-encoder, cross-encoder, alignment layer, pooling layer, 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 a length I and $b = (b_1, b_2, ..., b_J)$ with a 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, each sentence are first embedded by the embedding layer into a matrix. And then, N same-structured blocks encoder the matrix. Each block has a self-encoder, cross-encoder and alignment layer. The output of last block is feeded into self-encoder again to integrate the features and a pooling layer to get the final represent of the whole sentence. Finally, DAMM use the two vectors as input and predicts the final target.

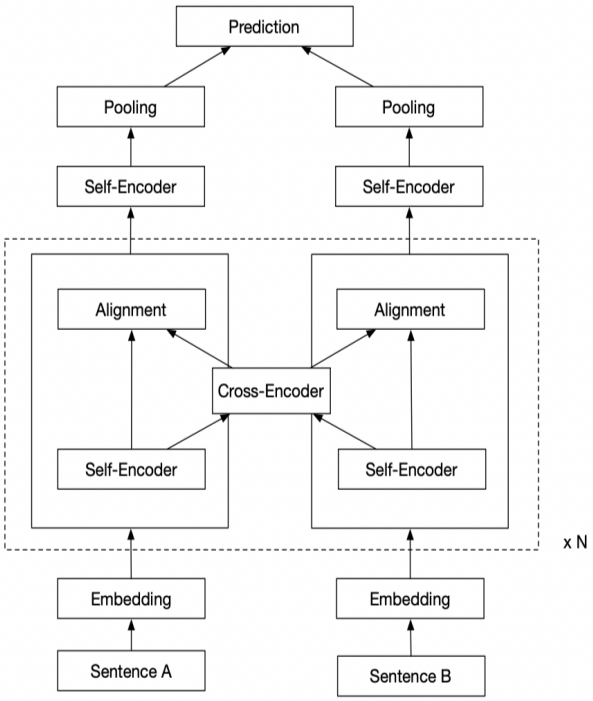


Fig. 1: Architecture of Deep Attention Matching Model. Dashed frame including Self-Encoder, Cross-Encoder and Alignment could be multiple N times. Self-Encoder and Cross-Encoder denote that extract semantic information and interactive features respectively.

A. 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 [**14**], Word2Vec [**15**] and Fasttext [**16**]. In our model, we use GloVe vector (840B Glove) to get the fixed vector for $sentence\ a$ and $sentence\ b$ and the vector is fixed during training. Now, we have $sentence\ a$ representation $A \in R^{la*d}$ and $sentence\ b$ representation $B \in R^{lb*d}$, where la refers to the max sequence length of $sentence\ a$, lb refers to the max sequence length of $sentence\ b$.

B. Self-Encoder

In the Self-Encoder, the $sentence\ a$ representation A and the $sentence\ b$ representation B are passed through multi sub-layer networks which are composed of a multi-head self-attention layer and a feed-forward layer to capture the richer semantic features of each sentence themselves.

First, the input of the multi-head self-attention network consists of queries matrix Q_i, keys matrix K_i and values matrix V_i which are respectively using a linear transform on the representation A. 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

into a two layer feed-forward network with $gelu$ activation functions. Formulations for H_B are similar and omitted here. This process is described by the following formulas:

$$Q_i^S = AW_i^{QS} \tag{1}$$

$$K_i^S = AW_i^{KS} \tag{2}$$

$$V_i^S = AW_i^{VS} \tag{3}$$

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

$$M_A = [Att_1^S; Att_2^S; ...; Att_h^S] \tag{5}$$

$$H_A = gelu(M_A W_1^S)W_2^S \tag{6}$$

where h is number of the head of the multi-head self-attention network, i is an integer from 1 to h , the projections are parameter matrices $W_i^{QS} \in R^{d*d_q}$, $W_i^{KS} \in R^{d*d_k}$, $W_i^{VS} \in R^{d*d_v}$, $W_1^S \in R^{d*d'}$, $W_2^S \in R^{d'*d}$, $[;]$denotes the concatenation operation.

C. Cross-Encoder

In a sentence matching model, the sentences interaction features could be important as same as the sentences semantic features generating by Self-Encoder above. For the sentences interaction features, our model employ a Cross-Encoder to extract. The Cross-Encoder is the similar with the Self-Encoder but the key and value matrix is different. We calculate the interaction features from $sentence\ a$ to $sentence\ b$ as following, we omitted the another direction here:

$$Q_i^C = AW_i^{QC} \tag{7}$$

$$K_i^C = BW_i^{KC} \tag{8}$$

$$V_i^C = BW_i^{VC} \tag{9}$$

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

$$M_{B2A} = [Att_1^C; Att_2^C; ...; Att_h^C] \tag{11}$$

$$H_{B2A} = gelu(M_{B2A} W_1^C)W_2^C \tag{12}$$

where H_{B2A} denotes the interaction features from $sentence\ a$ semantic features H_A to $sentence\ b$ semantic features H_B, other parameters are similar to Self-Encoder.

D. Alignment Layer

After the Self-Encoder and Cross-Encoder, we have two features matrix H_A and H_{B2A} which respectively represent the $sentence\ a$ semantic matrix and interactive matrix. The two features both are significant components for a sentence matching task. However,for continue stacking operation, we need to align the two matrix as following :

$$C_A = [H_A; H_{B2A}; H_A - H_{B2A}; H_A * H_{B2A}] \tag{13}$$

$$E_A = C_A W_A \tag{14}$$

$$H_A = H_A + E_A \tag{15}$$

where $-$, $*$ are the element-wise subtraction and element-wise product, the projections are parameter matrices $W_A \in R^{4d*d}$.

E. Pooling Layer

The pooling layer's goal is to converts the vectors H_A and H_B to fixed-length vector 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. We also consider that some key words in two sentences may have an important impact for the final classification, and TextCNN [**17**] is a good way to extract key words features. Hence, we combine the max pooling strategy and TextCNN 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:

$$v_a^{max} = \max_{i=1}^{la} H_{A,i} \tag{16}$$

$$v_a^{cnn} = TextCNN(H_A) \tag{17}$$

$$v_a = [v_a^{max}; v_a^{cnn}] \tag{18}$$

where operation $TextCNN$ has a detail explanation in TextCNN [**17**].

F. Prediction Layer

In our models, v_a and v_b are the $sentence\ a$ and $sentence\ b$ features vectors from the output of the pooling layer. The prediction layer is to aggregate the v_a and v_b in a proper way, and then predict the label using a feed-forward neural network. We first aggregate v_a and v_b in various ways which are useful for a symmetric task as follows:

$$v = [v_a; v_b; v_a - v_b; v_a * v_b] \tag{19}$$

Finally, with the aggregated features v, we employ a two-layer feed-forward neural network for classification task and $gelu$ activation function is adopted after first layer. We use multi-class cross-entropy loss function with Label Smooth Regularization (LSR) [**18**] to train our model.

$$\hat{y} = softmax(gelu(vW_1^o)W_2^o) \tag{20}$$

TABLE I: DETAILS OF THE DATASETS. SAMPLES NUMBER OF TRAIN AND TEST DATASETS, SENTENCES LENGTH OF SAMPLES

Datasets		SNLI	SciTail	Quora
Category 1	train	183416	8602	245042
	test	3368	842	5000
Category 2	train	182764	14994	139306
	test	3329	1284	5000
Category 3	train	183187	-	-
	test	3278	-	-
Sentence a average lengths	train	12.87	18.51	12.56
	test	13.94	18.16	12.28
Sentence b average lengths	train	7.4	12.23	12.82
	test	7.5	13.01	12.50

$$y = y(1-\epsilon) + \frac{\epsilon}{C} \tag{21}$$

$$Loss = -\sum_{j=1}^{C} y_j log(\hat{y_j}) + \lambda \sum_{\theta \in \Theta} \theta^2 \tag{22}$$

where features vectors is $v \in R^{1*d}$, the projections are parameters $W_{o1} \in R^{d*d'}$, $W_{o2} \in R^{d'*C}$ 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.

IV. EXPERIMENTS

We conduct experiments on three sentence matching benchmark datasets: SNLI, Scitail, Quora Question Pairs (Quora). SNIL and Scitail are for natural language inference, Quora Question Pairs is for paraphrase identification. We show some details of three datasets in Table I about samples number of train and test datasets, average lengths of $sentencea$ and $sentenceb$.

A. Datasets Details

SNLI (The Stanford Natural Language Inference corpus) is a popular benchmark dataset for natural language inference. It focuses on three basic relationships betweent a premise and a hypotheis: entailment(the premise entails the hypothesis), contradiction(the premise and the hypothesis contradict), neutral(the premise and the hypothesis are not related). The original SNLI dataset contains a special category "-", which indicates the annotators cannot reach an agreement. As in the related work, we remove this category. We used the same split as in the original paper and other previous work. SNLI have 570k human annotated samples as show in Table I. Category 1-3 respectively denote three relations entailment, contradiction and neutral.

SciTail is a textual entailment dataset from science question answering. The premises and hypotheses in Scitail are different from existing entailment datasets. The hypotheses 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. SciTail have 27k samples as show in Table I. Category 1-2 respectively denote two relations entailment and neutral.

Quora Question Pairs is a dataset for paraphrase identification provided by Quora. This task is a binary classification to determine whether one question is a paraphrase of another. Quora Question Pairs have about 400k question pairs as show in Table I. Category 1-2 respectively denote the pairs with same means or not.

TABLE II: HYPER PARAMETERS DETAILS ON THREE DATASETS

Hyper Parameters	SNLI	SciTail	Quora
Batch Size	1024	512	512
Learning Rate	0.001	0.00018	0.00015
Hidden Layers	4	4	4
Max Length of Sentence a	30	30	25
Max Length of Sentence b	30	30	25
Parameter ϵ of Label Smooth	0	0.1	0.1

B. 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 which $mean$ 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 applied attention dropout with a keep probability of 0.8 after the attention operation of Self-Encoder and Cross-Encoder. The hid- den size is set to 300 in all experiments. Activations in all feed-forward networks are $gelu$ activations. After the residual connections and two-layer feed-forward networks, we use a 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. We employed 8 different randomly initialized models with same hyper-parameters for our ensemble approach. The other hyper-parameters for different datasets are listed in Table II.

C. Results on SNLI and SciTail

We evalutated our model on the natural language inference task over SNLI and SciTail task. Results on SNLI and SciTail are listed in Tabel III and Tabel IV. 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

TABLE III: Classification accuracy (%) on SNLI test set.

Model	Acc. (%)
BiMPM	86.9
ESIM	88.0
DIIN	88.0
DRCN	88.9
RE2	**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**

TABLE IV: Classification accuracy (%) on Scitail test set.

Model	Acc. (%)
ESIM	70.6
DecompAtt	72.3
DGEM	77.3
HCRN	80.0
CAFE	83.3
RE2	**86.0**
DAMM(ours)	85.7

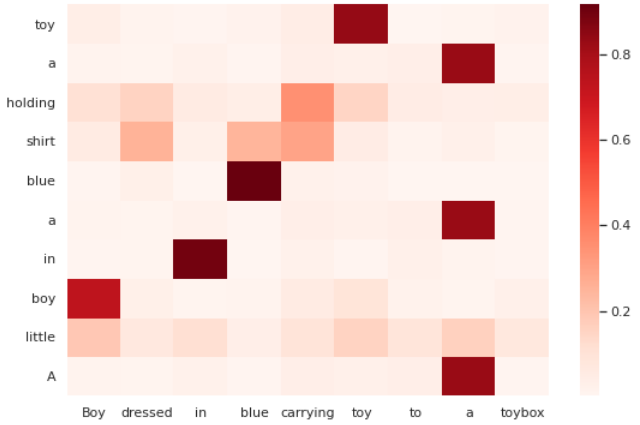
model and 90.1% by ensemble in test dataset which obtains a state-of-the-art performance. For SciTail dataset, we obtains a result nearly the most highest performance. SciTail dataset is a more difficult and challenging task for natural language inference, because it has only 27k samples while SNLI has 570k samples.

D. Results on Quora Question Pairs

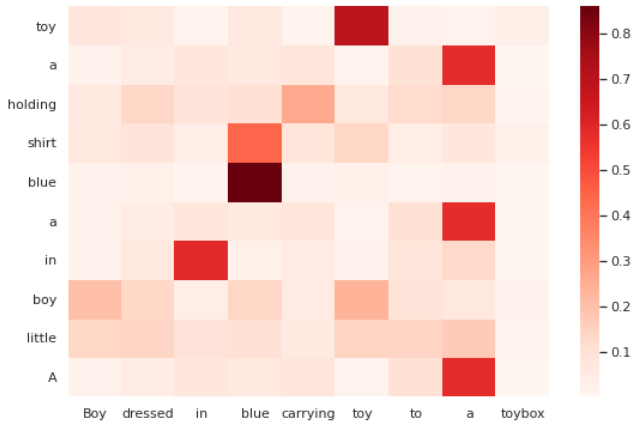
The results on Quora Qustion Pairs are shown in Tabel V. Most methods such as BiMPM and DIIN, apply attention method for features alignment after bi-directional long short-term memory or convolutional neural network encoder. However, DAMM abandons complex encoder methods and uses a stacked structure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

TABLE V: Classification accuracy (%) on Quora test set.

Model	Acc. (%)
BiMPM	88.2
DIIN	89.1
MwAN	89.1
CSRAN	89.2
SAN	**89.4**
RE2	89.2
DAMM(ours)	**89.4**



(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".

E. Analysis

Ablation Study We conducted an ablation study of our model for 5 ablation baselines:(1) remove TextCNN in pooling layer, (2) remove ResNet in Cross-Encoder (Equtation 15), (3) remove TextCNN in pooling layer and remove ResNet in Cross-Encoder. The ablation study is conducted on the test set of SNLI. Ablation experiments results are shown in Tabel VI.

Firstly, we verified the effectiveness of TextCNN in pooling layer in ablation experiment (1). Only using a max-pooling may extract limited information for sentence matching. The results of ablation experiment (2) demonstrate that residual connection is a key component of Cross-Encoder. With the residual connection, DAMM has more powerful capability to aggregate semantic features and interactive feature. The results of ablation experiment (3) also show that TextCNN in pooling layer and residual connection in Cross-Encoder make a great contribution for the whole model.

Case Study

TABLE VI: Ablation study on the SNLI dev sets.

Model	Acc. (%)
DAMM	88.8
− CNN	87.9
− RES	87.6
− CNN − RES	87.2

In this section, we use 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 visualize the attentive weights in the first and last Cross-Encoder between premise and hypothesis.

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 of the premise and hypothesis.

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

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

VI. ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China under Grant 2018YFC0831502.

REFERENCES

[1] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. "A large annotated corpus for learning natural language inference." In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

[2] Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. "Bilateral multi-perspective matching for natural language sentences". In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pages 4144–4150.

[3] Gong, Y., Luo, H., and Zhang, J. 2018. "Natural language inference over interaction space." In International Conference on Learning Representations.

[4] Po-Sen Huang Xiaodong He Jianfeng Gao. 2013. "Learning Deep Structured Semantic Models for Web Search using Clickthrough Data." In ACM International Conference on Information and Knowledge Management (CIKM).

[5] Qian Chen, Xiaodan Zhu, Zhenhua Ling. 2017. "Enhanced LSTM for Natural Language Inference." In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.

[6] Liu, P., Qiu, X., Chen, J., and Huang, X. 2016. "Deep fusion lstms for text semantic matching." In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, 1034–1043.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar. 2017. "Attention Is All You Need." arXiv, cs.CL, 1706.03762.

[8] Ankur Parikh, Oscar, Dipanjan Das, and Jakob Uszkoreit. 2016. "A decomposable attention model for natural language inference." In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2249–2255, Austin, Texas. Association for Computational Linguistics.

[9] Xiaodong Liu, Kevin Duh, and Jianfeng Gao. 2018. "Stochastic answer networks for natural lan guage inference." Computing Research Repository, arXiv:1804.07888.

[10] Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2018. "Semantic sentence matching with densely-connected recurrent and co-attentive information". Computing Research Repository, arXiv:1805.11360. Version 2.

[11] Chuanqi Tan, Furu Wei, Wenhui Wang, Weifeng Lv, and Ming Zhou. 2018. "Multiway attention networks for modeling sentence pairs." In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 4411–4417. International Joint Conferences on Artificial Intelligence Organization.

[12] Romano,L.,Kouylekov,M, Szpektor,I.,Dagan, I., and Lavelli, A. 2006. "Investigating a generic paraphrase- based approach for relation extraction." In 11th Conference of the European Chapter of the Association for Computational Linguistics.

[13] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. 2017. "Supervised learning of universal sentence representations from natural language inference data." arXiv preprint arXiv:1705.02364.

[14] Pennington, J., Socher, R., and Manning, C. D. 2014. "Glove: Global vectors for word representation." In Empirical Methods in Natural Language Processing (EMNLP), 1532–1543.

[15] Mikolov,T.,Sutskever,I.,Chen,K.,Corrado, G. S., and Dean, J. 2013. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, 3111–3119.

[16] Armand Joulin, Edouard Grave, Piotr Bojanowski and Tomas Mikolov. 2016. "Bag of Tricks for Efficient Text Classification." arXiv, cs.CL, 1607.01759

[17] Yoon Kim. "Convolutional neural networks for sentence classification." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[18] Szegedy C, Vanhoucke V, Ioffe S, et al. "Rethinking the inception architecture for computer vision." In Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.