**Academic Journal of Science and Technology**
ISSN: 2771-3032 | Vol. 3, No. 2, 2022

# Lightweight Text Matching Method

**Qingyu Li, Yujun Zhang***

School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China
* Corresponding author

**Abstract:** This paper constructs a lightweight text matching model. This model fully extracts the features of two text through the fusion of double tower and interactive methods, and then conducts deep interaction. Then, it classifies the last extracted text features through the classification network, and conducts training, validation and testing on the Chinese text matching dataset LCQMC. Although the experimental results show that the model in this paper is relatively good, it still needs to be improved.

**Keywords:** Text matching, NLP, Deep learning.

## 1. Introduction

As a basic and core task of natural language processing [1], text matching can be applied to many fields such as information retrieval, retelling discrimination, etc. Nowadays, with the rapid development of deep learning, traditional machine learning has been replaced. More researchers are committed to studying efficient text matching models for deep learning. For example, the DSSM model [2] proposed by Huang et al. is the first research result of deep learning on text matching. This model provides a good research idea for later researchers, Later, with the popularity of CNN and RNN, CNN-DSSM model [3] and LSTM-DSSM model [4] were developed. Because these models did not pay attention to the deep interaction between texts, researchers later shifted their focus to how to obtain the interaction information between texts, and developed many excellent models, such as MatchPyramid model [5], ESIM model [6] and BiMPM model [7]. These models provide a good idea for our research, but from the current background, the model performance is not excellent, the semantic information of text cannot be extracted completely. With the rise of the large pretraining language model BERT [8], pretraining and fine tuning have become the mainstream research paradigm, but the large pretraining model has large parameters and long training time. How to find a compromise way to get a model with high performance and fast speed has become the focus of research. This paper proposes a lightweight text matching method, which takes less training time than large pretraining models.

## 2. Our Method

This paper refers to the design idea of RE2 framework [9], uses BERT model to obtain the word embedding of two texts, and then respectively input the two texts into the encoder layer to encode the text, and then interactively align the features obtained from the Encoder layer, then fuse the features through the Fusion layer, and finally conduct classification prediction after pooling. The overall frame diagram is shown in figure 1.

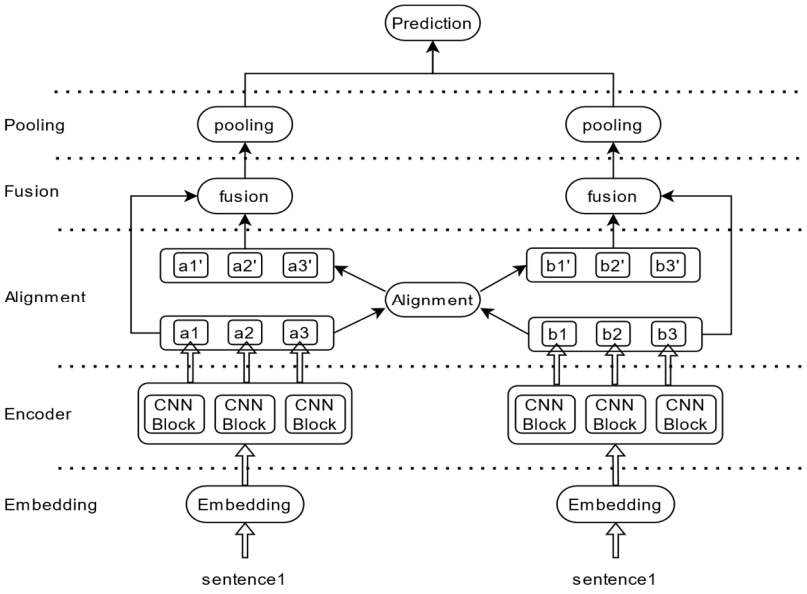


**Figure 1.** Overall Frame Diagram

### 2.1. Encoder

In this paper, the Encoder layer uses one-dimensional convolutions with convolution kernels of 2, 3 and 4 to encode the word Embedding Ea and Eb obtained from the Embedding layer to obtain text features a1, a2, a3, b1, b2, and b3 of

different scales. The calculation formula is as follows:

$$a_i = conv(E_a, k_i) \quad (1)$$

$$b_i = conv(E_b, k_i) \quad (2)$$

Where conv represents one-dimensional convolution operation, and k represents the size of convolution kernel.

## 2.2. Alignment

Align the features of the two pieces of text obtained from the Encoder layer. First, construct the interaction matrix attn of the two pieces of text, and then perform the softmax operation on the interaction matrix respectively, and then multiply the corresponding text feature matrix to obtain the aligned matrix. As the Encoder layer uses three different convolution kernels to extract features from two pieces of text, we align the features corresponding to two pieces of text to get align_ a1, align_ a2, align_ a3, align_ b1, align_ b2, align_ b3, the calculation formula is as follows:

$$attn_i = a_i b_i^T * t_i \quad (3)$$

$$align\_a_i = soft\max(attn_i) * b_i \quad (4)$$

$$align\_b_i = soft\max(attn_i)^T * a_i \quad (5)$$

## 2.3. Fusion

The feature matrix obtained from the alignment layer and the feature matrix information obtained from the Encoder layer are fused to obtain multiple semantic features. After the two matrices are spliced, they pass through the full connection layer. Here we choose three splicing methods. The calculation formula of text a is as follows, while the calculation formula of text b is the same. After Fusion operation, the fused features Fai and Fbi are obtained:

$$F1x = linear([x, align\_x]) \quad (6)$$

$$F2x = linear([x, x - align\_x]) \quad (7)$$

$$F3x = linear([x, x * align\_x]) \quad (8)$$

$$Fa_i = [F1a_i, F2a_i, F3a_i] \quad (9)$$

$$Fb_i = [F1b_i, F2b_i, F3b_i] \quad (10)$$

## 2.4. Pooling

The feature information of the two texts obtained from the previous layer is pooled to the maximum to obtain more accurate text information a1, a2, a3, b1, b2, b3. The calculation formula is as follows:

$$a_i = \max\ pooling(Fa_i) \quad (11)$$

$$b_i = \max\ pooling(Fb_i) \quad (12)$$

## 2.5. Prediction

This layer is used to classify the final features of two pieces of text. First, the features are spliced, and then the final classification information is obtained through two fc connection layers, the calculation formula is as follows:

$$a = a1 + a2 + a3 \quad (13)$$

$$b = b1 + b2 + b3 \quad (14)$$

$$pred = linear(linear([a, b])) \quad (15)$$

# 3. Experiment

## 3.1. Dataset

In this paper, LCQMC[10], a large Chinese text matching dataset, is selected for experiments. It contains two pieces of text and a label. If two pieces of text are judged to match, the label is 1, and if not, the label is 0.

## 3.2. Experimental environment and parameter settings

This experiment is implemented with pytorch and trained on NVIDIA GeForce RTX 3090 24G. In the model, BERT selects the Chinese_wwm_ext [11] pretained by cui et al. The text length is 32, the word emebdding dimension is 768, the hidden layer dimension is 384, and the batch_size is 128, the lr of bert is 2e-5 and the other part is 0.001, the dropout is 0.2, and epoch is 20.

## 3.3. Evaluation Criteria

In this paper, the acc and F1 are selected to evaluate the performance of the model. The formula is as follows:

$$acc = \frac{(T_P + T_N)}{(T_P + T_N + F_P + F_N)} \quad (16)$$

$$F1 = \frac{2 * P * R}{(P + R)} \quad (p = \frac{T_P}{(T_P + F_P)},\ R = \frac{T_P}{(F_P + F_N)}) \quad (17)$$

TP represents the positive sample predicted by the model as a positive class, TN represents the negative sample predicted by the model as a negative class, FP represents the negative sample predicted by the model as a positive class, and FN represents the positive sample predicted by the model as a negative class.

## 3.4. Experimental result

The acc and f1 value of the model obtained after training, validation and test on LCQMC dataset are 84.14% and 85.62% respectively. Compared with the experimental results in recent literatures, the method in this paper is relatively good. The experimental comparison results are shown in the following, see Table 1.

**Table 1.** Experiment Result

| Model | ACC (%) | F1-score (%) |
|---|---|---|
| TeDCSA[12] | 79.40 | -- |
| CATsNET[13] | 83.15 | -- |
| Our Method | 84.14 | 85.62 |

Although the performance of the model in this paper cannot be compared with the large pretraining model, it takes about 20 minutes to train each epoch, with small parameters and high efficiency.

# 4. Conclusion

The model proposed in this paper is a lightweight text matching model. The model is evaluated on the Chinese text matching dataset. The results show that the operation of using convolution with different convolution cores to extract features and align them respectively is more sufficient to mine the semantic information of the text. Although the effect of this lightweight model is still some distance from large pretraining model, the training time of the model is short and the number of parameters is small, It is convenient for online use, so the next step is to train a text matching model with higher performance than the large pretraining model under the premise of ensuring lightweight.

# References


[1] Pang Liang, Lan Yanyan, Xu Jun, Guo Jiafeng, Wan Shengxian, Cheng Xueqi. Overview of Deep Text Matching [J]. Journal of Computer Science, 2017,40 (04): 985-1003.

[2] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data[C]//Proceedings of the 22nd ACM international conference on Information & Knowledge Management. 2013: 2333-2338.

[3] Shen Y, He X, Gao J, et al. A latent semantic model with convolutional-pooling structure for information retrieval[C]//Proceedings of the 23rd ACM international conference on conference on information and knowledge management. 2014: 101-110.

[4] Palangi H, Deng L, Shen Y, et al. Semantic modelling with long-short-term memory for information retrieval[J]. arXiv preprint arXiv:1412.6629, 2014.

[5] Pang L, Lan Y, Guo J, et al. Text matching as image recognition[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2016, 30(1).

[6] Chen Q, Zhu X, Ling Z, et al. Enhanced LSTM for natural language inference[J]. arXiv preprint arXiv:1609.06038, 2016.

[7] Wang Z, Hamza W, Florian R. Bilateral multi-perspective matching for natural language sentences[J]. arXiv preprint arXiv:1702.03814, 2017.

[8] Yang Runqi, Zhang Jianhai, Gao Xing, Ji Feng, Chen Haiqing. (2019). Simple and Effective Text Matching with Richer Alignment Features. 4699-4709. 10.18653/v1/P19-1465.

[9] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.

[10] Lcqmc:A large-scale chinese question matching corpus[C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 1952-1962.

[11] Cui Y, Che W, Liu T, et al. Pre-training with whole word masking for chinese bert[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29: 3504-3514.

[12] X. Zhang, F. Lei and S. Yu, "Self-attention Based Text Matching Model with Generative Pre-training," 2021, pp. 84-91,doi:10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00027.

[13] Zhen Wang, Xiangxie Zhang, Yicong Tan. Chinese Sentences Similarity via Cross-Attention Based Siamese Network.2021, https://doi.org/10.48550/arXiv.2104.08787