# An Enhanced ESIM Model for Sentence Pair Matching with Self-Attention

Yongkang Liu*, Xiaobo Liang*, Feiliang Ren*,†, Yan Li, Yining Hou, Yi Zhang, Lingfeng Pan

School of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China

†Corresponding Author: renfeiliang@cse.neu.edu.cn

**Abstract.** Sentence pair matching is one of the most basic tasks in natural language processing and is receiving extensive research attention. Most of existing models can be categorized into sentence interaction-based models and sentence encoding-based models. Here we propose a new enhanced ESIM model that could take full advantage of both these two kinds of methods. Specifically, our method can obtain the common information of a sentence pair as the sentence interaction-based models do, and can also extract the key information of a sentence pair itself as the sentence encoding-based models do. Besides, our method also uses Siamese networks to learn a unique structure to naturally rank the similarity between two sentences, which reduces both the model complexity and the size of parameters. With the proposed method, we participated in the task-3 of CCKS2018, which is called as *WeBank Intelligent Customer Service Question Match*. Finally, our method ranks the first among the hugely competitive models.

**Keywords:** Sentence pair matching, Neural Network, Self-Attention, Siamese nets.

## 1 Introduction

Sentence pair matching is a task that aims to determine whether two given sentences

* Equal contribution. Listing order is random.

have a specific relationship. The core of this task is to judge whether two given sentences have equal or similar intent. It usually could be viewed as a binary classification task with a label set is {1,0}, in which 1 means the two given sentences have equal or similar intent and 0 means the two given sentences don't have equal or similar intent. Table 1 shows a concrete example of the sentence pair matching task.

**Table 1. An example for sentence pair matching task**

| Sentence1 | Sentence2 | Label |
|---|---|---|
| 一般几天能通过审核(Generally, how long can I pass the review?) | 一般审核通过要多久（Generally, how long does it take for the review to pass.） | 1 |
| 一般会在什么时候来电话(When will the phone usually arrive?) | 一直在等待电话通知(I have been waiting for a phone call.) | 0 |

Sentence pair matching is one of the most basic tasks in natural language processing (NLP). A high performance sentence pair matching system would benefit lots of other NLP tasks such as Q&A, Chatterbot, information retrieval, machine translation, and so on.

Recently, due to the dataset released by Stanford, more and more research attention has been paid to the text entailment task, which is to predict a relation $y \in Y$ for a given pair of sentences based on their semantic, where Y = {entailment, contradiction, neutral}. It is very naturally to view the sentence pair matching task as a specific case of the textual entailment task because both of them determine the relationship between sentence pairs. Therefore, most of the existing textual entailment models can be applied to the sentence pair matching task after some necessary but simple modifications.

For text entailment task, many excellent models have been proposed and achieved good experimental results. For example, Gong et al [2017] design DIIN (Densely Iterative Inference Network) model [1], Wang et al [2017] propose BIMPM (bilateral multi-perspective matching) model [2], Ling et al propose ESIM model [3], Duan et al. [2018] present AF-DMN model, etc.

However, most of these existing methods map two sentences into different vector spaces. Intuitively, it would make more sense that two sentences should be mapped into the same vector space in order to extract sentences' semantic similarity information.

Moreover, we find that all of these existing models focus on the interaction

between sentences, but pay little attention to the semantic information of the sentences themselves.

To address these issues, we propose an enhanced ESIM Model that uses siamese nets [5] to map sentences into ONE vector space. By a parameter share mechanism, our new model reduces the size of parameters. Besides, we also introduce a self-attention mechanism in our model to tackle the long-term dependency issue in long sentences and further enhance the extraction of intra-sentence information.

Based on the proposed method, we participated in the task-3 of CCKS2018 (2018 China Conference on Knowledge Graph and Semantic Computing), which is called as *WeBank Intelligent Customer Service Question Match*. And our method ranks the first among the hugely competitive models.

## 2 Related work

Sentence pair matching is of great value for many NLP tasks and has received wide attention across academia and industry. According to whether there is an interaction between sentences, we can divide existing sentence pair matching methods into sentence interaction-based methods and sentence encoding-based methods.

Sentence interaction-based methods usually focus on the interaction information between sentence pairs. This kind of methods usually consist of three components: (1) an encoder layer that converts two sentences into their semantic representations, and LSTM is widely used in this layer; (2) an interaction layer that is responsible for linking and fusing information between the two sentences and generate new representations for the two sentences; (3) a prediction layer that predicts the relation of the input sentence pair. Lots of sentence pair matching methods belong to this category. For example, Gong et al [2017] design Densely Iterative Inference Network [1] that is able to achieve high-level understanding of a sentence pair by hierarchically extracting semantic features from an interaction space. This method pushes the multi-head attention to an extreme by building a word-word dimension-wise alignment tensor. Wang et al [2017] propose a bilateral multi-perspective matching (BiMPM) model that matches two encoded sentences using BiLSTM. In this method, a word of one sentence will be matched against all words of the other sentence from multiple perspectives [2]. Ling et al [2017] propose a sequential model named ESIM, which enhances the local inference information by computing the difference and the

element-wise product for a sentences pair [3].

Sentence encoding-based models pay more attention on sentences' own information than their interactions information. Usually, this kind of methods also consists of three components: (1) an encoder layer that converts two sentences into their semantic representations; (2) a matching layer that aligns the information between the two sentences at word level and produces new representations for the two sentences; (3) a prediction layer that predicts the relation of the input sentence pair. There are many sentence pair matching methods belong to this category. For example, Liu et al [2016] propose a sentence encoding-based model that encodes a sentence with a two-stage process [6]. Conneau et al [2016] show how universal sentence representations trained using the supervised Stanford Natural Language Inference datasets can consistently outperform unsupervised methods [7].

In our method, we merge above two kinds of models together and enhance the ESIM model by introducing a self-attention mechanism. Besides, we also use the siamese structure in our model.

# 3 Model

We denote two input sentences as $w_a = (w_1, w_2, ...w_{la})$ and $w_b = (w_1, w_2, ...w_{lb})$, where $w_a$ is the first sentence and $w_b$ is the second sentence. The goal of our model is to predict a label $y$ that indicates whether $w_a$ and $w_b$ have equal or similar intent. Here we will introduce data preprocessing first. Then, we will introduce our sentence pair matching model in detail.

### 3.1 Data Preprocessing

In our paper, data preprocessing consists of three parts: wrong words modification, synonym substitution and word segmentation.

Due to colloquial expressions in a dataset, there may be lots of wrong words which will affect the accuracy of the word segmentation. We use a custom vocabulary for wrong words modification.

For synonym substitution, we also use a custom synonym vocabulary. Besides, there are usually lots of unknown words in the test set. Replacing an unknown word with a synonym that appears in the corpus can reduce the number of unknown words and could further improve the final matching performance. We also do

traditional-to-simplified Chinese conversion operation and a numeric format conversion operation.

Word segmentation is the last step in data processing. We use jieba[1], a word segmentation tool, to performance word segmentation operation. We found that the effect of HMM pattern segmentation is not ideal due to the small granularity, thus we use a non-HMM pattern segmentation in practice. Besides, we also use a custom dictionary for word segmentation.

### 3.2 Our Model

Figure1 is the architecture of our model. There are four layers in our model: input encoding layer, local inference layer, aggregation layer, and prediction layer.

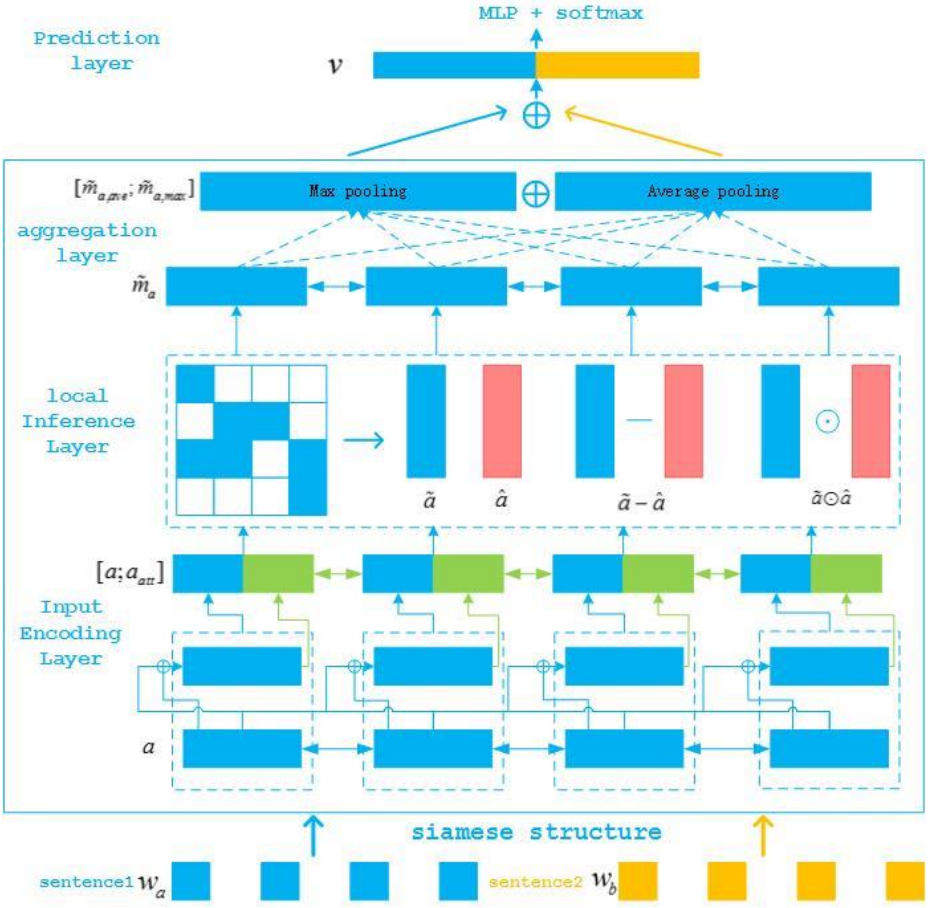


Figure 1. Architecture of Our Model

[1] https://github.com/fxsjy/jieba

**Input Encoding.** For each word in a sentence, we use bidirectional LSTM (BiLSTM[1] for short), which runs a forward and backward LSTM on a sequence starting from the left and the right respectively, to learn its representation. The hidden states generated by these two LSTMs at each time step are concatenated to represent a word at that time step. Here we denote $a_i$ and $b_i$ as the hidden (output) state generated by a BiLSTM at time $i$ over the input sequences $w_a$ and $w_b$.

$$a_i = BiLSTM(w_a, i), \forall i \in [1, ...l_a] \tag{1}$$

$$b_i = BiLSTM(w_b, i), \forall i \in [1, ...l_b] \tag{2}$$

To address the long-term dependency issue, we use a self-attention mechanism on this layer. Specifically, for sentence $a$, we first compute a self-attention matrix $S^t \in R^{m*m}$.

$$S_j^i = < a_i, a_j > \tag{3}$$

Where $S_j^i$ indicates the relevance between the *i-th* word and *j-th* word in sentence $\vec{a}$. Then the self-attentive vector for each word can be computed as following:

$$S_{att}^i = softMax(S^i) \tag{4}$$

$$a_{att,j}^i = \vec{a}_j \bullet S_{att,j}^i \tag{5}$$

We can derive the self-attentive vector for sentence $\vec{b}$ with the same method. Then the results of self-attention and BiLSTM are concatenated as the input of next layer.

$$\tilde{a} = [a; a_{att}] \tag{6}$$

**Local Inference Modeling.** Modeling local subsentential inference between two sentences is a basic component for determining the overall inference between them. It needs to employ some forms of hard or soft alignment to associate the relevant subcomponents between two sentences. Here we also use both LSTM and interactive attention mechanism for local inference modeling. The former technique helps to collect local inference for words and their contexts, and the latter enhances the

[2] We also tried GRUs (Gated Recurrent Units) as encoding method and experiments showed that they are inferior to BiLSTMs.

information between words and clauses. Here we use the attention mechanism as used in ESIM model [2], which leverages attention over the bidirectional sequential encoding of the input. Specially, we compute the attention weights as the similarity of a hidden state tuple $< \tilde{a}_i, \tilde{b}_i >$ between two sentences with Equation (7).

$$e_{ij} = \tilde{a}_i^T \tilde{b}_j \tag{7}$$

Intuitively, the content in $\tilde{b}_j$ that is relevant to $\tilde{a}_i$ will be selected and represented as a factor of the representation $\hat{a}_i$, Thus local inference of $\hat{a}_i$ is a weighted summation that is determined by the attention weight $e_{ij}$, which is computed as following.

$$\hat{a}_i = \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} \tilde{b}_j, \forall i \in [1, ...l_a] \tag{8}$$

$$\hat{b}_j = \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{kj})} \tilde{a}_i, \forall j \in [1, ...l_b] \tag{9}$$

Each word in another sentence is performed for with Equation (9).

The local inference information collected above is further enhanced in our models. Specially, we compute the difference and element-wise product for the tuple $< \tilde{a}, \hat{a} >$ and $< \tilde{b}, \hat{b} >$ as following.

$$m_a = [\tilde{a}; \hat{a}; \tilde{a} - \hat{a}; \tilde{a} \odot \hat{a}] \tag{10}$$

$$m_b = [\tilde{b}; \hat{b}; \tilde{b} - \hat{b}; \tilde{b} \odot \hat{b}] \tag{11}$$

We expect such operations could help sharpen local inference information between elements in the tuples and capture inference relationships such as contradiction.

**Aggregation Layer.** To determine the overall inference relationship between two sentences, we use an aggregation layer to combine the enhanced local inference information of two sentences. We sequentially perform the aggregation operation using BiLSTM with Equation (12-13).

$$\tilde{m}_a = BiLSTM(m_a, i), \forall i \in [1, ...l_a] \tag{12}$$

$$\tilde{m}_b = BiLSTM(m_b, i), \forall i \in [1, ...l_b] \tag{13}$$

Our aggregation layer further converts the resulting vectors obtained above to a

fixed-length vector by computing both average and max pooling with Equation (14-15). Then all these vectors are concatenated to form the final fixed length vector (Equation 16) and feed it to the final classifier to determine the overall inference relationship between two sentences.

$$\tilde{m}_{a,ave} = \sum_{i=1}^{l_a} \frac{\tilde{m}_{a,i}}{l_a} \quad , \quad \tilde{m}_{a,max} = \max_{i=1}^{l_a} \tilde{m}_{a,i} \tag{14}$$

$$\tilde{m}_{b,ave} = \sum_{j=1}^{l_b} \frac{\tilde{m}_{b,j}}{l_b} \quad , \quad \tilde{m}_{b,max} = \max_{j=1}^{l_b} \tilde{m}_{b,i} \tag{15}$$

$$v = [\tilde{m}_{a,ave}; \tilde{m}_{a,max}; \tilde{m}_{b,ave}; \tilde{m}_{b,max}] \tag{16}$$

**Prediction Layer.** The aim of this layer is to evaluate the label probability distribution of two sentences. To this end, we employ a multi-layer perception (MLP) classifier. After obtaining the representation $V$ of the two sentences, the distribution $p(.)$ can be formalized as:

$$P(y \mid w_a, w_b) = softMax(\tanh(W_1 V + b_1)) \tag{17}$$

where $W_1$ *and* $b_1$ are trainable parameters.

# 4 Experiments

### 4.1 Dataset

The task3 of CCKS2018, which is called as *intelligent customer service question matching*, aims to match the intent of two sentences. Specially, it is required to determine whether the intentions of two sentences are equal or similar. The original corpus contains 100,000 sentence pairs as training set, 10,000 sentence pairs as online dev set to verify submission, and 110,000 sentence pairs as online test set for final submission. This task uses precision, recall, F1, and accuracy as evaluation metrics.

### 4.2 Experimental Setup

During offline training, we divide the original train set into 10 parts and select one of them as dev set and the rest as train set.

In experiment, the statistical results show that 98% of the sentences are less than

20. So we set max sentence length is 20. We initialize word embedding using the Chinese Word Vectors[3]. Word embedding dimension is set to 300. The recurrent unit size is set to 150, and the batch size is set to 64. Adam is used for optimization, and the learning rate is set to 0.005.

### 4.3 Results and Discussion

In our experiments, we first use ESIM as baseline, and we adjust the model to Siamese structure. Then we add self-attention mechanism in the encoding layer. The experimental results are shown in Table 2. It shows that our method obtains better results than the baselines.

**Table 2. single model offline result**

| Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| ESIM | 0.90322 | 0.91502 | 0.91056 | 0.90908 |
| + Siamese | 0.90436 | 0.94123 | 0.92210 | 0.92367 |
| +Self-Att(Our model) | 0.92349 | 0.95860 | 0.93930 | 0.94072 |

In the second part of our experiments, we conduct experiments to evaluate the ensemble method. We train our model several times, and integrate the generated models together as the ensemble model. Table3 is the experimental results, which show that a simple ensemble method is effective, and the performance could be further improved significantly.

**Table 3. our model ensemble result (online dev set)**

| Model(our model) | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| Single | 0.85459 | 0.83220 | 0.84530 | 0.84324 |
| Ensemble | 0.86505 | 0.85640 | 0.86140 | 0.86070 |

In the third part of our experiments, we try to adopt the label voting ensemble mechanism or probability voting ensemble mechanisms. Table4 is the experimental results, which show that the performance is further improved.

Finally, the online result of our model is shown in Table5.

[3] https://github.com/Embedding/Chinese-Word-Vectors

**Table 4. multi-model ensemble result (online dev set)**

| Method | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| Probability voting | 0.85815 | 0.87000 | 0.86310 | 0.86404 |
| Label voting | 0.85852 | 0.87020 | 0.86340 | 0.86432 |
| Mixed voting | 0.85821 | 0.87160 | 0.86380 | 0.86485 |

**Table 5. final submit score (online test set)**

| Model | Precision | Recall | Accuracy | F1 |
|---|---|---|---|---|
| Multi-model | 0.84785 | 0.85480 | 0.85070 | 0.85131 |

# 5.Conclusion

In this paper, we propose an enhanced ESIM model for sentence pair matching task. Our method merges the ability of obtain the common information of two sentence and the ability of extracting the key information of the sentences themselves. We also use Siamese nets in our model, which reduces both model complexity and parameter size. Experimental results show that our method is effective and it ranks the first among the hugely competitive models in the task3 of CCKS2018.

In the future, we will further mine the features of data and try to introduce syntactic structure information in the model. Besides, we will also try to design a new neural network structure to improve the representation and reasoning ability for the sentence matching task.

# 6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC No. 61572120, 61672138 and 61432013).

# Reference

1. Gong Y, Luo H, Zhang J. Natural language inference over interaction space[J]. arXiv preprint arXiv:1709.04348, 2017.
2. Wang Z, Hamza W, Florian R. Bilateral multi-perspective matching for natural language sentences[J]. arXiv preprint arXiv:1702.03814, 2017.
3. Chen Q, Zhu X, Ling Z H, et al. Enhanced LSTM for Natural Language Inference[C]// Meeting of the Association for Computational Linguistics. 2017:1657-1668.
4. Duan C, Cui L, Chen X, et al. Attention-Fused Deep Matching Network for Natural Language Inference[C]//IJCAI. 2018: 4033-4040.
5. Mueller J, Thyagarajan A. Siamese Recurrent Architectures for Learning Sentence Similarity[C]//AAAI. 2016, 16: 2786-2792.
6. Liu Y, Sun C, Lin L, et al. Learning natural language inference using bidirectional LSTM model and inner-attention[J]. arXiv preprint arXiv:1605.09090, 2016.
7. Conneau A, Kiela D, Schwenk H, et al. Supervised learning of universal sentence representations from natural language inference data[J]. arXiv preprint arXiv:1705.02364, 2017.
8. Munkhdalai T, Yu H. Neural semantic encoders[C]//Proceedings of the conference. Association for Computational Linguistics. Meeting. NIH Public Access, 2017, 1: 397.
9. Chen Q, Zhu X, Ling Z H, et al. Recurrent Neural Network-Based Sentence Encoder with Gated Attention for Natural Language Inference[J]. 2017:36-40.
10. Munkhdalai T, Yu H. Neural Semantic Encoders[C]// 2017:397-407.
11. Almiman A, Ramsay A. A Hybrid System to apply Natural Language Inference over Dependency Trees[C]// RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning. 2017:64-70.
12. Huang D. Research on attention memory networks as a model for learning natural language inference[C]//Proceedings of the Workshop on Structured Prediction for NLP. 2016: 18-24.
13. Parikh A P, Täckström O, Das D, et al. A decomposable attention model for natural language inference[J]. arXiv preprint arXiv:1606.01933, 2016.
14. Almiman A, Ramsay A. A Hybrid System to apply Natural Language Inference over Dependency Trees[C]//Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. 2017: 64-70.
15. Yang H, Costa-jussà M R, Fonollosa J A R. Character-level Intra Attention Network for Natural Language Inference[J]. arXiv preprint arXiv:1707.07469, 2017.