Semantic Similarity Calculation based on Siamese TRAT

Xing-Cen Lu*, Inwhee Joe, Ph.D*
*Dept. of Computer Science, Hanyang University
luxingcenss@162.com, iwjoe@hanyang.ac.kr

트랜스포머 인코더와 시암넷 결합한 시맨틱 유사도 알고리즘

육성잠*, 조인휘*
*한양대학교 컴퓨터소프트웨어학과

Abstract

To solve the problem that existing computing methods cannot adequately represent the semantic features of sentences, Siamese TRAT, a semantic feature extraction model based on Transformer encoder is proposed. The transformer model is used to fully extract the semantic information within sentences and carry out deep semantic coding for sentences. In addition, the interactive attention mechanism is introduced to extract the similar features of the association between two sentences, which makes the model better at capturing the important semantic information inside the sentence. As a result, it improves the semantic understanding and generalization ability of the model. The experimental results show that the proposed model can improve the accuracy significantly for the semantic similarity calculation task of English and Chinese, and is more effective than the existing methods.

1. Introduction

Semantic similarity calculation is a very important research direction in NLP field. With the development of neural networks, semantic similarity computing tasks have made significant breakthroughs.

The Siamese Recurrent Architectures [1] proposed by MIT compare the similarities between two sentences of different length, encoding them into vectors of the same length. Siamese encodes the sentence embedding initialized with Word2vec by using two LSTMs with shared weights. The input sentence is finally output by encode as the hidden layer of the last time step of LSTM. After the encode vector of each sentence is obtained, it has good stability for any simple similarity equation. The measurement method uses dis=e^(-d) based on Manhattan distance d. Since the score is 1-5, dis*4.0+1.0 is processed. Simple metrics allow sentence representation to better express complex semantic relationships.

However, using LSTM model has the problem of slow speed, and can not fully extract the similar information between sentences. Because of its pooling mechanism, CNN model can filter out a lot of low-level information and sentence, so it cannot adequately encode semantics.

Therefore, instead of LSTM, we use Transformer-Encoder [2] to enable the model to capture contextual information in the semantics. The parallel structure of the Transformer can speed up the calculation model, then using interactive attention mechanism [3], makes the model can be automatically concentration in the two sentences similar semantic information, and does not require any additional knowledge, enhance model robustness, better semantic similarity calculation.

2. Related Work

The method based on neural network model is to use word2vec and other word vector methods to convert words into word vectors, and then input into the neural network language model to obtain semantic feature representation of sentences, and then send into the full connection layer or use distance formula to calculate semantic similarity.

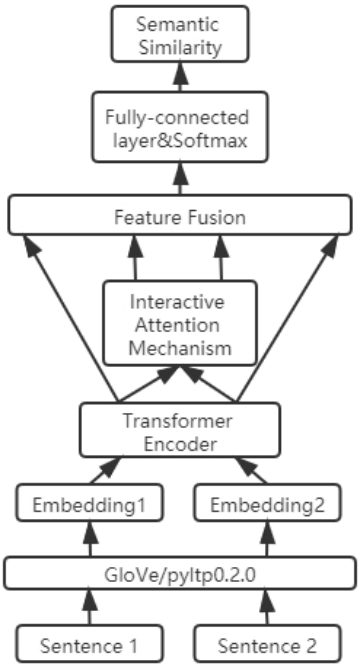
Siamese LSTM is the first one who proposed using Siamese network for semantic similarity analysis. They chose LSTM as the main network architecture, combined with Manhattan distance to calculate the semantic similarity between sentence pairs, and the performance is far better than other models.

Difference. (1) We chose Transformer, which is good at capturing one kind of contextual information, as the primary architecture instead of LSTM. (2) The quality of the word vector will have a direct impact on the experimental results, so we choose the Glove [4] model, which performs better than Word2vec, and Tencent's 8 million word vector [5] which has the characteristics of large corpus and wide coverage to train the word vector. (3) We abandoned the measurement method of distance calculation, and used the fully connected layer to adjust the features, and used the Softmax function for classification and prediction. (4) The interactive attention mechanism is used to improve the performance of the model to capture the important semantic information inside the sentence.

3. Model

Siamese TRAT mainly includes the following five parts: word vector embedding representation, Transformer coding layer, interactive attention layer, feature fusion layer, and output layer. The structure is shown in Figure 1.

3.1 Word embedding. We use Tencent's 8 million word vector as the Chinese word vector, and use Harbin Institute of Technology word segmentation tool kit pyltp0.2.0 for Chinese word segmentation. We used GloVe vectors pretrained by Stanford University. Unsigned words are randomly initialized by a Gaussian distribution. All parameters including word vectors are updated with the training process. The maximum length of the sequence is selected according to the length coverage of the experimental corpus above 95%.



(Figure 1) Structure of Siamese TRAT

3.2 Transformer-Encoder. We use a 6-layer stack of Transformer Block as the encoder.The structure of each Transformer block is shown in Figure 2, which consists of four parts: multi-head attention mechanism, residual connection, layer normalization, and fully-connected network. The input is the sum of word vector encoding and position encoding of the sentence vocabulary. The purpose of position encoding is to distinguish the position relationships of words in the sentence.The calculation formula is as follows:

$$PE(pos,2i) = \sin(pos/10000^{2i/d_{\mathrm{model}}}) \quad (1)$$

$$PE(pos,2i+1) = \cos(pos/10000^{2i/d_{\mathrm{model}}}) \quad (2)$$

pos: shows the position of a word in a sentence
i: represents the position of the word vector
d: represents the dimension of the word vector

The calculation formula of multi-head attention is as follows:

$$Attention(Q,K,V) = \mathrm{softmax}(\frac{QK^{T}}{\sqrt{d_k}})V \quad (3)$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (4)$$

$$MultiHead(Q,K,V) = Con(head_1 ... head_h)W^0 \quad (5)$$

$W_i^Q, W_i^K, W_i^V \in R^{d_{\mathrm{model}} \times d_k}, W^0 \in R^{d_{\mathrm{model}} \times d_{\mathrm{model}}}, i = 1,..., h$,
Q、K and V are equal to the input vector matrix obtained above.
W_i^Q、W_i^K、W_i^V: respectively represent the matrix of linear transformation on Q, K, and V.
h: represents the number of attention, each attention captures the information of a subspace in the text, and splicing h attention heads to get multiple attention values through matrix W^0
The calculation formula of layer normalization is as follows:

$$m = \frac{1}{H}\sum_{1}^{H} x_i \quad (6)$$

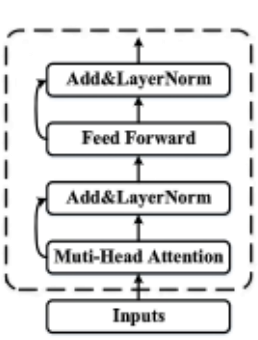
$$\sigma = \sqrt{\frac{1}{H}\sum_{i=1}^{H}(x_i - m)^2} \quad (7)$$

$$LN(x) = \alpha \times \frac{(x-m)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \quad (8)$$

x: represents the i-th dimension of the input matrix x
m and σ: respectively represent the mean and variance of the input x
α and β are introduced parameters to be learned to make up for information lost in the process of normalization
ε is an infinitesimal number that prevents the divisor from being zero
The function of layer normalization is to accelerate the convergence speed of the model and improve the training efficiency.
The calculation formula of the fully-connected layer is as follows:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (9)$$

W_1, W_2: the weight matrices of the fully-connected layer
b_1, b_2: the offset of the fully-connected layer



(Figure 2) Structure of Transformer Block

3.3 Interactive Attention Mechanism. The design idea is as follows: firstly, the similarity matrix is calculated to get the similarity between the words in two sentences, and then the attention mechanism is used to recode the words in each sentence respectively. The calculation formula is as follows:

$$E = T_{1i}^{T} T_{2j} \tag{10}$$

$$e_{ij} = T_{1i}^{T} T_{2j} \tag{11}$$

$$s_{1i} = \sum_{j=1}^{l_{s2}} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_{s2}} \exp(e_{ik})} T_{2j}, \forall i \in [1,2,..., l_{s1}] \tag{12}$$

$$s_{2j} = \sum_{j=1}^{l_{s1}} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_{s1}} \exp(e_{kj})} T_{1i}, \forall j \in [1,2,..., l_{s2}] \tag{13}$$

E: the similarity matrix of two sentences
T1: the transpose of the global semantic features of sentence 1 encoded in Transformer
T2: the semantic feature of sentence 2
e_{ii} : the element in row i and column j of matrix E, representing the similarity of the ith word in sentence 1 to the j-th word in sentence 2
s_1 : uses the attention mechanism to extract similar information in sentence 2 and sentence 1
s_2 : uses the attention mechanism to extract similar information in sentence 1 and sentence 2
s_1 and s_2 are important features for the model to predict

3.4 Fure Fusion. The Transformer layer encodes the global semantic features t_1 and t_2 of the input sentence. t_1 represents the semantic features of sentence 1 and t_2 represents the semantic features of sentence 2.
The interactive attention layer extracts the local similar features s_1 and s_2 in each sentence.
s_1 represents similar features of sentence 1 and s_2 represents similar features of sentence 2.
The feature fusion layer fuses the features of the two parts according to Formula 14:

$$m = [t_1; t_2; t_1 - t_2; s_1; s_2; s_1 - s_2] \tag{14}$$

$t_1 - t_2, s_1 - s_2$ are the subtraction operation of vectors, with the purpose of obtaining differential features. Finally, the final feature fusion vector m is obtained by means of vector splicing.

3.5 Outputs. The output layer adopts the fully-connected network to adjust the weight of the features and the softmax function to predict the classification result $\hat{y}$, and the input is the fusion feature fusion vector m. The calculation process is shown in Formula 15 and 16. The experiment proves that using the fully-connected layer as the output effect is better than using Euclidean distance or cosine similarity as the output:

$$\hat{p}(y|S_1, S_2) = soft\ \max(W^m m + b^m) \tag{15}$$

$$\hat{y} = \arg\max(\hat{p}(y|S_1, S_2)) \tag{16}$$

The loss function of the model adopts the cross entropy loss function as follows:

$$J(\theta) = -\frac{1}{k} \sum_{i=1}^{k} r_i \ln y_i + \lambda \left\| \theta \right\|_F^2 \tag{17}$$

$r_i \in \Re^m$ is the true value of the one-hot coded tag
$y_i \in \Re^m$ is calculated by the softmax function to predict the probability of each category.
k is the number of categories.
a is the hyperparameter of L2 regularization.
We use both L2 regularization and dropout to prevent overfitting of the model.
In order to prevent jitter in the training process, Adam [6] algorithm is used for optimization and EarlyStopping is used to prevent model fitting.

4. Datasets and Evaluation

In order to verify the effectiveness of the model, we added the traditional TF-IDF method and Siamese CNN, Siamese LSTM and Siamese LSTM-ATT for comparative analysis. The Chinese dataset uses intelligent customer service data (2018 China Knowledge Graph and Semantic Computing Conference Intelligent Customer Service Question Matching Contest). The dataset is a classification task to judge whether two sentences are semantically similar or dissimilar. Accuracy and F1 score were selected as evaluation indexes. SICK [7] dataset was selected for the experiment.
Pearson's correlation coefficient was selected as the evaluation index.
The specific calculation formula is as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{18}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{19}$$

$$\rho_{X,Y} = \frac{\mathrm{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{20}$$

TP: represents the number of positive examples predicted to be positive examples.
FP: represents the number of negative cases predicted to be positive cases.
Fn: represents the number of positive cases predicted to be negative.
Tn: represents the number of negative cases predicted to be negative.
X: the actual value.
Y: the predicted value.

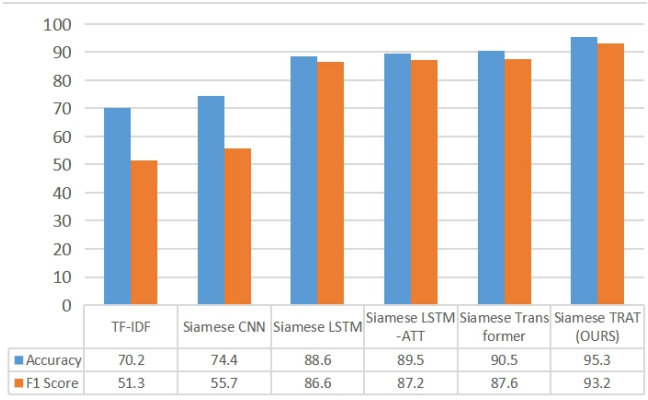
Table1 and Table2 show that the performance of Transformer based model is much better than traditional CNN and LSTM models, which proves that Transformer network has stronger semantic coding ability.
It can be seen from Table 1 that in the Chinese data set, the accuracy of our model is improved by 6.7% compared with Siamese LSTM. F1 score improved by 6.6% over the Siamese LSTM.
As can be seen from Table 2, in the English data set, the

Pearson coefficient of our model is improved by 14.1% compared with Siamese LSTM.
By comparing the data of Siamese Transformer and Siamese Trat in the two tables, we can also prove that the interaction information between sentences plays an important role in similarity calculation.

<Table 1> results(Tencent's)

Model	Accuracy	F1 Score
TF-IDF	70.2	51.3
Siamese CNN	74.4	55.7
Siamese LSTM	88.6	86.6
Siamese LSTM-ATT	89.5	87.2
Siamese Transformer	90.5	87.6
Siamese TRAT(OURS)	**95.3**	**93.2**

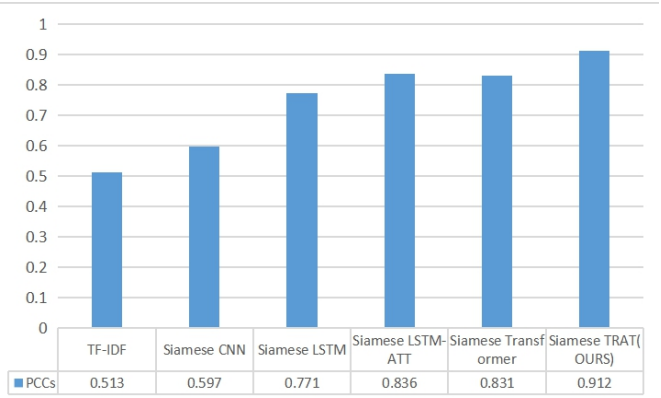
(Figure 5) models test on Tencent's dataset



<Table 2> results(SICK)

Model	Pearson's correlation coefficient
TF-IDF	0.513
Siamese CNN	0.597
Siamese LSTM	0.771
Siamese LSTM-ATT	0.836
Siamese Transformer	0.831
Siamese TRAT(OURS)	**0.912**

(Figure 4) models test on SICK



5. Conclusion

In this paper, a semantic similarity computing model based on Transformer encoder is proposed and interactive attention mechanism is introduced to extract between sentences.
The results of experiments on different data sets show that the proposed model is significantly better than the baseline model on Chinese and English data, and the effectiveness of the interactive attention mechanism is also proved.

6. Reference

[1] Mueller J, Thyagarajan A "Siamese recurrent archi tectures for learning sentence similarity" Thirtieth AA AI Conference on Artificial Intelligence 2016
[2] Vaswani A, Shazeer N, Parmar N, et al. "Attentio n is all you need" Advances in neural information pr ocessing systems, 2017:5998-6008.
[3] Parikh A P, Täckström O, Das D, et al. "A deco mposable attention model for natural language inferen ce" arXiv: 1606.01933, 2016.
[4] Pennington J, Socher R, Manning C. "Glove: Glo bal vectors for word representation" Proceedings of th e 2014 conference on empirical methods in natural la nguage processing (EMNLP), 2014:1532-1543.
[5] Song Y, Shi S, Li J, et al. "Directional skip-gram: explicitly distinguishing left and right context for wo rd embeddings" Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technol ogies, Volume 2 (Short Papers), 2018:175-180.
[6] Kingma D P, Ba J. "Adam: A method for stochas tic optimization" arXiv, 2014:1412.6980.
[7] Marelli M, Menini S, Baroni M, et al. "A SICK cure for the evaluation of compositional distributional semantic models" LREC. 2014:216-223.