ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

Research on Text Similarity Algorithms Based on Interactive Attention

Junhong Chen[1,2,a,*], Kaihui Peng[3,b]

[1]School of Software Engineering, South China University of Technology, Guangzhou, China
[2]LeiHuo studio, NetEase, Hangzhou, China
[3]Faculty of Business and Economics, University of Malaya, Kuala Lumpur, Malaysia
[a]jupyterchen@163.com,[b]pengkaihui66@163.com
**Corresponding author*

Abstract: *This paper proposes an enhanced text matching model with augmented recurrent attention that utilizes interactive attention mechanisms. During vector encoding, the proposed model employs attention to interact between two input texts. Following the interaction, it leverages Bi-LSTM to re-encode the sequence at a more advanced level, enabling the model to comprehensively learn global information. Additionally, an attention mechanism is incorporated to emphasize the importance of high-weights words. Furthermore, a fusion layer is added to better integrate the two text segments into a single result, which facilitates subsequent text similarity computations. The model demonstrates a high accuracy in text similarity calculations.*

Keywords: *Text similarity calculation; Attention mechanism; Pre-trained models*

1. Introduction

The Siamese network architecture consists of two coupled artificial neural networks[1]. Siamese networks operate by obtaining inputs from two sample sets and each of its two sub-networks receives an input and generates a high-dimensional spatial representation. Finally, a certain metric, such as cosine similarity calculation[2], is used to compare the similarity of the two samples. Siamese networks typically have a three-layer structure, starting with the input layer, which is responsible for converting words into word vectors and feeding these vectors[3] into subsequent encoding layers. Next is the encoding layer, which encodes the input word vectors to produce sentence vectors that represent the entire sentence. Lastly, the similarity measurement layer is tasked with determining the similarity relationship between the two sentence vectors, using methods such as simple vector distance calculations, like cosine similarity, or by combining the vectors and employing classifiers or multilayer perceptrons to represent the direct similarity relationship between the vectors. Text similarity calculation models based on Siamese networks have achieved good accuracy, but the two inputs in the Siamese network are relatively independent during the encoding process, lacking interaction between sentences. The advantage of this independent encoding is that it allows sentences to be relatively independent during the input process, not relying on information from other sentences. However, the drawback is that it may lead to a loss of precision in subsequent similarity computation. To address the issue of independent encoding in the Siamese network model, this paper uses an interactive attention mechanism[4] to enhance the interaction between the two inputs. This model is capable of extracting richer interaction information between the two inputs, which is beneficial for improving the accuracy in text similarity calculations.

2. Design of Text Similarity Algorithms Based on Interactive Attention Mechanisms

2.1 Algorithm Overview

This paper proposes the Enhanced Recurrent Attention Matching Model (ERAMM), which employs interactive attention mechanisms. The input layer uses BERT-wwm[5], a model more accommodating to Chinese text, to convert words into vectors and feed these transformed vectors into subsequent processing steps. In the encoding layer, attention mechanisms are employed to facilitate interaction between the two inputs, resulting in interactional vectors. Following interaction, Bi-LSTM[6] is utilized to re-encode the sequence at a higher level, enabling the model to comprehensively learn global information. Additionally,

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

an attention mechanism is added to amplify the influence of words with higher importance. After respectively obtaining the interactive representations of the texts, a fusion layer is used to combine the representations of both texts into a single vector representation, which aids in text similarity computation. The similarity measurement layer is responsible for determining the similarity relationship between the two sentence vectors, where the text similarity is measured probabilistically. The structure of the ERAMM model is illustrated in Figure 1.

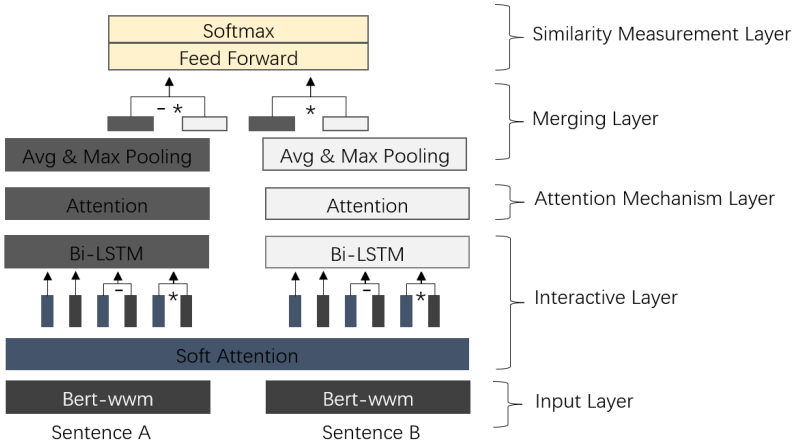

Figure 1: Structure of the ERAMM Model

2.2 Input Layer

Firstly, for input sentences, semantic capture using Bert-wwm is required. The ERAMM also employs the standard Bert-wwm input, which means the sentences are segmented by characters, and [CLS] tokens are added at the beginning, while [SEP] tokens are added at the end of the sentences, denoted as S_a and S_b, respectively. These segmented sentences are then input into the Bert-wwm for semantic capture. After encoding through Bert-wwm, the text has extracted rich contextual semantic information. However, during this process, the information extraction of the two text segments is independent, and there is still a lack of interaction between the two segments. Therefore, it is necessary to input both text segments into the interactive layer for semantic interaction.

2.3 Interactive Layer

The interactive layer overcomes the issue of independent encoding in Siamese network models by interacting with the sentences to enhance the accuracy of subsequent text similarity measurement. After the interaction between the two text segments is complete, different combination methods are used to enhance local reasoning information. Finally, Bi-LSTM is employed to re-encode the sequence at a higher level, enabling the model to comprehensively learn global information.

To interact using attention mechanisms, it is first necessary to calculate the similarity between the two sentences to address comparisons more specifically[7]. Let h_{a_i} represent the semantic representation of the i-th character in the a sequence, and h_{b_i} represent the semantic representation of the i-th character in the b sequence, then the similarity calculation is as follows.

$$e_{ij} = h_{a_i}^T h_{b_j} \tag{1}$$

The subsequent calculation of the interactional information between sentences requires combining sentences a and b to generate asentence that is weighted by their mutual similarity. Let the length of sentence a be denoted as l_a and the length of sentence b as l_b. The calculations are represented by formulas (2) and (3).

$$\tilde{h}_{a_i} = \sum_{j=1}^{l_b} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_b} \exp(e_{ik})} h_{b_j}, \forall i \in [1,\cdots,l_a] \tag{2}$$

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

$$\tilde{h}_{b_j} = \sum_{i=1}^{l_a} \frac{\exp(e_{ij})}{\sum_{k=1}^{l_a} \exp(e_{kj})} h_{a_i}, \forall j \in [1,\cdots,l_b] \tag{3}$$

Among them, $\tilde{h}_{a_i}$ and $\tilde{h}_{b_j}$ are the representation vectors obtained by interacting and mutually weighting the original vectors. Next, different transformation and combination methods are used to enhance local reasoning information, with the computational process represented by formulas (4) and (5).

$$m_a = \left[h_{a_i}; \tilde{h}_{a_i}; h_{a_i} - \tilde{h}_{a_i}; h_{a_i} \circ \tilde{h}_{a_i}\right] \tag{4}$$

$$m_b = \left[h_{b_i}; \tilde{h}_{b_i}; h_{b_i} - \tilde{h}_{b_i}; h_{b_i} \circ \tilde{h}_{b_i}\right] \tag{5}$$

After transforming and combining the vector results, finally, the Bi-LSTM is utilized to re-encode the sequence at a higher level. This allows the model to comprehensively learn global information, going beyond the limitation of simply calculating the similarity features of words, and enables it to learn comprehensive information at the global level as well.

$$\bar{m}_{a_i} = BILSTM(m_a, i), \forall i \in [1,\cdots,l_a] \tag{6}$$

$$\bar{m}_{b_j} = BILSTM(m_b, j), \forall j \in [1,\cdots,l_b] \tag{7}$$

Among them, $\bar{m}_a$ and $\bar{m}_b$ are the representation vectors processed by the Bi-LSTM. These two vectors can be used as the new representation vectors for sentences a and b, respectively.

2.4 Attention Mechanism Layer

The use of the attention mechanism in natural language processing tasks allows the model to focus on more important information while reducing the weight of less important information[8]. In an article, some less significant information may be considered noise in the text similarity calculation task. Therefore, after generating new vectors, an attention mechanism is employed. In the calculation of the representation attention mechanism, a fully connected layer is initially used to indicate the importance of each word vector. This importance is then used to redistributing the overall weight of the word vectors. By using self-attention mechanisms, key information can be made to have a greater impact, thereby reducing the influence of non-key information. The calculation of the representation attention mechanism involves first computing the weight values of time steps through a fully connected layer, as shown in the following calculation.

$$Q_{att,a} = f(W_a \cdot \bar{m}_a + bias) \tag{8}$$

$$Q_{att,b} = f(W_b \cdot \bar{m}_b + bias) \tag{9}$$

Among them, f represents the activation function, and in this paper, a sigmoid activation function is adopted. W denotes the parameters of the fully connected layer, which can be learned during the model training process. After obtaining the weight values, these weights are used to perform a weighted operation on the representation vectors to obtain the new vectors, as calculated below.

$$Q_a = Q_{att,a} \cdot \bar{m}_a \tag{10}$$

$$Q_b = Q_{att,b} \cdot \bar{m}_b \tag{11}$$

2.5 Merging Layer

The interactional attention mechanism can be used to interact between two texts, obtaining mutually represented vectors. The Bi-LSTM can comprehensively extract contextual information, and the representation attention mechanism can highlight high-value information. However, when measuring similarity, it is necessary to merge the two vectors into one, which is convenient for measurement. Therefore, this paper adds a merging layer, the purpose of which is to combine the two vectors and highlight the differences and similarities between them, which is helpful for measuring similarity in the subsequent process.

In the merging layer, this paper first extracts the more refined features of the two vectors through max pooling and avg pooling, and concatenates them into a sentence vector that represents the entire sentence. Let V_a and V_b be the sentence vectors produced after pooling. The calculation process is as follows.

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

$$V_{a,avg} = \sum_{i=1}^{l_a} \frac{Q_{a_i}}{l_a} \tag{12}$$

$$V_{a,max} = max_{i=1}^{l_a} Q_{a_i} \tag{13}$$

$$V_a = [V_{a,avg}; V_{a,max}] \tag{14}$$

$$V_{b,avg} = \sum_{j=1}^{l_b} \frac{Q_{b_j}}{l_b} \tag{15}$$

$$V_{b,max} = max_{j=1}^{l_b} Q_{b_j} \tag{16}$$

$$V_b = [V_{b,avg}; V_{b,max}] \tag{17}$$

After separately generating sentence vectors, it is necessary to merge these two vectors. The primary goal is to better integrate the interactive sentence representations of the two texts to facilitate the calculation of the similarity between the sentences in the subsequent process. The calculation process is as follows.

$$Z_{mul} = V_a \circ V_b \tag{18}$$

$$Z_{sub\&mul} = (V_a - V_b) \circ (V_a - V_b) \tag{19}$$

$$Z = [Z_{sub\&mul}; Z_{mul}] \tag{20}$$

Among them, Z_{mul} represents the result of multiplying the two vectors; using the product of the vectors as input enables the model to better compare the similarity between the two vectors. $Z_{sub\&mul}$ is a vector obtained by subtracting the two vectors first and then performing a multiplication operation; using it as input allows for a better comparison of the differences between the two models.Zis the final sentence relationship vector formed by the interactive fusion of the two vectors, which is then input into the similarity measurement layer for similarity judgment.

2.6 Similarity Measurement Layer

After obtaining the fused vector of the two texts, the similarity measurement layer will predict the similarity relationship between the texts based on this vector, and the calculation process is as follows.

$$P = softmax(W \cdot Z + bias) \tag{21}$$

In this case, W and biasare parameters of the model that are learned during the training process. P refers to the predictive result, which is a probability distribution of the labels. The loss function used by the model is cross-entropy, and the loss function is represented as follows.

$$\text{loss} = -\sum_{i=1}^{N} [y_i \cdot log(p_{1,i}) + (1 - y_i) \cdot log(p_{0,i})] \tag{22}$$

Among them, y_i denotes the relationship of the i-th pair of samples, which can be either 0 or 1. A value of 0 indicates that the two are dissimilar, i.e., a negative sample, whereas 1 represents that they are similar, i.e., a positive sample. $p_{1,i}$ indicates the probability that the i-th pair of samples are similar. N represents the total number of samples in this batch.

3. Experiment Results and Analysis

3.1 Experimental Dataset

The dataset used in this experiment includes the LCQMC[9] Chinese semantic matching dataset and the BQ Corpus[10] credit text similarity matching dataset.

3.2 Baseline Models

In this experiment, several classic similarity calculation models are compared.

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

(1) ESIM[11]: The model consists of three parts. The first part is the input layer, which encodes the input vectors and introduces the encoded word vectors into the LSTM network to obtain new sentence vectors. The second part is the interaction layer, which uses attention mechanisms to extract interactive information representations between texts. The third part is the prediction layer, utilizing the LSTM network to extract contextual information fully and predict the results. The advantage of the ESIM model lies in its use of text interaction and the addition of feature enhancement.

(2) BIMPM[12]: The model uses bidirectional LSTM to process the input but is designed with four different matching methods, which are then concatenated to obtain the representation vector of the sentence. The model can make full use of information from more angles, but this also increases the model' s parameters, leading to slower training.

(3) ABCNN[13]: The model uses CNN to process sentences and also employs interactive attention mechanisms. It proposes three ways to apply attention mechanisms, with the best-performing method being the application of attention to both the input vectors and the vectors after convolutional pooling.

(4) Siamese-LSTM[14]: It is a Siamese network model based on Bi-LSTM. It uses Bi-LSTM to extract contextual information and requires the parameters of Bi-LSTM to be shared between two subnetworks.

3.3 Evaluation Metrics

The semantic similarity calculation models used in this chapter predict two probability values, which can be considered as a form of binary classification model. A common evaluation criterion for binary classification is accuracy, but this evaluation method can be misleading when there is an imbalance between positive and negative samples. Therefore, to evaluate the results more accurately and scientifically, this chapter employs the evaluation method based on the confusion matrix. The confusion matrix is shown in Table 1.

Table 1: Confusion Matrix

	Similar (Original Label)	Non-Similar (Original Label)
Similarity (Predicted Result)	TP (True Positive)	FP (False Positive)
Non-Similarity (Predicted Result)	FN (False Negative)	TN (True Negative)

The TP, or True Positive, refers to the case where the original sample is a positive sample, and the model's prediction is also positive, indicating that the model has made its prediction correctly. The TN, or True Negative, represents the case where the original sample is a negative sample, and the model predicts it as negative, which means the model has made its prediction correctly. The FP, or False Positive, indicates that the original sample was a negative sample, but the model has mispredicted it as positive, showing that the model made an incorrect prediction. The FN, or False Negative, signifies that the original sample was a positive sample, but the model mispredicted it as negative, again indicating an incorrect prediction. The calculation formulas for the relevant indicators are as follows.。

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{23}$$

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

$$Recall = \frac{TP}{TP + FN} \tag{25}$$

$$F1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \tag{26}$$

Among them, Accuracyrepresents the accuracy rate, which indicates the proportion of predictions that are correct out of all the sample results the model has judged. Precision denotes the precision rate, and Recall represents the recall rate. The F1 score is determined by both Recall and Precision. The F1 score is a composite score that integrates various metrics to more accurately and scientifically evaluate the performance of the model. In this paper, the F1 score and Accuracy score were selected as the evaluation criteria for the model presented in this chapter.

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

3.4 Analysis of Experimental Results

In the experiment, the first step is to compare the performance of various baseline models with the model proposed in this paper on both the LCQMC dataset and the BQ Corpus dataset. Then, the impact of different parts of the ERAMM model on the accuracy of computation will be analyzed. Finally, through two contrastive methods, the influence of the text summarization extraction algorithm on long text matching will be examined. Table 2 presents the results of the models on the LCQMC dataset, and Table 3 shows the results of the models on the BQ Corpus dataset.

Table 2: Results of Various Models on the LCQMC Dataset

	Acc	F1
ABCNN	0.8010	0.7921
Siamese-LSTM	0.8362	0.8391
ESIM	0.8501	0.8407
BIMPM	0.8526	0.8471
ERAMM	**0.8682**	**0.8696**

Table 3: Results of Various Models on the BQ Corpus Dataset

	Acc	F1
ABCNN	0.7987	0.7906
Siamese-LSTM	0.8261	0.8195
ESIM	0.8375	0.8204
BIMPM	0.8417	0.8372
ERAMM	**0.8571**	**0.8618**

From the tables, it is evident that the Enhanced Recurrent Attentional Matching Model (ERAMM) demonstrates the best performance. On the LCQMC dataset, the ERAMM model outperforms the best-performing baseline model, BIMPM, by 1.56% in accuracy and 2.25% in F1 score. On the BQ Corpus dataset, compared to BIMPM, ERAMM achieves improvements of 1.54% in accuracy and 2.46% in F1 score. Through the comparison of the results from different models, it can be observed that the ERAMM model proposed in this paper is superior to the baseline models in all evaluated metrics.

To explore the impact of each part of the model on the overall performance, this paper has designed multiple combinations for comparative experiments to validate the effects of the interactive representation attention mechanism and the fusion layer in the model.

(1) Model Combination 1: Utilizes only the representation attention mechanism without the fusion layer. When computing similarity, the two vectors are directly stitched together for calculation.

(2) Model Combination 2: Does not use the representation attention mechanism, but uses the fusion layer to combine the two vectors, and then applies the fused vector to the text similarity computation.

(3) Model Combination 3: Neither the representation attention mechanism nor the fusion layer is used. When computing similarity, the two vectors are directly stitched together for calculation.

Table 4 presents the results of these various model combinations on the LCQMC dataset.

Table 4: Results of Various Model Combinations on the LCQMC Dataset

	Acc	F1
ERAMM	0.8682	0.8696
Model Combination 1	0.8647	0.8687
Model Combination 2	0.8543	0.8584
Model Combination 3	0.8523	0.8579

Compared with Model Combination One and ERAMM, using only the additional representation of attention mechanisms resulted in a slight decline in performance, with the accuracy dropping by 0.35% and the F1 score by 0.09%. While the decrease was not substantial, it does indicate that the fusion layer can promote vector fusion to some degree, thus having a certain positive effect on similarity computation. When comparing Model Combination Two to ERAMM, Combination Two did not incorporate additional attention mechanisms but utilized the fusion layer. In this case, the accuracy dropped by 1.39% and the F1 score by 1.12%. The greater decline compared to Combination One suggests that the attention mechanism has a more significant impact on improving the accuracy of text similarity calculation. This also demonstrates that the use of self-attention allows key information to be harnessed more effectively, diminishing the influence of non-critical information, which can enhance the matching outcome. In the

ISSN 2616-5775 Vol. 7, Issue 10: 24-30, DOI: 10.25236/AJCIS.2024.071004

comparison between Combination Three and the other results, Combination Three, which did not add either the attention mechanism or the fusion layer, experienced the most significant decline in model performance. This further indicates that the addition of these two structures is advantageous for text similarity computation.

4. Conclusions

This paper first introduces the steps involved in text similarity computation using the ERAMM algorithm, including the specific execution procedures for the input layer, interaction layer, attention mechanism layer, fusion layer, and similarity measurement layer. Secondly, it describes the experimental environment and the relevant dataset used for the text similarity computation experiments, as well as the evaluation indicators for the experimental results. Finally, through related experiments, it compares with common baseline methods to verify the superiority of the proposed text similarity algorithm based on interactive attention over the baseline methods.

References

[1] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition[C]//ICML deep learning workshop. 2015, 2(1): 1-30.
[2] Xia P, Zhang L, Li F. Learning similarity with cosine similarity ensemble[J]. Information sciences, 2015, 307: 39-52.
[3] Bojanowski P, Grave E, Joulin A, et al. Enriching word vectors with subword information[J]. Transactions of the association for computational linguistics, 2017, 5: 135-146.
[4] Vaswani A. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017.
[5] 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.
[6] Yu Y, Si X, Hu C, et al. A review of recurrent neural networks: LSTM cells and network architectures[J]. Neural computation, 2019, 31(7): 1235-1270.
[7] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.
[8] Galassi A, Lippi M, Torroni P. Attention in natural language processing[J]. IEEE transactions on neural networks and learning systems, 2020, 32(10): 4291-4308.
[9] Liu X, Chen Q, Deng C, et al. Lcqmc: A large-scale chinese question matching corpus[C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 1952-1962.
[10] Chen J, Chen Q, Liu X, et al. The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification[C]//Proceedings of the 2018 conference on empirical methods in natural language processing. 2018: 4946-4951.
[11] Chen Q, Zhu X, Ling Z, et al. Enhanced LSTM for natural language inference[J]. arXiv preprint arXiv:1609.06038, 2016.
[12] Wang Z, Hamza W, Florian R. Bilateral multi-perspective matching for natural language sentences[J]. arXiv preprint arXiv:1702.03814, 2017.
[13] Yin W, Schütze H, Xiang B, et al. Abcnn: Attention-based convolutional neural network for modeling sentence pairs[J]. Transactions of the Association for computational linguistics, 2016, 4: 259-272.
[14] Varior R R, Shuai B, Lu J, et al. A siamese long short-term memory architecture for human re-identification[C]//Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14. Springer International Publishing, 2016: 135-153.