OTE: An Optimized Chinese Short Text Matching Algorithm Based on External Knowledge

Haoyang Ma[1,2], Zhaoyun Ding[2(✉)], Zeyu Li[3], and Hongyu Guo[1]

[1] North China Institute of Computing Technology, Beijing, China
guohongyu@sina.com
[2] Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha, China
{mhy99,zyding}@nudt.edu.cn
[3] Communication University of China, Beijing, China
lizeyu@cuc.edu.cn

Abstract. Short text matching is a key problem in natural language processing (NLP), which can be applied in journalism, military, and other fields. In this paper, we propose an optimized Chinese short text matching algorithm based on external knowledge (OTE). OTE can effectively eliminate semantic ambiguity in Chinese text by integrating the HowNet external knowledge base. We use SoftLexicon to optimize the word lattice graph to provide more comprehensive multi-granularity information and integrate the LaserTagger model and EDA for data augmentation. Experimental results show that OTE has an average accuracy improvement of 1.5% in three datasets compared with existing models.

Keywords: Text matching · Multi-granularity information · Data augmentation · External knowledge · Pre-training · Natural language processing

1 Introduction

Short text matching is a critical technology in NLP. Given a pair of sentences, the text matching is to calculate their text similarity. This technology has extensive research needs in question answer systems [1], recommendation systems [2], and public opinion monitoring [3].

However, there are many challenges in studying similarity in Chinese short texts. (a) The limited length of Chinese short texts leads to the sparsity of text features, resulting in unavailability of adequate information. Moreover, traditional models can neither provide adequate semantic information of Chinese, nor offer enough multi-granularity information. (b) The fusion of deep learning technologies improves the accuracy of model matching degrees. In real scenarios, it takes time and effort to obtain label data.

Supported by the Ministry of Science and Technology of China (No. 2020AAA0105100).

G. Memmi et al. (Eds.): KSEM 2022, LNAI 13368, pp. 15–30, 2022.
https://doi.org/10.1007/978-3-031-10983-6_2

As a result, the data scale is not large enough, and the number of label categories is not balanced.

Word segmentation tools are usually used to construct word lattice graphs [4], but insufficient multi-granularity information may sometimes occur. To address this problem, we can use SoftLexicon model [5] to build a word lattice graph to provide adequate multi-granularity information SoftLexicon is an optimized model based on Lattice-LSTM [6]. Lattice-LSTM has a complex model architecture, limiting its application in many industrial fields. SoftLexicon incorporates word lexicon into character representations, avoiding the need to design a complex sequence modelling architecture while easily being used with pre-trained models such as BERT.

Meanwhile, Chinese words may contain many meanings, leading to ambiguity in judgment [7]. As is shown in Fig. 1, the Chinese word "老古董" may mean an old object (antique) or a rigid person (old fogey). In this paper, we use HowNet [8] as an external knowledge source to provide more relevant senses to solve this problem. HowNet, which was put forward in the middle of this century, has been thoroughly improved after more than ten years of development. In HowNet, "老古董" has two different senses, i.e. "antique" and "old fogey". "Old fogey" contains two sememes: "human" and "stiff", which are also the sememes of "stubborn", so we can draw that these two words are highly similar. Therefore, the model can better disambiguate and match two sentences that might be similar.

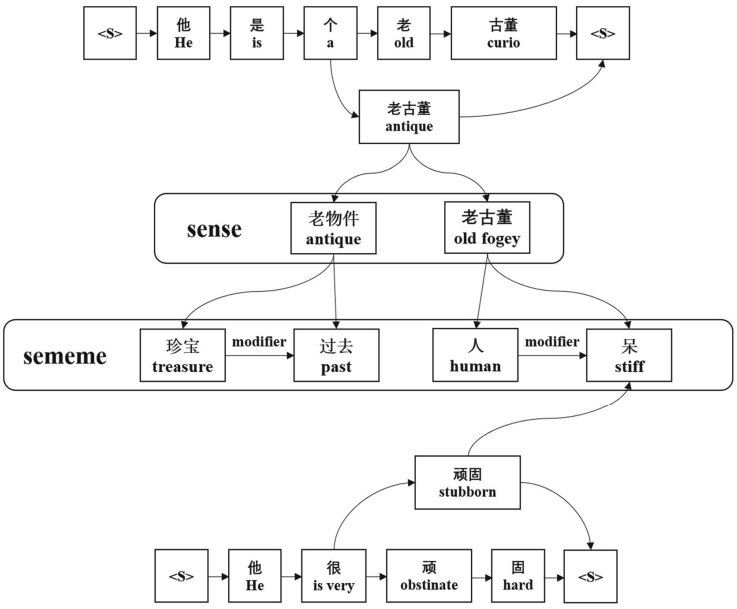

Fig. 1. An example of possible word ambiguity

For the second problem, we use a hybrid data augmentation method based on Easy Data Augmentation (EDA) [9] and text generation model LaserTagger [10]. A random swap strategy augments the original text in the EDA method. Then the EDA improved text and the original text form text pairs into the LaserTagger model to obtain the rephrased text about the input text pairs. The final augmented text received by the hybrid method is mixed with the original text as training data.

In this paper, we mainly complete the following tasks: (a) proposing an optimized short text matching algorithm based on external knowledge by using the SoftLexico model and hybrid data augmentation model; (b) proving that multi-granularity information, semantic information, and data augmentation can improve the accuracy of text matching.

2 Related Work

2.1 Pre-trained Models

Google proposed a pre-trained model named BERT [11] in 2018, which achieved good results in 11 NLP tasks. This model uses the encoder part of Transformer [12] to capture word-level and sentence-level text representations. Therefore, vectors generated by such model are called "dynamic word embeddings". Subsequently, Liu et al. [13] proposed RoBERTa model in 2019 without changing the structure of the BERT model. The BERT pre-training method was optimized by removing the next sentence prediction and introducing dynamic coding tasks to improve the pre-training model's performance. In the same period, many scholars improved the problems in different aspects of BERT and produced a variety of variants. Among them, Lan et al. [14] proposed ALBERT model, a lightweight pre-training model improved based on BERT, to address BERT's shortcomings of high GPU/TPU and longer training times. The two parameter-reduction techniques greatly lowered the memory consumption and increased the training speed of BERT. Zhang et al. [15] proposed ERNIE model, which is an enhanced language representation model trained by large-scale textual corpus and combined knowledge graphs. Yang et al. [16] proposed XLNet model, an extensible autoregressive language model that enables bidirectional prediction by adopting the principle of permutation and combination and overcomes BERT's limitations due to the autoregressive method.

2.2 Data Augmentation

Synonym Replacement (SR) [17] is a simple and intuitive data augmentation method, which generates new text by replacing some words in the original text with their synonyms. The advantage of SR is that it does not destroy the original text information, but the similarity between old and new data is too high. To solve this problem, Wei et al. [9] proposed an easy data augmentation (EDA). In addition to SR, EDA also incorporates random deletion (RD), random swap (RS), and random insertion (RI) for text processing. Still, its excessive reliance on the unexpected way is easy to destroy the context of the text. Sennrich et al. [18] proposed the use of Back Translation for data augmentation in machine translation tasks. The method is used to translate the source language of the corpus into other languages and then back translate it into the source language to obtain new corpus information. Xie et al. [19] used WMT'14 English-French translation models to perform back-translation on sentences. Back-translation can directly call the existing translation software and preserve the original text's context as much as possible. However, its over-reliance on the accuracy of the translation software may introduce a lot of noise to a certain extent. Many scholars have chosen EDA and back-translation due

to their high efficiency and simple operation. Among complex data augmentation methods, Kobayashi [20] proposed Contextual Augmentation (CA), which predicts available candidate words by observing contextual information of the target word through a bidirectional language model, and then randomly selects candidate words to replace the current target word. Hu et al. [21] proposed a model for text generation, VAEHD, which combines variational auto-encoders (VAE) and holistic attribute discriminators. It can learn interpretable latent representations and generate sentences with given sentiment and tense. Although the above three methods can improve the quality of data augmentation to a certain extent, they are not conducive to improving the efficiency of data augmentation due to their high algorithm complexity and high training cost.

2.3 Multi-granularity Information

Multi-granularity information is essential in natural language processing. Different models may cause semantic ambiguity, and different tools may provide different granularities. Lattice LSTM [6] model can obtain multi-granularity sentence expression by using word information without word segmentation. Lattice LSTM encodes the input characters and all matched words in a lexicon into the model, selects the most relevant terms from the glossary to reduce the probability of ambiguity, and considers the input of both character and word granularity. The model has achieved significant improvement in multiple NLP tasks. In particular, in the named entity recognition (NER) task, the Lattice LSTM-based model [22] can encode a sequence of input characters and all potential words that match a lexicon to obtain better NER results. Inspired by the success of Lattice in other NLP tasks, as for the text matching task in 2019, LAI used the lattice-based convolutional neural networks [23] to extract sentence-level features from word lattice. LET [24] proposes a Chinese short text matching method based on word lattice and HowNet. This paper improves on this model.

3 Model

In this paper, we follow the steps below to complete the task of Chinese short text matching. First, we use data augmentation model to augment the data of the training set. Define each text pair as $C^a = \{c_1^a, c_2^a, \ldots, c_{T_a}^a\}$ and $C^b = \{c_1^b, c_2^b, \ldots, c_{T_b}^b\}$. We need method $f(C^a, C^b)$ to predict whether the senses of C^a and C^b are equal.

We propose an optimized short text matching algorithm based on external knowledge. For each text, we use SoftLexicon model to generate a word lattice graph $G = (V, E)$. A word w_i is its corresponding node $x_i \in V$ in the word lattice graph. Then we can obtain all senses through the HowNet. If there is an edge connecting nodes x_i and x_j, we define x_i itself and all its reachable nodes in its forward and backward directions as $N_{fw}^{+-}(x_i)$ and $N_{bw}^{+-}(x_i)$. For each text pair, we have two graphs G^a and G^b, which can be used for similarity prediction.

The schematic diagram of the OTE model structure is shown in Fig. 2. OTE model consists of the following five parts: data augmentation model, input model, semantic information transformer, sentence matching layer, relation classifier.

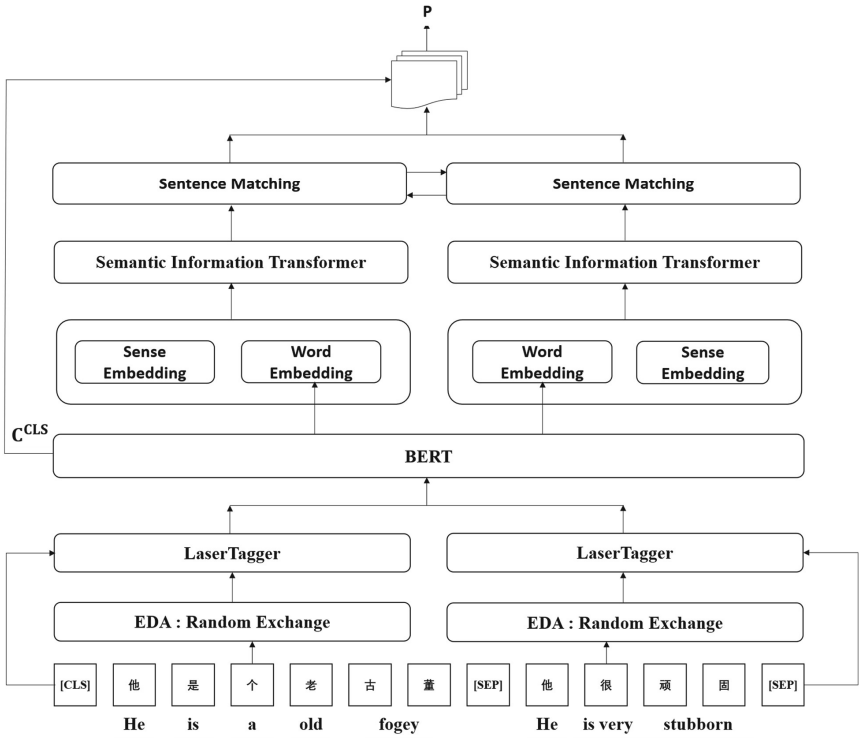

Fig. 2. Schematic diagram of OTE model structure

3.1 Data Augmentation Model

The data augmentation model combines the EDA model and the LaserTagger model. The schematic diagram of the data augmentation model structure is shown in Fig. 3. We adopt the method of random swap, in which two words are randomly selected from a sentence and repeated once, and then their positions are swapped.

The LaserTagger model can complete the text rephrase task and rewrite text A into text B with similar meaning to achieve the effect of data augmentation. The LaserTagger model needs to tag a sequence of characters, and then convert the tag sequences into text. It assigns a tag to each character. A tag is composed of two parts: a base tag and an added phrase. The base tag is either *KEEP* or *DELETE*, and the added phrase is denoted by P. We first align each source text with its target text, that is, use the Longest Common Subsequence (LCS) algorithm [25] to find their longest common substring. Then, all the phrases in the target text that are not part of the LCS are included in the phrase set V, and finally, the most frequent phrases l are selected as the final phrase set V. After the source text with length n_s is converted into tag sequences with the same length, the new text needs to be converted according to the tag of each position. The *KEEP* tag indicates keeping corresponding word, the *DELETE* tag means deleting corresponding word, and the added phrase means adding a phrase before, the corresponding word.

Taking AFQMC [26] dataset as an example, there are 102,477 items in this dataset, including 83,793 items with label 0 and 18,684 items with label 1. The dataset is not pre-divided into training, validation and test sets, so we divide it into these three sets with a ratio of 6:2:2 (a classical ratio for small-scale datasets in machine learning). After division, there are 61,486 training sets, among which 49,352 are labeled 0 and

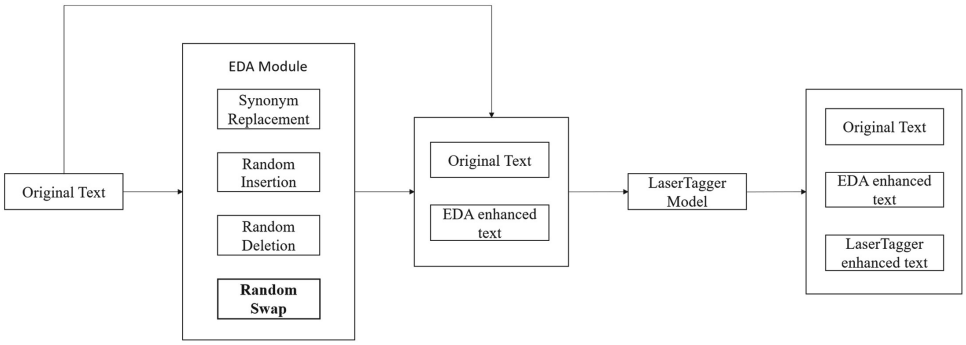

Fig. 3. Schematic diagram of data augmentation model structure

12,134 are labeled 1, with a ratio of 4:1. Thus, problems of small size of training set and unreasonable proportion of labels occur.

To address such problems, we use the model to augment the data labeled 1. After data augmentation, the number of training sets with label 1 reaches 24,270, the label ratio becomes 2:1, and the total number of training sets reaches 73,621. The above-mentioned problems have been well solved.

3.2 Input Model

To generate graph attention network, we use SoftLexicon model to generate a word lattice graph $G = (V, E)$. SoftLexicon divides all matched words of each character into four word sets $seq = \{B, M, E, S\}$, which represents a set of words with characters in different positions. The specific formula is as follows,

$$B(c_i) = \left\{w_{i,k}, \forall w_{i,k} \in L, i < k \leq n\right\},$$

$$M(c_i) = \left\{w_{j,k}, \forall w_{j,k} \in L, 1 \leq j < i < k \leq n\right\},$$

$$E(c_i) = \left\{w_{j,i}, \forall w_{j,i} \in L, 1 \leq j < i\right\},$$

$$S(c_i) = \{c_i, \exists c_i \in L\} \tag{1}$$

Where L stands for lexicon. Then the word set is compressed, mainly to compress each category of word embedding into one embedding, using the word weighting method as follow,

$$\boldsymbol{v}^s(S) = \frac{4}{Z} \sum\nolimits_{w \in S} z(w) \boldsymbol{e}^w(w), \tag{2}$$

where S represents the word set, and $Z = \sum_{w \in \text{BUMUEUS}} z(w)$. The frequency of occurrence of each word in a static dataset is used as the weight to speed up the training. Meanwhile, if w is overwritten by another subsequence that matches the lexicon, the frequency of w will not increase. This prevents the problem that the frequency of the shorter word is always less than that of the longer word overwriting it.

Then we need generate a graph attention network base on word lattice graph. In 2018, Petar et al. [27] proposed a graph attention network applied to graph structured data. The set of all nodes connected to x_i is denoted by $N^+(x_i)$. We use h_i and $h_i^{'}$ to represent the feature vector and the new feature vector of the node x_i. The weight coefficient of neighboring node x_j to x_i can be set as $\alpha_{ij} = a(Wh_i, Wh_j)$. After calculation, the degree of relevance between x_i and all neighboring nodes can be obtained. After normalization by softmax, the attention weight of x_i and all neighboring nodes can be obtained. The weighted average value of updated node h_i^l can be calculated as:

$$h_i^l = \sigma\left(\sum\nolimits_{x_j \in \mathcal{N}^+(x_i)} \alpha_{ij}^l \cdot \left(W^l h_j^{l-1}\right)\right) \tag{3}$$

To avoid the possible limitations in the capacity to model complex dependencies, Shen et al. proposed a multi-dimensional attention mechanism [28]. For each h_j^{l-1}, a feature-wise score vector is first calculated and then normalized using feature-wise multi-dimensional softmax, which is denoted by β,

$$\alpha_{i,j}^l = \beta_j\left(\widehat{\alpha}_{i,j}^l + f_m^l\left(h_j^{l-1}\right)\right), \tag{4}$$

where f_m^l is used to estimate the contribution of each feature dimension of h_j^{l-1},

$$f_m^l\left(h_j^{l-1}\right) = W_2^l \sigma\left(W_1^l h_j^{l-1} + b_1^l\right) + b_2^l, \tag{5}$$

then the Eq. (3) can be revised as:

$$h_i^l = \sigma\left(\sum_{x_j \in \mathcal{N}^+(x_i)} \alpha_{ij}^l \odot \left(W^l h_j^{l-1}\right)\right) \tag{6}$$

We need to input text pairs into the BERT model to get a contextual representation of each character as $\{c^{CLS}, c_1^a, c_2^a, \ldots, c_{T_a}^a, c^{SEP}, c_1^b, c_2^b, \ldots, c_{T_b}^b, c^{SEP}\}$. Then we use a feed forward network to obtain a feature-wise score vector for each character, which is denoted by γ. After that, we can normalize it with feature-wise multi-dimensional softmax,

$$u_k = \beta_k(\gamma(c_k)), \tag{7}$$

then we can obtain contextual word embedding:

$$v_i = \sum\nolimits_{k=t_1}^{t_2} u_k \odot c_k \tag{8}$$

To get sense embedding, we need to use HowNet. HowNet has a well-established sense and sememe architecture. As an example of the HowNet structure, the Chinese word "老古董" has two senses, i.e. "antique" and "old fogey", and the "old fogey" has two sememes: "human" and "stiff". HowNet makes it easier to calculate whether two sentences match.

We generate the set of the senses as $S^{w_i} = \{s_{i,1}, s_{i,2}, \ldots, s_{i,k}\}$ for each word w_i, then generate the set of sememes as $O^{s_{i,k}} = \{o^1_{i,k}, o^2_{i,k}, \ldots, o^n_{i,k}\}$ for each sense. We use sememe attention over target model [29] to calculate each sememe's embedding vector $e^n_{i,k}$, then use multi-dimensional attention function to calculate each sememe's representation $o^n_{i,k}$ as:

$$o^{n'}_{i,k} = \beta(e^n_{i,k}, \{e^{n'}_{i,k} | o^{n'}_{i,k} \in O^{s_{i,k}}\}) \tag{9}$$

For the embedding of each sense, we can obtain it with attentive pooling of all its sememe representations:

$$s_{i,k} = AP(\{o^n_{i,k} | o^n_{i,k} \in O^{s_{i,k}}\}) \tag{10}$$

3.3 Semantic Information Transformer

Contextual information is now separated from semantic information. In order to get more useful information, we propose a word lattice graph transformer. For word w_i, we use v_i and $s_{i,k}$ as the original word representation h^0_i, the sense $s_{i,k}$ as the original sense representation $g^0_{i,k}$. Then update them iteratively.

To update sense representation from $g^{l-1}_{i,k}$ to $g^l_{i,k}$, we need both backward information and forward information of x_i, then update its representation with a gated recurrent unit (GRU) [30],

$$m^{l,bw}_{i,k} = \beta\left(g^{l-1}_{i,k}, \left\{h^{l-1}_j | x_j \in N^+_{bw}(x_i)\right\}\right),$$

$$m^{l,fw}_{i,k} = \beta\left(g^{l-1}_{i,k}, \left\{h^{l-1}_j | x_j \in N^+_{fw}(x_i)\right\}\right),$$

$$g^l_{i,k} = GRU(g^{l-1}_{i,k}, m^l_{i,k}) \tag{11}$$

Where $m^l_{i,k} = \{m^{l,bw}_{i,k}, m^{l,fw}_{i,k}\}$. We use GRU to control the mix of contextual information and semantic information because $m^l_{i,k}$ contains contextual information merely. Then we use $g^l_{i,k}$ to update the word representation from h^{l-1}_i to h^l_i. The transformer uses multi-dimensional attention to obtain the first sense of word w_i from its semantic information, then updates it with GRU.

$$q^l_i = \beta\left(h^{l-1}_i, \left\{g^l_{i,k} | s_{i,k} \in S^{w_i}\right\}\right),$$

$$h^l_i = GRU(h^{l-1}_i, q^l_i) \tag{12}$$

3.4 Sentence Matching Layer

To incorporate word representation into characters, we use characters in C^a as example. We generate a set $W^{c_t^a}$ that contains the words using character c_t^a, then use attentive pooling to get $\hat{c}_t^a$ of each character:

$$\hat{c}_t^a = AP(\{h_i^a | w_i^a \in W^{c_t^a}\}) \tag{13}$$

After obtaining $\hat{c}_t^a$ and c_t^a, we use layer normalization to get semantic information enhanced character representation y_t^a:

$$y_t^a = LN(\hat{c}_t^a + c_t^a) \tag{14}$$

Then for each character c_t^a, we can use multi-dimensional attention to obtain its aggregative information from C^a and C^b, which are denoted by m_t^s and m_t^c. When they are almost equal, we can know that these two sentences are matched. To compare them, we need use multi-perspective cosine distance [31],

$$m_t^s = \beta\left(y_t^a, \left\{y_{t'}^a | c_{t'}^a \in C^a\right\}\right),$$

$$m_t^c = \beta\left(y_t^a, \left\{y_{t'}^a | c_{t'}^b \in C^b\right\}\right),$$

$$d_k = CD\left(w_k^{cos} \odot m_t^s, w_k^{cos} \odot m_t^c\right) \tag{15}$$

where w_k^{cos} represents the different weights of different dimensions of a text. We can get the final character representation $d_t := [d_1, d_2, \ldots, d_k]$ using feed forward networks, then use attentive pooling to obtain the sentence representation vector:

$$\hat{y}_t^a = \gamma\left(\left[m_t^s, d_t\right]\right),$$

$$r^a = AP(\hat{y}_t^a | \hat{y}_t^a \in \hat{Y}^a) \tag{16}$$

3.5 Relation Classifier Layer

Finally, our model can predict the similarity between two sentences by using r^a, r^b, and c^{CLS}.

$$P = \gamma\left(\left[c^{CLS}, r^a, r^b, \left|r^a - r^b\right|, r^a \odot r^b\right]\right). \tag{17}$$

For each $\left\{C_i^a, C_i^b, y_i\right\}$ raining sample, our ultimate goal is to reduce the BCE loss:

$$\mathcal{L} = -\sum\nolimits_{i=1}^{N} (y_i log(p_i) + (1 - y_i)\log(1 - p_i)) \tag{18}$$

where $y_i \in \{0, 1\}$ is the label of the i-th training sample we input to the model and $p_i \in [0, 1]$ is the prediction of our model taking sentence pairs as input.

4 Experiment

4.1 Experiment Dataset

In the experimental part, we use three Chinese datasets, i.e. LCQMC [32], AFQMC [26] and BQ [33], to test our model.

The LCQMC dataset is a question semantic matching dataset constructed by Harbin Institute of Technology in COLING2018. Its format consists of sentence pair number, two sentences to be compared and 4 columns of similarity labels. It contains 260,068 pieces of data in total, including 238,766 for training set, 12,500 for test set and 8,802 for validation set. LCQMC is widely used in Chinese short text similarity calculation.

The AFQMC dataset is the dataset of ANT Financial ATEC: NLP Problem Similarity Calculation Competition, and it is a dataset for classification task. All data are from the actual application scenarios of Ant Financial's financial brain, that is, two sentences described by users in a given customer service are determined by algorithms to determine whether they represent the same semantics. Synonymous sentences are represented by 1, non-synonymous sentences are represented by 0, and the format is consistent with LCQMC dataset. Since the AFQMC dataset was not pre-divided into training set, test set and validation set, we divide the dataset by the ratio of 6:2:2 in this experiment. The data volumes of training set, test set and validation set are 61,486, 20,496, and 20,495, respectively, and the total number of all samples is 102,477.

The BQ dataset is a question matching dataset in the field of banking and finance. Comprising question text pairs extracted from one year of online banking system logs, it is the largest question matching dataset in the banking domain. It classifies two paragraphs of bank credit business according to whether they are semantically similar or not. 1 represents the similarity judgment while 0 represents the dissimilarity judgment. The BQ dataset contains 120,000 pieces of data in total, including 100,000 for training set, 10,000 for validation set and 10,000 for test set.

4.2 Experiment Result

Accuracy (ACC.) and F1 score are used as the evaluation metrics. The calculation formula of ACC is as follows,

$$ACC. = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

Where True Positive (TP) indicates the number of cases correctly predicted as positive, False Positive (FP) indicates the number of cases incorrectly predicted as positive, True Negative (TN) indicates the number of cases correctly predicted as negative, False Negative (FN) indicates the number of cases incorrectly predicted as negative. Based on this, precision rate (P), Recall (R) and F1 values can be calculated. Their calculation formulas are shown below:

$$P = \frac{TP}{TP + FP}. \tag{20}$$

$$R = \frac{TP}{TP + FN}. \tag{21}$$

$$F1 = \frac{2 * P * R}{P + R}. \tag{22}$$

We use SoftLexicon model to generate the word lattice graph, use OpenHowNet to acquire external knowledge for semantic information embedding. We use a method that integrates the LaserTagger model and EDA for data augmentation. In order to prevent over-fitting, 50% down-sampling is performed on the original data. The batch size of LCQMC, AFQMC and BQ is 32,32 and 64, respectively. The number of epoch is set to 4. Word representation, sense representation and hidden layer's dimension are all 128.

Data Augmentation Validation. In order to verify the effectiveness of data augmentation, taking AFQMC dataset as an example, we design three groups of comparative experiments, the training sets of which are original text, downsampling and data augmentation, respectively. In the downsampling method, 50% of the training set labeled 0 were randomly selected as the final training samples. Therefore, the training set samples for the final experiment were 24,622 less than the original training set. Data augmentation augments the data labeled as 1 to obtain a dataset with 2:1 of 0 and 1 labels. Compared with the original text method and the downsampling method, the accuracy of the data augmentation method is improved by 3.7% and 3%, respectively. The experimental results are shown in Table 1. We think this is due to the large deviation of the label ratio in the original text. Although the downsampling method solves the problem of ratio, it causes the loss of original data.

Table 1. Comparison of data augmentation results

Method	ACC.	F1
Original	72.98	71.87
Downsampling	73.43	73.95
Data Augmentation	**75.69**	**75.88**

To verify the effectiveness of hybrid data augmentation, we compare the original text, the text generated by back translation, and the text generated by combining LaserTagger and EDA. Back translation means that Chinese text is converted into English text and then back translated into Chinese text through translation tools. In this experiment, Youdao Translation and Google Translation are used separately to achieve this step. Taking AFQMC dataset as an example, the experimental results are shown in Table 2. Compared with original text and back translation, the hybrid method improves the accuracy by 2.9% and 2.3% respectively. Therefore, the hybrid data augmentation method can effectively avoid the introduction of noise to the data-augmented text and improve the accuracy of the Chinese short text semantic similarity calculation model.

Multi-granularity Information Validation. In order to verify the impact of multi-granularity information on the model, we also set up a comparative experiment. The

Table 2. Comparison of hybrid data augmentation results

Method	ACC	F1
Original	73.56	74.22
Back translation	74.01	74.78
Hybird	**75.69**	**75.88**

experiment is divided into three categories: no word segmentation, jieba and word lattice graph. AFQMC is also used as an example in this experiment, and the experimental results are shown in Table 3. As can be seen from the experimental results, the accuracy of lattice is improved by 2.7% and 2.1% respectively compared with that of no word segmentation and jieba. We believe that jieba does not provide enough multi-granularity information, and there may be word segmentation errors, leading to the insignificant improvement of accuracy. This experiment proves that the introduction of multi-granularity information can effectively improve the accuracy of Chinese short text similarity calculation.

Table 3. Multi-granularity information results comparison

Method	ACC.	F1
No	73.72	74.04
Jieba	74.13	74.22
SoftLexicon	**75.69**	**75.88**

Semantic Information Validation. At the same time, in order to test the effectiveness of semantic information, we also set up the experiment without HowNet. Because short texts don't contain enough contextual information, HowNet can provide more semantic information. In this experiment, we remove the updating and embedding of semantic information in the model. Taking AFQMC dataset as an example, through experimental comparison, there is a 1.4% decrease in accuracy and 0.9% decrease in F1 score after removing HowNet. This experiment proves that semantic information provided by integrating external knowledge can increase the accuracy of Chinese short text similarity calculation.

Ablation Experiment. To demonstrate the effectiveness of the joint use of data augmentation and multi-granularity information, we set up an ablation experiment, which is divided into four groups: Neither, Only DA (data augmentation), Only MI (multi-granularity information) and Both When data augmentation is not included, original text is used as the training sample. When multi-granularity information is not included, we cancel the embedding of semantic information and only use BERT as word embedding. The experimental results are shown in Table 4. The bar chart is shown in Fig. 4.

According to the experimental data, compared with Neither group, the Only DA and Only MI can improve the accuracy by 1.5% and 2.4%, respectively. As for the Both group, the accuracy is improved by 5.4%. We believe that because the multi-granularity information model contains more semantic information, the improvement is greater than the data augmentation. Therefore, it can be proved that the combination of data augmentation and multi-granularity information can better improve the accuracy of similarity calculation of Chinese short texts.

Table 4. Ablation experiment

DA	MI	ACC	F1
×	×	71.79	71.86
√	×	72.87	71.95
×	√	73.53	73.72
√	√	**75.69**	**75.88**

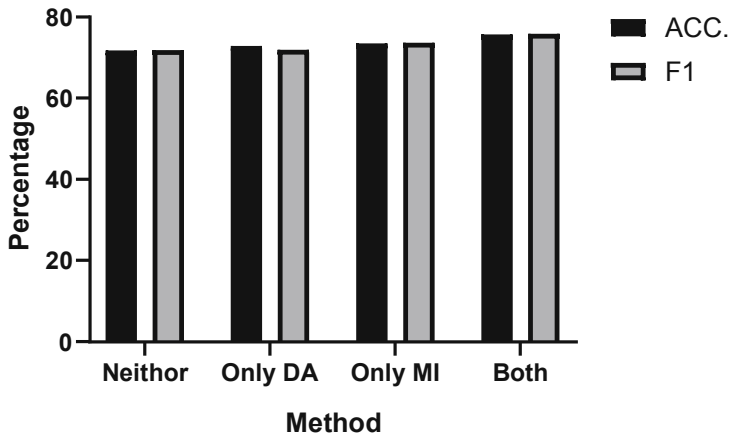

Fig. 4. Ablation experiment

Model Comparison. For the control group, we select BiLSTM [34], ERNIE [15], BERT [11], and BERT-wwm-ext [35] models(BERT(W)). The comparison of experimental results with other models is shown in Table 5. The accuracies of LCQMC, AFQMC and BQ are increased by 1.8%, 2.2% and 0.6%, respectively. The F1 scores of LCQMC, AFQMC and BQ are increased by 1.1%, 0.7% and 1%, respectively.

Table 5. Comparison of experimental results with other models

MODEL	LCQMC		AFQMC		BQ	
	ACC	F1	ACC	F1	ACC	F1
BiLSTM	76.10	78.90	64.68	54.53	73.51	72.68
ERNIE	87.04	88.06	73.83	73.91	84.67	84.20
BERT	85.73	86.86	73.70	74.12	84.50	84.00
BERT(W)	86.68	87.77	74.07	74.35	84.71	83.94
OTE	**88.29**	**88.72**	**75.69**	**75.88**	**85.26**	**84.77**

The bar chart for comparison of experimental results is shown in Fig. 5.

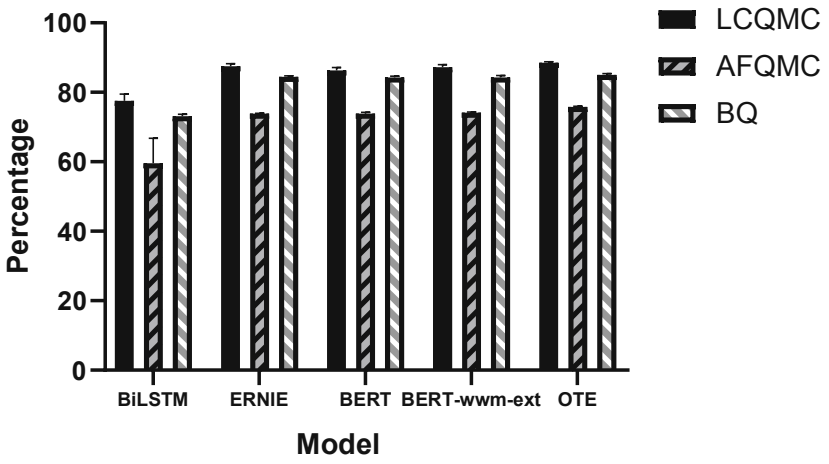

Fig. 5. Comparison of experimental results with other models

BiLSTM is a bidirectional long short-term memory network that can. BERT is essentially a two-stage NLP model. The first stage is called pre-training, which, like word embedding, trains a language model using existing unlabeled corpus. The second stage is called fine-tuning, which uses pre-trained language models to complete specific downstream tasks of NLP. Both Bert-wwm-ext and ERNIE are BERT variants. ERNIE aims to learn the language representation enhanced by knowledge masking strategy. Bert-wwm-ext mainly changes the generation strategy of training samples in the original pre-training stage, and increases the training dataset and the number of training steps. According to the experimental results, compared with BiLSTM, OTE has a great improvement, and also has a certain improvement as compared to BERT-based models. The experimental results show that OTE has an average accuracy improvement of 16.3% as compared to BiLSTM. Compared with BERT-based models, the accuracy has been improved to a certain extent. Among them, the accuracy is improved by 2% on average compared with BERT model and 1.3% on average compared with ERNIE model.

5 Conclusion

In this paper, we propose an optimized short text matching algorithm based on external knowledge, using HowNet as an external data source to generate semantic knowledge. We use SoftLexicon to optimize the word lattice graph to provide more comprehensive multi-granularity information and integrate the LaserTagger model and EDA for data augmentation. We have obtained good experimental results. Compared with other pre-training models, it is proved that multi-granularity information, semantic information and data augmentation can better improve the accuracy of the model.

References

1. Tan, M., Dos Santos, C., Xiang, B., Zhou, B.: Improved representation learning for question answer matching. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 464–473 (2016)
2. Chen, H.: Personalized recommendation system of e-commerce based on big data analysis. J. Interdisc. Math. **21**, 1243–1247 (2018)
3. Kilimci, Z., Omurca, S.: Extended feature spaces based classifier ensembles for sentiment analysis of short texts. Inf. Tech. Control. **47**(3), 457–470 (2018)
4. Chen, L., et al.: Neural graph matching networks for Chinese short text matching. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6152–6158 (2020)
5. Ma, R., Peng, M., Zhang, Q., Huang, X.: Simplify the usage of lexicon in Chinese NER. arXiv preprint arXiv:1908.05969 (2019)
6. Zhang, Y., Wang, Y., Yang, J.: Lattice LSTM for Chinese sentence representation. IEEE/ACM Trans. Audio Speech Lang. Process. **28**, 1506–1519 (2020)
7. Xu, J., Liu, J., Zhang, L., Li, Z., Chen, H.: Improve Chinese word embeddings by exploiting internal structure. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1041–1050 (2016)
8. Dong, Z., Dong, Q.: HowNet-a hybrid language and knowledge resource. In: International Conference on Natural Language Processing and Knowledge Engineering, 2003, Proceedings. 2003, pp. 820–824. IEEE (2003)
9. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint: arXiv:1901.11196 (2019)
10. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.: Encode, tag, realize: high-precision text editing. arXiv preprint arXiv:1909.01187 (2019)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding (2018)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
13. Liu, Y., et al.: RoBERTa: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
14. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
15. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: enhanced language representation with informative entities. arXiv preprint arXiv:1905.07129 (2019)

16. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
17. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. Adv. Neural Inf. Process. Syst. **28**, 649–657 (2015)
18. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709 (2015)
19. Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., Le, Q.V.: Unsupervised data augmentation for consistency training. arXiv preprint arXiv:1904.12848 (2019)
20. Kobayashi, S.: Contextual augmentation: data augmentation by words with paradigmatic relations. arXiv preprint arXiv:1805.06201 (2018)
21. Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Toward controlled generation of text. In: International Conference on Machine Learning, PMLR, pp. 1587–1596 (2017)
22. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. arXiv preprint arXiv:1805.02023 (2018)
23. Lai, Y., Feng, Y., Yu, X., Wang, Z., Xu, K., Zhao, D.: Lattice CNNs for matching based Chinese question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6634–6641 (2019)
24. Lyu, B., Chen, L., Zhu, S., Yu, K.: LET: linguistic knowledge enhanced graph transformer for Chinese short text matching. arXiv preprint arXiv:2102.12671 (2021)
25. Hirschberg, D.S.: Algorithms for the longest common subsequence problem. J. ACM (JACM). **24**, 664–675 (1977)
26. Xu, L., Zhang, X., Dong, Q.: CLUECorpus2020: a large-scale Chinese corpus for pre-training language model. arXiv preprint arXiv:2003.01355 (2020)
27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
28. Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., Zhang, C.: DiSAN: directional self-attention network for RNN/CNN-free language understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence (2018)
29. Niu, Y., Xie, R., Liu, Z., Sun, M.: Improved word representation learning with sememes. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2049–2058 (2017)
30. Caruana, R.: Learning many related tasks at the same time with backpropagation. In: Advances in Neural Information Processing Systems, pp. 657–664 (1995)
31. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. arXiv preprint arXiv:1702.03814 (2017)
32. Liu, X., Chen, Q., Deng, C., Zeng, H., Chen, J., Li, D., Tang, B.: LCQMC: a large-scale Chinese question matching corpus. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1952–1962 (2018)
33. Chen, J., Chen, Q., Liu, X., Yang, H., Lu, D., Tang, B.: The BQ corpus: a large-scale domain-specific Chinese corpus for sentence semantic equivalence identification. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 4946–4951 (2018)
34. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: Proceedings of the AAAI Conference on Artificial Intelligence (2016)
35. Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv arxiv:1906.08101 (2019)