*Intelligent Automation & Soft Computing*
DOI:10.32604/iasc.2021.017648
***Article***

Tech Science Press

# Short Text Entity Disambiguation Algorithm Based on Multi-Word Vector Ensemble

**Qin Zhang[1], Xuyu Xiang[1,*], Jiaohua Qin[1], Yun Tan[1], Qiang Liu[1] and Neal N. Xiong[2]**

[1]College of Computer Science and Information Technology, Central South University of Forestry and Technology, Changsha, 410004, China
[2]Department of mathematics and computer science, Northeastern State University, OK, 74464, USA
[*]Corresponding Author: Xuyu Xiang. Email: xyuxiang@163.com
Received: 05 February 2021; Accepted: 16 April 2021

**Abstract:** With the rapid development of network media, the short text has become the main cover of information dissemination by quickly disseminating relevant entity information. However, the lack of context in the short text can easily lead to ambiguity, which will greatly reduce the efficiency of obtaining information and seriously affect the user's experience, especially in the financial field. This paper proposed an entity disambiguation algorithm based on multi-word vector ensemble and decision to eliminate the ambiguity of entities and purify text information in information processing. First of all, we integrate a variety of unsupervised pre-trained word vector models as vector embeddings according to different word vector models' characteristics. Moreover, we use the classic architecture of long short-term memory (LSTM) combined with the convolutional neural network (CNN) to fine-tune pre-trained Chinese word vectors such as BERT to integrate the output of entity recognition results. Then build the knowledge base and introduce the focal loss function on the basis of CNN and binary classification to improve the effect of entity disambiguation. Experimental results show that the algorithm performs better than the traditional entity disambiguation algorithm based on the single word vector. This method can accurately locate the entity to be disambiguated and has a good performance in disambiguation accuracy.

**Keywords:** Named entity recognition; named entity disambiguation; BERT; focal loss

## 1 Introduction

Driven by the rapid development of informatization, the Internet has entered an era of data explosion, in which text data represented by Chinese short text plays an important role. However, the ubiquitous ambiguous words bring great confusion to natural language processing (NLP).

To solve the ambiguity problem in the entities with the same name, the method named entity disambiguation is proposed. It is a process that the ambiguous entities in natural language texts point to the corresponding entities in the knowledge base accurately [1]. The traditional methods mainly use the similarity measurement based on the bag-of-words model or the natural language processing technology based on syntax analysis to judge whether two target entity words have the same meaning. However, there have been no major breakthrough results on the whole. Therefore, it has been a hot spot in natural language processing in recent years how to quickly and accurately locate the named entities of the required company names and their actual meanings from the massive Chinese short text data.

Recently, as deep learning has made significant progress in the field of image processing [2–4] and steganography technology [5–7], it has also received widespread attention in natural language processing technology [8,9]. The vigorous development of word embedding and neural networks provides a new solution to the named entity disambiguation task. The entity disambiguation method based on artificial intelligence mainly starts from three aspects: entity knowledge base construction, named entity recognition, and named entity linking. In terms of entity knowledge base construction, Du et al. [10] used Chinese Wikipedia as world knowledge. They used the word sense options contained in the Wikipedia named entity's disambiguation page to be disambiguated as the candidate named entity to disambiguate the Chinese named entity. However, this work mainly focused on the named entity of the general domain, but not considering the entity knowledge of other non-universal domains. Chen et al. [11] proposed a subdivided domain entity-relationship discovery scheme combining domain metaknowledge and word embedding vector analogy, which can achieve a good performance on the entity-relationship recognition with only a small amount of domain knowledge extracted from the encyclopedia.

In terms of named entity recognition, Peng [12] used the general field text content and part of financial field data as annotated data sets to train the named entity recognition model based on the bidirectional LSTM neural network and conditional random fields (CRF). Experiments showed that this method helped to improve the accuracy of named entities in the financial field. In terms of named entity linking, Fang et al. [13] used deep learning methods to study text semantics. Xue et al. [14] proposed the Recurrent Random Walk based EL (RRWEL) model, which uses CNN to learn the semantic representation of local context, mention, entity, and type information. This model also used the random walk network to learn document information and combined local information with global information to get the correct entity corresponding to each document's reference. Francis-Landau et al. [15] used CNN to learn the text's representation and then obtained the cosine similarity score and text vector of candidate entity vector.

However, many existing works on named entity disambiguation are oriented only towards long text in the common domains. Since the long text has complete context information, the entity's true meaning can be judged based on the entity context clues and the existing prior knowledge background, which is helpful to entity recognition and entity disambiguation to a certain extent. From the current situation, the existing methods of entity disambiguation have only achieved good results in limited fields and limited entity types, but these technologies cannot be well migrated to other specific fields [16]. Compared with the existing entity disambiguation methods, the Chinese short text disambiguation context is not rich, contains more polysemous words, and the syntactic structure is more complex, making it more difficult to understand the knowledge base and semantics. This situation puts forward a higher requirement for the disambiguation algorithm.

On the one hand, the traditional single word vector-based entity disambiguation algorithm is difficult to deal with the short text. On the other hand, the named entity disambiguation technology has limitations in the short text entity of the financial field. In this case, this paper proposes a financial entity disambiguation algorithm based on multi-word vector ensemble and decision.

The main contributions of this paper are as follows:

1) Build a financial entity knowledge base based on big data. We use crawler technology to capture the relevant entity description data on the network and obtain entity terms. Simultaneously, we splice the text description of data set and expand definition for auxiliary construction, making full use of the annotated data to ensure the knowledge base's integrity.

2) Integrate several pre-trained word vector representation models to improve the accuracy of entity recognition. Different models have their own characteristics, and this method can give full play to each model's advantages.

3) Introduce the focal loss function to improve the accuracy of disambiguation. By analyzing the characteristics of Chinese short texts, we combine LSTM and CNN for feature extraction, introduce the focal loss function, and reduce the weight of easy-to-classify samples so that the model focuses more on difficult-to-classify samples during training.

4) Improve the efficiency of financial entity disambiguation models under the premise of high accuracy. In this paper, we transform the entity disambiguation into a binary classification problem, use the fast-training CNN to extract local features and replace Word2vec with BERT to ensure the training speed further.

The organization of the remaining part is given as follows. We describe the related work in Section 2 and detailly introduces the proposed method in Section 3. Section 4 gives extensive experimental evaluations. Finally, Section 5 concludes this paper.

## 2 Related Work

The text entity disambiguation based on deep learning can be divided into three steps. The first step is to transform words into digital vectors in the dimension or use a pre-trained vocabulary (Word Embeddings), a vector representation of words for generating their contexts [17], to represent vector sets. The second step is to pass the word vectors to the neural network for training and learning. The third step is to adjust the network parameters according to the loss function. This section introduces several pre-trained word embedding methods first and then introduces the feature extraction method of LSTM + CNN and the used focal loss.

### *2.1 Pre-Trained Word Vector Representation*

In 2013, Mikolov et al. [18] proposed the concept of word vectors and invented Word2vec, a software tool for training word vectors. It can infer the meaning of a word from its context according to the assumptions of the two language models: Skip-gram and CBOW. The Skip-gram model predicts words in their context by target words while CBOW predicts target words in their contexts by words. However, there are many parameters in the word vector training model, and so a larger data set is needed to train these parameters for avoiding overfitting, which requires a high cost to obtain large-scale annotated data sets. By contrast, the unlabeled *corpus* was easier to build [19]. In this case, pre-training was first proposed as an effective regularization method by Dai in 2015 [20]. The weight of the pre-trainied model trained on the large dataset was used as the initial weight and then trained on the small datasets to update the weights [21]. This method reduced the risk of model overfitting and accelerated the convergence rate of the model.

In 2018, Devlin et al. [22] of Google AI team released a deep bidirectional pre-trained word vector representation model Bidirectional Encoder Representations from Transformers (BERT) model, which has been the most breakthrough development in natural language processing. BERT adopts two pre-training methods to obtain the features of expression words and sentences: Masked Language Model (Masked LM) with trained two-way characteristics and Next Sentence Prediction (NSP) linked to capturing a sentence. To improve the accuracy of the traditional word segmentation method based on BERT in Chinese recognition, in 2019, Cui et al. [23] of Harbin Institute of technology and iFLYTEK jointly proposed BERT's improved model, namely BERT-wwm, to mask the whole word. In BERT-wwm, if the parts of a complete word are covered, other parts of the same word will also be covered. Besides, this laboratory combined Chinese Whole Word Masking technology and the RoBERTa model to publish the Chinese Roberta-wwm-ext pre-trained model, which intensified the input data's randomness. Simultaneously, the NSP task is canceled, and more Chinese semantic information is added to the knowledge map to improve the model's learning ability. In the same year, researchers from Tsinghua

University and Huawei proposed Enhanced Representation from knowledge Integration (ERNIE) [24] and the amounts of data. This model was pre-trained by masked semantic units such as word and entity concepts. It maked full use of vocabulary, syntax, and knowledge information to learn the semantic representation of complete concepts, making the representation of semantic knowledge units of models more realistic.

The three pre-trained word vector models above improved BERT's training sample generation strategy, dynamic coverage mechanism, and the ability of general semantic representation in the pre-training stage. They all achieved better results than the BERT model on several Chinese NLP tasks.

### *2.2 LSTM + CNN*

The ability of local feature extraction of the pre-trained word vector model of deep learning and its ability to deal with long-term dependent timeliness problems cannot be compared with that of the machine learning model. This paper combines the machine learning model with the deep learning model. The previous studies have shown that training models such as CNN and LSTM are more effective. Convolutional neural network (CNN) [25] is a deep learning algorithm based on traditional neural networks, which has advantages in focusing on its fast-training speed, constantly correcting the weights of each layer during training and extracting sequence features and sentence encoding [26]. For sequence modeling problems, LSTM [27] is an improved method for classifying samples using time series data. It introduces the forgetting gate to solve the gradient disappearing problem, and thus has a strong ability to extract long sequence features. Each prediction result of the LSTM neural unit will be combined with the latter word as a feature and continue to predict, realizing the effective extraction and integration of the context information and ensuring the prediction result's accuracy.

In our algorithm, we introduce the LSTM layer as a feature extraction tool. After the LSTM layer encodes the word vectors input, we transform the rich encoded information into a NER annotation sequence and train it. The probability of features being labeled can be obtained by learning the mapping of features to label resulting.

The combination of CNN and LSTM can simultaneously use CNN's ability to identify local features and LSTM's ability to extract features. Sosa [28] found that the performance of the LSTM + CNN model was 8.5% higher than the CNN model alone and 2.7% higher than the LSTM model alone. Inspired by this work, this paper selects to use the LSTM + CNN model to process the previously obtained word vector. We use LSTM to extract comprehensive text features and input them into the CNN network model to mine local associations in text, improving name entity recognition accuracy.

### *2.3 Loss Function*

In general, the cross-entropy loss function is used in binary classification, measuring the degree of difference between two different probability distributions in the same random variable. In machine learning, it is expressed as the difference between the true probability distribution and the predicted probability distribution. The smaller the value of cross-entropy, the better the prediction effect of the model. The cross-entropy loss function of the binary classification task is expressed as:

$$L = -y \log\, y' - (1 - y)\log(1 - y') = \begin{cases} -log\, y', & y = 1 \\ -\log(1 - y'), & y = 0 \end{cases} \tag{1}$$

where $y'$ is the output between 0 and 1 after the activation function.

There will always be such a problem for classification models: the optimization target is inconsistent with the assessment index. Generally, when used as a loss function, cross-entropy's source is the maximum likelihood estimation. However, the final evaluation goal is not to focus on how small the cross-entropy is, but to focus on the model's accuracy. In general, the accuracy rate will be high when the cross-entropy is small, but this relationship is not inevitable. So how to improve the correlation between them is particularly important. To solve this problem, Lin et al. [29] proposed the focal loss function, which measured the contribution of hard-to-classify and easy-to-classify samples to the total loss. This paper chooses to use the focal loss function to make the model pay more attention to the difficult samples to classify and misclassify.

## 3 The Proposed Disambiguation Algorithm

In this section, we will elaborate on the proposed disambiguation algorithm. It is mainly divided into three parts: knowledge base construction, named entity recognition, and entity disambiguation. Among them, named entity recognition is the premise of the entity disambiguation step, and the other parts together constitute a disambiguation system. The built knowledge base will be used for subsequent entity recognition and entity disambiguation.

### *3.1 The Framework*

In this paper, the named entity recognition adopts the fine-tuning model of pre-trained “BERT + LSTM + CNN”, and entity disambiguation uses the method of “pre-trained Chinese word vector embedding + CNN + binary classifier”. Aiming at the problem that entity recognition and entity disambiguation cannot reach the optimal level simultaneously in the actual training, we design to train entity recognition first, then load the weight of the optimal entity recognition model to train the disambiguation model, and finally get the disambiguation result. Fig. 1 shows the overall structure of the financial entity disambiguation model.

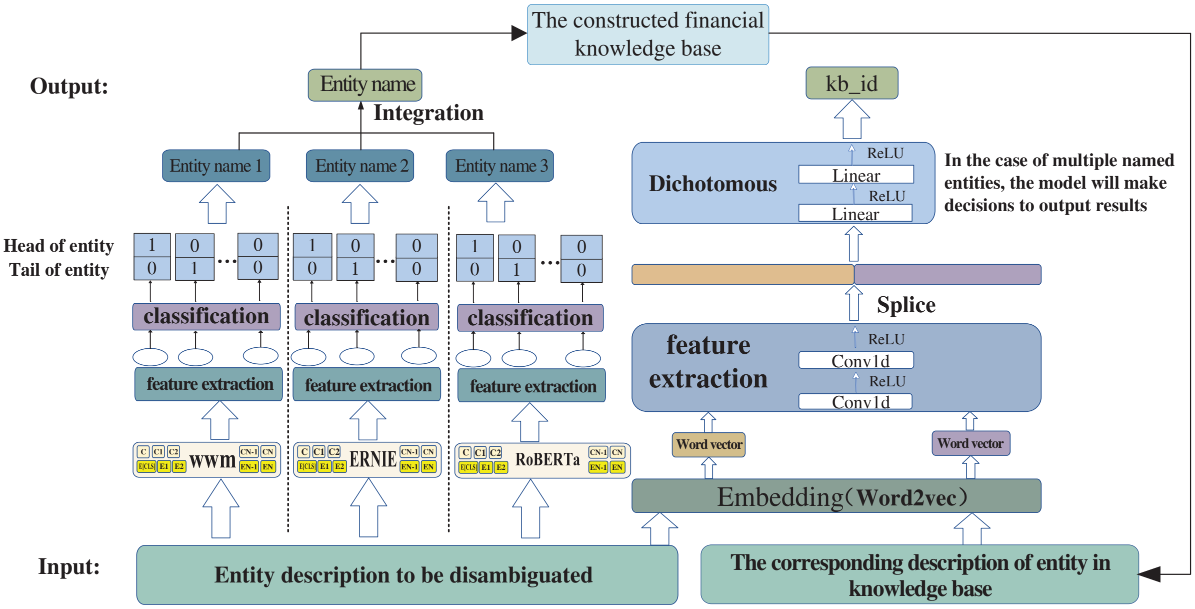

The constructed financial knowledge base
Output:
Entity name
Integration
kb_id
Entity name 1
Entity name 2
Entity name 3
Dichotomous
ReLU
Linear
ReLU
Linear
In the case of multiple named entities, the model will make decisions to output results
Head of entity
Tail of entity
1 0 0 1 ... 0 0
classification
Splice
feature extraction
ReLU
Conv1d
ReLU
Conv1d
C C1 C2 E[CLS] E1 E2 wwm CN-1 CN EN-1 EN
C C1 C2 E[CLS] E1 E2 ERNIE CN-1 CN EN-1 EN
C C1 C2 E[CLS] E1 E2 RoBERTa CN-1 CN EN-1 EN
Word vector
Word vector
Embedding(Word2vec)
Input:
Entity description to be disambiguated
The corresponding description of entity in knowledge base

**Figure 1:** The overall structure of the financial entity disambiguation model

### *3.2 Knowledge Base Construction*

Before recognizing and disambiguating the financial entities, it is necessary to construct the required financial entity knowledge base. Since this paper is mainly aimed at identifying the financial entity in the short text, the text quantity of the short text is small, making it difficulty to understand the context. Therefore, we use the distributed crawler technology to crawl the valuable entity description text from the network text to expand the entity knowledge base. Simultaneously, the text description of all text items corresponding to the entity to be disambiguated is spliced by using multiple text descriptions of the same entity as supplementary descriptions to form a long text description of the knowledge base.

The specific steps are as follows: traversing the complete name set of entity words, first obtain the entity word set to be identified, and then use the distributed crawler to crawl the description information of corresponding entity words from Baidu baike. If there is no description of the entity words in Baidu baike, find all the texts containing the entity word and its *kb_id* is not −1 in the training set train. json, *kb_id* is the number of the disambiguation result corresponding to the entity table and splice them into a text as the entity word's knowledge base sample. In order to facilitate subsequent processing, the dimension of the merged text is controlled within 512 dimensions. If there is no description of the target entity word in Baidu baike and of the word in the training samples, the definition of "full name of xx company is XXX" is directly used as the description text. The process of constructing the knowledge base is shown in Algorithm 1.

**Algorithm 1:** The knowledge base construction

```
Input: Entity name and kb_id of the financial company to be disambiguated
Output: D //D is the knowledge base of text description
for kb_id = 0 to max (kb_id) do
if  CR==−1 //CR is the result of crawling the entity text description of the financial company to be
disambiguated through Baidu baike, and -1 means no result
    then if the entity has disambiguation text whose kb_id! = −1 in train.json
        then D ← splice all the disambiguated texts of the entity as description text //limit the dimension to 512
        else use "The full name of xx company is XXX" to define the description text
    end if
end for
return D
```

This algorithm uses crawler technology to capture the related entity description data on the network for entity nouns starting from the disambiguation training task. The data can be separated from the description in the training set to the greatest extent. Besides, the text description method and extended definition of splicing data set are used for auxiliary construction, ensuring the knowledge base's integrity while making full use of the annotated data.

### *3.3 Named Entity Recognition Based on Multi-Word Vector Integration*

The entity disambiguation task's premise is to identify entity reference items in the text. The accuracy of entity disambiguation will directly affect the accuracy of the subsequent disambiguation. In this paper, BERT improves the model training corpus with three kinds of unsupervised learning. At the same time, it is fine-tuned with LSTM + CNN to construct the entity recognition model. The model is composed of BERT's output splicing classification task and convolution feature extraction. The model structure is shown in Fig. 2.

**The main process of entity recognition.** Firstly, we obtain the vector representation of disambiguation text embedding by pre-trained BERT layer coding. (Along with the model training, BERT also conducts some training on the parameters and modifies the BERT model's parameters and the full connection layer during the fine-tuning phase). After that, the 768 dimensions word vector is input into the LSTM layer to extract features used as CNN's input. After a convolution layer, conv1d is used for further feature extraction and the classification of full connection layer. Finally, the sigmoid activation function is used to limit all positions' output to a real number between 0 and 1(where 1 indicates yes and 0 indicates no), which indicates the probability that the word is the beginning or the end of the entity. In this paper, the threshold value is set to 0.5 to judge each prediction. If it is greater than 0.5, it can be regarded as the entity's head or the tail position. The pseudo-code is shown in Algorithm 2.

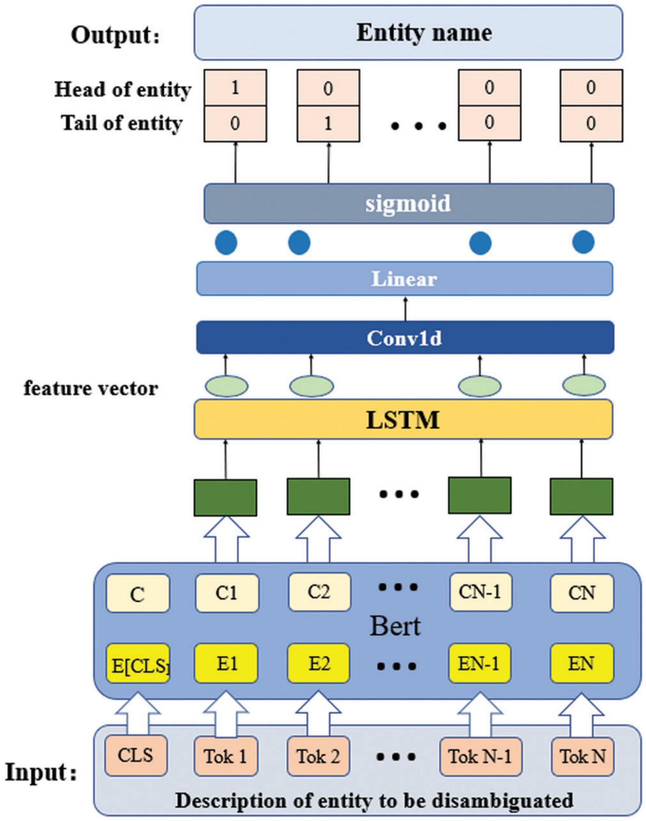


**Figure 2:** The structure of the named entity recognition model

**Algorithm 2:** Named entity recognition

```
Input: $text = \{t_1, t_2, \ldots, t_i\}$ //text is the text to be disambiguated
Output: H and T       //H and T represent the head and tail position of the entity respectively
for i = 1 to length (text) do
    vector ← BERT (text)
    FL← LSTM (vector)       //extract LSTM feature
    FC←CNN(FL)              //extract CNN feature
    result← Classify (FC)   //classify
    number ← Sigmoid (result)
    if number > 0.5
        then output entity head position H
        else output entity tail position T
    end if
end for
return H and T
```

In the whole training process of the model, the Bert model will also do some training on the parameters of some layers along with the training process of the task, and modify the parameters of the Bert model and the full connection layer in the fine-tuning stage. Using the model fine-tuning and the pre-trained words embedded vector to initialize the input parameters, it can achieve faster convergence and better fitting effect and become more suitable for the current task, so that the whole model's results reach the optimal. By fine-tuning the BERT model of Chinese pre-training, this task is transformed into a problem of sentence-relation judgment, which is more in line with the BERT pre-training process.

**Voting integration mechanism.** In this paper, we use three improved pre-trained BERT word vector models-BERT-wwm, RoBERTa-wwm-ext, and ERNIE in entity recognition. Since the same statement often contains multiple different entities in the text, the three models will output different entity recognition results for the same text after the named entity recognition models such as LSTM and CNN. Here we need to put these results to the vote. The model voting follows the principle of "the minority is subordinate to the majority". When there is a disagreement, we adopt the principle of "ERNIE model is the main one and WWM model is the auxiliary one", which means that the output result of most models shall prevail, but when there is a disagreement, the output result of ERNIE model shall prevail, and the other two results shall serve as the reference to output the final entity recognition result. In this way, all named entities in the sentence can be identified to solve multiple named entities' problems.

In this model, the input is word embedding, and the output is the probability of words as the head and end of the recognition answer mention, that is, by using "0/1 annotation" to separate the beginning and end positions of the entity. At the same time, LSTM embeds word features and learns parameter features through many networks. The recognition results output by the model will be used in the next part of the disambiguation task.

### *3.4 Named Entity Disambiguation Based on Word Vector Embedding and CNN Decision*

The entity disambiguation of this algorithm is essentially a binary classification problem. In this paper, the disambiguation method is based on the combination of pre-trained word vector embedding and convolution neural network, loading the word vector pre-trained by word2vec as the embedding layer, training CNN network, and predicting and outputting the results through binary classification. In the entity disambiguation phase, the recognition results used in the entity recognition module's model output are loaded. The structure of the entity disambiguation model is shown in Fig. 3.

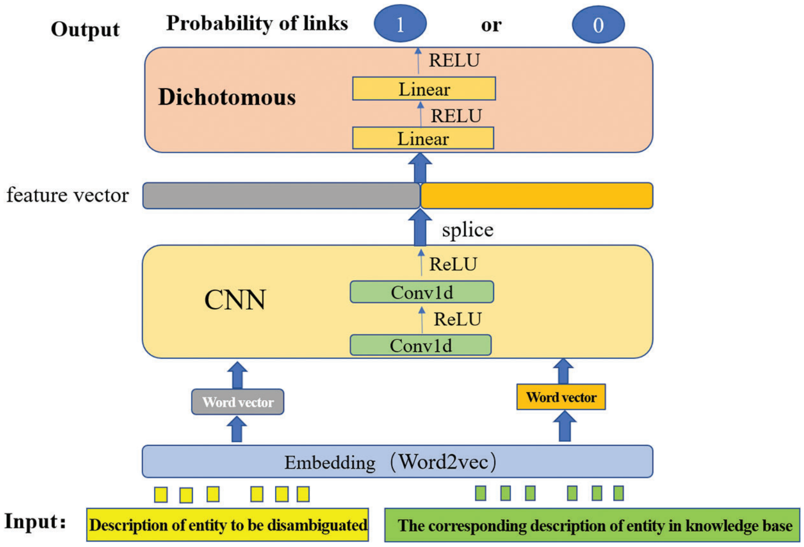


**Figure 3:** The structure of the entity disambiguation model

**The main process of entity disambiguation.** First, load the results identified by the named entity recognition model and extract the corresponding description text from the constructed knowledge base. Embed it with the input query text through the pre-trained 300-dimension Word2vec to respectively get the *Text_seq* and *kb_seq* word vector, and then take them as the input of the CNN layer. After twice convolution conv1d and modified linear unit ReLU, their local features are extracted, and then the text to be disambiguated and the entity corresponding KB description text is paired. After twice fully connected linear and ReLU activation functions, the dimension is reduced to 1. In the case of multiple named entities, the model will decide whether each entity is to be disambiguated while accurately disambiguating. The named entity disambiguation algorithm is shown in Algorithm 3.

**Algorithm 3:** Named entity disambiguation

```
Input: Text = {t_1, t_2, ..., t_i}, kb-text //Text is the description of entity to be disambiguated and kb-text is the corresponding entity description text in knowledge base
Output: kb_id
for i = 1 to length (Text) do
    Text_seq← Word2vec (Text)
    kb_seq← Word2vec (kb-text)
    FCText ← CNN (Text_seq) //extract CNN fearure
    FCkb-text ← CNN (kb_seq) //extract CNN fearure
    FC← Concat (FCText, FCkb-text)
    result← Classify (FC) // classify
    number← Sigmoid(result)
    if number > 0.5
        output kb_id
        else output kb_id = -1
    end if
end for
return kb_id
```

The link probability of the output is a real number between 0 and 1. The output represents the matching degree between the entity description to be disambiguated and the entity description of the knowledge base. If the probability value is greater than 0.5, it will be considered that the link to the knowledge base is successful and output specific *kb_id*. Otherwise, it is regarded as unsuccessful and output $kb\_id = -1$.

### *3.5 Focal Loss Function*

The normal cross-entropy loss function for a positive sample has such a principle that the greater the output probability, the smaller the loss. For negative samples, the smaller the output probability, the smaller the loss. The loss function is relatively slow in the iterative process of many simple samples and may not be optimized to the best. In this case, we use the focal loss function in this paper. It is modified on the basis of the standard cross-entropy loss, and its formula is expressed as:

$$L_{fl} = \begin{cases} -\alpha(1-y')^{\gamma}\log y', & y=1 \\ -(1-\alpha)y'^{\gamma}\log(1-y'), & y=0 \end{cases} \tag{2}$$

The factors $\alpha$ and $\gamma$ are added to the standard cross-entropy loss function. The $\alpha$ balances the uneven ratio of positive and negative samples, and $\gamma$ adjusts the rate of the weight of simple samples decreases.

Among them, $\gamma > 0$ reduces the loss of easy-to-classify samples, and increases the calculation of difficult-to-classify samples.

The focal loss function can solve the problems of imbalance in classification and differences in classification difficulty. It reduces the weight of easy-to-classify samples and makes the model focus more on difficult-to-classify samples during training.

## 4 Experimental Results and Analysis

In this section, we test and report the evaluation results of our approach. On this basis, we also analyze and discuss the influence of essential factors on the proposed method.

### *4.1 Experimental Setting and Dataset*

The experiments of this paper are implemented in the following configuration: Intel(R) Core (TM) i7-7800X CPU @ 3.50GHz, 64.00 GB RAM, and two Nvidia GeForce GTX 1080 Ti GPUs. To evaluate the performance of the named entity recognition and named entity disambiguation model proposed in this paper, we use the data set provided by the Hang Seng Electronics Group. Tab. 1 is a description of the structure of the data set.

**Table 1:** Dataset composition

| Name | Content | Remarks |
|---|---|---|
| company_2_code_sub.txt | Corresponding table of entities to be disambiguated | *kb_id*: Entity number<br>*stock_name*: Company abbreviation<br>*stock_full_name*: Full name of the company<br>*stock_code*: Company code |
| company_2_code_full.txt | Correspondence table of financial sector corporate entity | Company abbreviation, full name of the company, company code |
| train.json | Training set | – |
| dev.json | Validation set | – |
| test_texts.txt | Test data set | – |
| raw_texts.txt | Unlabeled data set (including the set of sentences in the list of entities to be disambiguated) | It can be used to expand and enhance the annotation data |

### *4.2 Experimental Performance Evaluation Indicators*

Three evaluation indicators of accuracy, recall rate, and F1 score are used to evaluate the entity recognition part's performance. Given the text input (expressed by Query), the $N$ entity mentions in $Q$, positions, and their links to the entity id of the knowledge base are manually annotated as follows: $ME_Q = \{(m_1, l_1, e_1), \ldots, (m_k, l_k, e_k)\}$. Accordingly, the output of the model is: $ME_Q = \{(m_1, l_1, e_1), \ldots, (m_k, l_k, e_k)\}$. The calculation formulas of accuracy $P$, recall rate $R$ and $F1$ score are shown in Eqs. 3–5 respectively.

$$P = \frac{\sum_{q\varepsilon Q}\left|ME_q \cap ME'_q\right|}{\sum_{q\varepsilon Q}\left|ME'_q\right|} \tag{3}$$

$$R = \frac{\sum_{q\varepsilon Q}\left|ME_q \cap ME'_q\right|}{\sum_{q\varepsilon Q}\left|ME_q\right|} \tag{4}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{5}$$

### *4.3 Financial Entity Recognition Experiment Results*

In terms of entity recognition, in order to fully verify the recognition effect of the model, use the given *dev* and the data randomly selected in the raw_texts.txt (20% of the whole training set, excluding the data that has been extracted to the training set) as the verification set, named dev_raw. During the training process, the *F1* value of each round in training is shown in Fig. 4. The entity recognition experimental results of the three pre-trained BERT models and their integrated models are shown in Tab. 2.

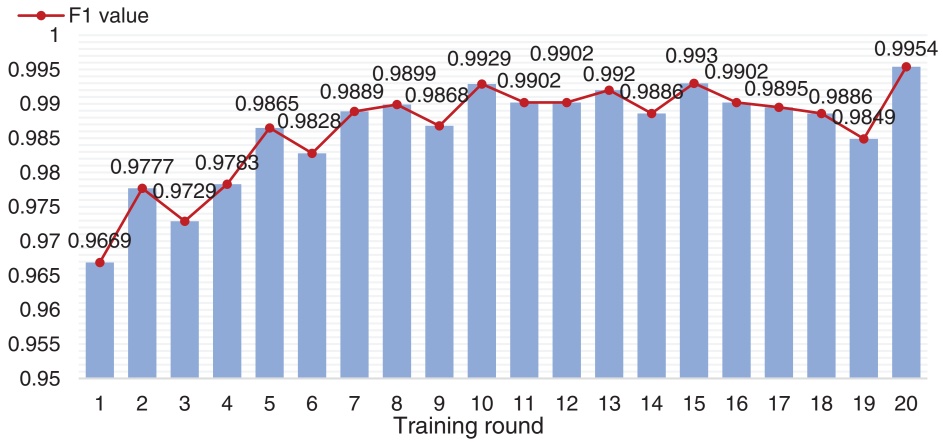


**Figure 4:** *F1* score of named entity recognition in each round of training

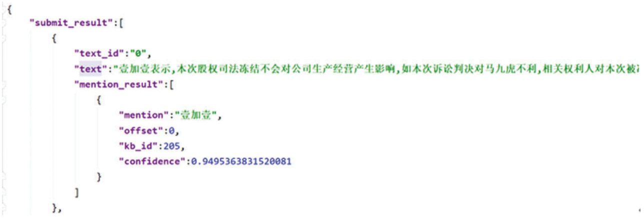
**Table 2:** Comparison of three pre-training BERT models and their integrated entity recognition effect (%)

| Model | dev | | | dev_raw | | |
|---|---|---|---|---|---|---|
| | Precision | Recall | F1 | Precision | Recall | F1 |
| BERT-wwm+LSTM+CNN | 98.96 | 99.20 | 99.08 | 99.31 | 98.26 | 98.78 |
| ERNIE+LSTM+CNN | 99.26 | 99.33 | 99.29 | 99.27 | 98.84 | 99.05 |
| RoBERTa-wwm-ext+LSTM+CNN | 99.57 | 98.47 | 99.01 | 99.17 | 98.47 | 98.68 |
| **Model fuse** | 99.39 | 99.69 | **99.54** | – | – | – |

We can see that the named entity recognition models can show excellent performance through the experimental results when using a single word vector representation embedding. When the *F1* value is all above 99%, the accuracy of the named entity recognition algorithm based on multi-word vector integration can still be improved by 0.25%, which fully demonstrates the method's effectiveness.

### *4.4 Financial Entity Disambiguation Experiment Results*

The result of our entity disambiguation example is shown in Fig. 5. The system can accurately identify the entity name mention, the start position offset in the sentence, the *kb_id* number of the disambiguation results in the entity table, and the confidence score.



```
{
    "submit_result":[
        {
            "text_id":"0",
            "text":"壹加壹表示,本次股权司法冻结不会对公司生产经营产生影响,如本次诉讼判决对马九虎不利,相关权利人对本次被
            "mention_result":[
                {
                    "mention":"壹加壹",
                    "offset":0,
                    "kb_id":205,
                    "confidence":0.9495363831520081
                }
            ]
        },
```

**Figure 5:** Example display of entity disambiguation results

In terms of entity disambiguation experiments, this paper compares and verifies the improvement of Focal Loss function on the model and the accuracy of the three BERT improved models. The experimental results are shown in Tabs. 3 and 4. In the training process, the *F1* value in each training round of the named entity disambiguation algorithm is shown in Fig. 6. The verification set used in the experiments in Tab. 4 is dev.json.

**Table 3:** Comparison of entity disambiguation effect under two loss functions under ERNIE model (%)

| Loss Function | Precision | Recall | *F1* |
|---|---|---|---|
| BCEWithLogitsLoss | 90.42 | 90.71 | 90.57 |
| **Focal Loss** | 91.38 | 91.72 | **91.13** |

**Table 4:** Comparison of three pre-trained BERT models and their integrated entity disambiguation effect (%)

| Model | Loss Function | dev | | |
|---|---|---|---|---|
| | | Precision | Recall | *F1* |
| BERT-wwm | BCEWithLogitsLoss | 90.44 | 89.94 | 90.19 |
| ERNIE | | 90.42 | 90.71 | 90.57 |
| RoBERTa-wwm-ext | | 90.05 | 90.49 | 90.27 |
| **Model Fuse** | **Focal Loss** | 91.38 | 91.66 | **91.52** |

Tab. 3 is the comparison table of the ERNIE model's disambiguation accuracy data under these two loss functions of BCE With Logits Loss and Focal Loss. It can be clearly seen that the *F1* value of the single word vector model with the loss function of Focal Loss rises from 90.57% to 91.13%, indicating that adding the loss function can make the model pay more attention to difficult and misclassified samples, and effectively improve the efficiency of classification.

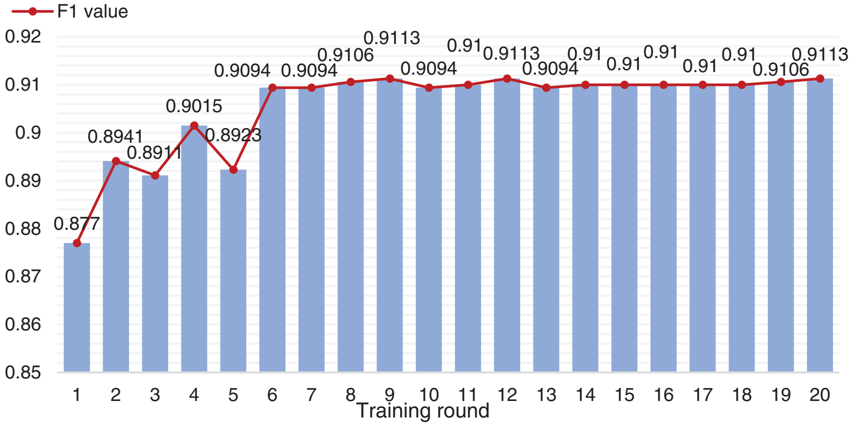


**Figure 6:** *F1* score of named entity disambiguation in each round of training

Here, we compare the single word vector disambiguation model whose loss function is BCE With Logits Loss with our model proposed in this paper: the disambiguation result of the multi-word vector integrated decision disambiguation model whose loss function is Focal Loss. It can be seen intuitively from Tab. 4 that the *F1* value of our model is 0.95% higher than the highest *F1* value of the above three single word vector models. This result further verifies the effectiveness of our algorithm.

## 5 Conclusion

In view of the polysemy problem of Chinese short text in the financial field, we propose a financial short text entity disambiguation algorithm based on multi-word vector integrated decision-making, which integrates a variety of unsupervised pre-trained word vectors and combines feature extraction of LSTM and CNN, improving the accuracy of named entity recognition in Chinese short texts. We transform the entity disambiguation into a binary classification problem and use the fast-training CNN network and replace BERT with Word2vec to ensure the training speed. At the same time, we introduce the focal loss function for the inconsistency of training loss and indicators to improve training efficiency. On the data set provided by the Hundsun Electronics Group, the entity recognition accuracy rate of our algorithm reaches 99.54%, the disambiguation accuracy rate is 91.52%. Experiments show that our algorithm can quickly and accurately locate named entities in financial short texts so as to realize fast and accurate disambiguation.

Considering that the network model that is easy to train and fine-tune is becoming mainstream, we will further explore and design an algorithm to reduce the size of the model under the premise of the accuracy of the model in our future work.

**Acknowledgement:** The author would like to thank the support of Central South University of Forestry & Technology and the support of the National Science Fund of China.

**Funding Statement:** This project is supported by the Degree & Postgraduate Education Reform Project of Hunan Province under Grant 2019JGYB154, the Postgraduate Excellent teaching team Project of Hunan Province under Grant [2019]370-133, the Natural Science Foundation of Hunan Province under Grant 2020JJ4141 and the Natural Science Foundation of Hunan Province under Grant 2020JJ4140.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

[1] Z. T. Duan, F. Li and Z. Chen, "Overview of entity disambiguation," *Control & Decision*, vol. 36, no. 5, pp. 1–15, 2020.

[2] T. Q. Zhou, B. Xiao, Z. P. Cai and M. Xu, "A utility model for photo selection in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 1, pp. 48–62, 2021.

[3] Q. Liu, X. Y. Xiang, J. H. Qin, Y. Tan, J. S. Tan *et al.,* "Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping," *Knowledge-Based Systems*, vol. 192, no. 2, pp. 105375–105389, 2020.

[4] J. H. Qin, H. Li, X. Y. Xiang, Y. Tan, W. Y. Pan *et al.,* "An encrypted image retrieval method based on Harris corner optimization and LSH in cloud computing," *IEEE Access*, vol. 7, no. 1, pp. 24626–24633, 2019.

[5] Z. Yang, S. Zhang, Y. Hu, Z. Hu and Y. Huang, "VAE-Stega: Linguistic steganography based on variational auto-encoder," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 880–895, 2021.

[6] Y. J. Luo, J. H. Qin, X. Y. Xiang and Y. Tan, "Coverless image steganography based on multi-object recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[7] J. H. Qin, J. Wang, Y. Tan, H. J. Huang, X. Y. Xiang *et al.,* "Coverless image steganography based on generative adversarial network," *Mathematics*, vol. 87, no. 9, pp. 1–11, 2020.

[8] Z. Zhou, J. H. Qin, X. Y. Xiang, Y. Tan, Q. Liu *et al.,* "News text topic clustering optimized method based on TF-IDF algorithm on spark," *Computers, Materials & Continua*, vol. 62, no. 1, pp. 217–231, 2020.

[9] Z. D. Wang, J. H. Qin, X. Y. Xiang and Y. Tan, "A privacy-preserving and traitor tracking content-based image retrieval scheme in cloud computing," *Multimedia Systems*, 2021.

[10] J. J. Du, B. Lu and Z. Q. Chen, "A method of named entity disambiguation based on Chinese Wikipedia," *Journal of Hangzhou University of Electronic Science and Technology*, vol. 32, no. 6, pp. 57–60, 2012.

[11] G. Chen and T. X. Xu, "Research on the discovery of entity relationships in subdivided domains under the guidance of a small-scale knowledge base," *Journal of Intelligence*, vol. 38, no. 11, pp. 1200–1211, 2019.

[12] X. Y. Peng, "Design and implementation of named entity recognition algorithm for financial field," Wuhan City, Hubei Province, China: M.S. thesis, Huazhong University of Science and Technology, 2019.

[13] Z. Fang, Y. Cao and Q. Li, "Joint entity linking with deep reinforcement learning," in *Proc of the World Wide Web Conference*, New York, NY, USA, pp. 438–447, 2019.

[14] M. Xue, W. Cai and J. Su, "Neural collective entity linking based on recurrent random walk network learning," in *Proc of the 28th Int. Joint Conf. on Artificial Intelligence*, Freiburg, Germany, pp. 5327–5333, 2019.

[15] M. Francis-Landau, G. Durrett and D. Klein, "Capturing semantic similarity for entity linking with convolutional neural networks," in *Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, France, pp. 12561261, 2016.

[16] S. D. Chen and X. Y. Ouyang, "Overview of named entity recognition technology," *Radiocommunication Technology*, vol. 46, no. 3, pp. 251–260, 2020.

[17] M. Kim, J. Kim and M. Shin, "Word embedding based knowledge representation with extracting relationship between scientific terminologies," *Intelligent Automation & Soft Computing*, vol. 26, no. 1, pp. 141–147, 2020.

[18] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. of the Int. Conf. on Learning Representations (ICLR 2013)*, Scottsdale, Arizona, 2013.

[19] X. P. Qiu, T. X. Sun, Y. G. Xu, Y. F. Shao, N. Dai *et al.,* "Pre-trained models for natural language processing: A survey," *Science China (Technological Sciences)*, vol. 63, no. 10, pp. 1872–1897, 2020.

[20] M. D. Andrew and V. L. Quoc, "Semi-supervised sequence learning," in *Proc. of the 28th Int. Conf. on Neural Information Processing Systems (NIPS'15)*, Cambridge, MA, USA, pp. 3079–3087, 2015.

[21] J. H. Qin, W. Y. Pan, X. Y. Xiang, Y. Tan and G. M. Hou, "A biological image classification method based on improved CNN," *Ecological Informatics*, vol. 58, no. 4, pp. 101093, 2020.

[22] J. Devlin, M. W. Chang and K. Lee, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, pp. 4171–4186, 2018.

[23] Y. M. Cui, W. X. Che and T. Liu, "Pre-training with whole word masking for Chinese BERT," [Online]. Available: arXiv preprint arXiv:1906.08101, 2019.

[24] Z. Y. Zhang, X. Han, Z. Y. Liu, X. Jiang, M. S. Sun *et al.,* "ERNIE: Enhanced language representation with informative entities," in *Proc. of the 57th Association for Computational Linguistics (ACL)*, Stroudsburg, PA, USA, pp. 1441–1451, 2019.

[25] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1746–1751, 2014.

[26] L. Y. Xiang, S. H. Yang, Y. H. Liu, Q. Li and C. Z. Zhu, "Novel linguistic steganography based on character-level text generation," *Mathematics*, vol. 8, no. 9, pp. 1558, 2020.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] P. M. Sosa, "Twitter sentiment analysis using combined LSTM-CNN models," 2017. [Online]. Available: http://konukoii.com/blog/2018/02/19/twitter-sentiment-analysis-using-combined-lstm-cnn-models/.

[29] T. Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár *et al.,* "Focal loss for dense object detection," in *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, Italy, pp. 2999–3007, 2019.