applied sciences

MDPI

Article

RSVN: A RoBERTa Sentence Vector Normalization Scheme for Short Texts to Extract Semantic Information

Lei Gao [1], Lijuan Zhang [1,*], Lei Zhang [1,2] and Jie Huang [1,*]

[1] School of Information and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2] School of Biological and Chemical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
* Correspondence: 121107@zust.edu.cn (L.Z.); huangjie@zust.edu.cn (J.H.)

Abstract: With the explosive growth in short texts on the Web and an increasing number of Web corpora consisting of short texts, short texts are playing an important role in various Web applications. Entity linking is a crucial task in knowledge graphs and a key technology in the field of short texts that affects the accuracy of many downstream tasks in natural language processing. However, compared to long texts, the entity-linking task of Chinese short text is a challenging problem due to the serious colloquialism and insufficient contexts. Moreover, existing methods for entity linking in Chinese short text underutilize semantic information and ignore the interaction between label information and the original short text. In this paper, we propose a RoBERTa sentence vector normalization scheme for short texts to fully extract the semantic information. Firstly, the proposed model utilizes RoBERTa to fully capture contextual semantic information. Secondly, the anisotropy of RoBERTa's output sentence vectors is revised by utilizing the standard Gaussian of flow model, which enables the sentence vectors to more precisely characterize the semantics. In addition, the interaction between label embedding and text embedding is employed to improve the NIL entity classification. Experimental results demonstrate that the proposed model outperforms existing research results and mainstream deep learning methods for entity linking in two Chinese short text datasets.

Keywords: entity linking; Chinese short text; flow model; label embedding

Citation: Gao, L.; Zhang, L.; Zhang, L.; Huang, J. RSVN: A RoBERTa Sentence Vector Normalization Scheme for Short Texts to Extract Semantic Information. *Appl. Sci.* **2022**, *12*, 11278. https://doi.org/10.3390/app122111278

Academic Editor: Elisa Quintarelli

Received: 15 September 2022
Accepted: 5 November 2022
Published: 7 November 2022

1. Introduction

With the rapid development of the Internet, short text in the Web increased rapidly. These short texts are the main constituent forms of many Web applications, such as search queries, news titles, and social media comments [1]. Therefore, short texts have become a prominent part of natural language processing. However, short texts contain many mentions whose meanings are often ambiguous and cannot be effectively linked to the knowledge base. Fortunately, entity linking as a critical step in many natural language processing tasks can solve this problem. The entity-linking task aims at linking a given mention in a piece of text to the correct entity in a specific knowledge base [2]. Hence, entity linking for short texts plays an increasingly important role in a variety of industrial applications.

This paper focuses on entity linking for short texts, which usually consist of only a few or a few tens of terms, while long texts typically consist of a single document. Traditional entity linking are primarily for long texts [3–5], which contain rich contextual information and background knowledge that can facilitate disambiguation. Nevertheless, there are fewer studies on entity linking for short texts because short texts have the following challenges compared to long texts. To begin, short texts have insufficient contextual information and sparse semantic expression that are hardly distinguishable by contextual semantics [6]. Next, the short texts contain some informal expressions, which increase the semantic ambiguity of the short texts. Therefore, entity linking in short texts can be more complicated due to the above challenges. Recently, research entity linking for Chinese

short text has gradually increased. The obvious difference between Chinese and English is the lack of space separators in Chinese sentences, so it can be more challenging to understand the contextual semantics in Chinese than in English. Moreover, the current entity linking for Chinese short-text methods does not consider the anisotropy problem of the sentence vector output by the pre-trained language model. This anisotropy refers to the fact that word embeddings occupy a narrow cone in the vector space, which leads to semantic information not being further fully utilized. In addition, the current entity linking for Chinese short-text methods ignore the interaction between labels and original texts when classifying unlinkable entities. Owing to these challenges and problems, traditional entity-linking methods cannot handle Chinese short text well, and how to effectively represent semantics and make full use of the interaction between labels and texts remains a challenging problem for entity linking in Chinese short text.

To more effectively address the challenge of Chinese short text in entity linking, we propose an entity-linking model based on RoBERTa sentence vector normalization with label embedding to fully utilize semantic information. First, context semantic information is adequately captured in sentence vectors by the RoBERTa model. In the next step, to better characterize the semantics, the sentence vectors are input to a post-processing model to correct the anisotropy. Specifically, the RoBERTa output sentence vectors are fed into a flow model and the standard Gaussian distribution is utilized to transform the sentence vectors into a smooth, isotropic spatial distribution to better describe the semantics. Furthermore, entity typing is applied to unlinkable mentions to enhance the performance of entity linking, and the interaction between label embedding and text embedding is further adapted to enhance the multi-classification of entity types.

The main contributions of this paper are detailed below:

(1) We design a model based on sentence vector normalization and label embedding, which can sufficiently capture the semantic information in Chinese short texts and effectively utilize the original label information.

(2) The flow model is applied to our proposed model to eliminate the anisotropy of sentence vectors, which enables sentence vectors to better characterize the semantics. Additionally, our proposed model employs label embedding to enhance the interaction between text and labels.

(3) Experimental results demonstrate that our proposed model achieves superior performance on entity linking for the CCKS2019 and CCKS2020 Chinese short-text datasets.

The remainder of this paper is structured as follows: Section 2 reviews the related work of entity linking. The various modules of our proposed model are presented in Section 3. In Section 4, our proposed model was evaluated and compared across two Chinese short-text datasets. Finally, Section 5 summarizes this paper and provides an outlook for future research directions.

2. Related Work

With the continuous development of natural language processing, entity linking has become a popular research topic in industry and academia. Many researchers have proposed different entity-linking methods from different perspectives. The main methods of entity linking can be classified into two categories: traditional characterization methods and deep learning-based methods.

2.1. Traditional Characterization Methods

Traditional characterization models in entity linking aim to encode sentences into computable vector forms, and earlier utilized the bag-of-words model to characterize context. The bag-of-words model is utilized by many researchers [7–9] to represent the context of mention and the descriptive text of candidate entities in vector form, and then calculated the similarity between the candidate entities and the mentions using the cosine similarity. However, the bag-of-words model usually adopts one-hot encoding that is extremely sparse and ignores syntax and word order, which leads to poor semantic repre-

sentation. Later, Yu et al. [10] utilize Latent Semantic Analysis (LSA) [11] to characterize documents, which utilizes Singular Value Decomposition (SVD) to perform dimensionality reduction to identify potential semantic relationships between texts. The LSA method can remove the irrelevant information from the original vector space and make the semantic relationship more explicit; however, the computational complexity of the SVD solution is large. Subsequently, Mikolov et al. [12] proposed the Word2Vec method to represent words as dense vectors, which had the advantage of making the vectors between semantically similar words closer and allowed the operation of vectors to reflect the relations between certain words. In addition, the Word2Vec method is optimized efficiently and simpler to implement compared to the LSA method. Later researchers further combine Word2Vec with other methods to enhance the performance of entity linking. Wang et al. [13] proposed an entity-linking method combining word vector and graph model, which employed Word2Vec to construct word vectors and adopted a random walk algorithm to calculate the similarity of keywords. Additionally, the classic Word2Vec model was modified by Ganea et al. [14] based on extensive entity word co-occurrence data, which improved the semantic representation of the word vector model and enhanced the effectiveness of subsequent links. Nevertheless, the Word2Vec model does not factor in context, and each word in a vocabulary exists in one place within the vector space. Jeffrey et al. [15] proposed the GloVe model, which integrates the advantages of Word2Vec and LSA models and can capture global corpus. Furthermore, Bryan et al. [16] proposed the CoVe model that can consider contextual semantic information, which is based on the GloVe model and a machine translation model to obtain contextual word embedding. Summarizing, many proposed word-embedding techniques have not considered contextual semantic information until CoVe was proposed; however, this method still primarily utilizes the model performance of downstream tasks.

2.2. Deep Learning-Based Methods

With the development of deep learning technology, it can capture features layer by layer of network [17] to improve the accuracy of learning. Deep learning techniques with superior performance have been frequently applied in various natural language processing work, and show significant advantages in entity-linking tasks. He et al. [18] proposed an entity-linking method based on deep neural networks, which utilized deep neural networks to learn entity and context representation and achieved superior performance without manually designed features. Since graph convolutional networks make entity association modeling easier, several models [19–21] employed a graph convolutional network to extract contextual features to improve the performance of entity linking. Moreover, the neural network needs further optimization for the semantic sparsity problem of short texts. Since LSTM (Long Short-Term Memory) has shown advantages in image recognition and other fields [22], Zeng et al. [23] proposed a short-text entity-linking method based on a double-attention-based LSTM network. However, the LSTM modeling could not capture both past and future contexts and had a large number of parameters. Then, a symmetric Bi-LSTM model with a dual attention mechanism was proposed by Hu et al. [24], in which Bi-LSTM could better capture bi-directional semantic dependence, and their proposed model combined structural information with the attention process to more completely extract semantic properties of entities. Furthermore, Matthew et al. [25] proposed the ELMo model to extract the deep contextual representation, which captures semantic features of words and sentences using a multi-layer BiLSTM model. Additionally, Onoe et al. [26] adopted the ELMo model and attention mechanism to obtain the mention embedding.

The abovementioned deep learning algorithms have achieved satisfactory results in entity linking. However, as some pre-trained models based on transformer architecture have outperformed previous deep learning models on major tasks, many researchers have turned their attention to transformer-based models. Firstly, Devlin et al. [27] presented the BERT model, which demonstrated excellent performance across a number of natural language processing tasks and was widely employed in entity linking. Then, in the entity-

linking task, Chen et al. [28] exploited BERT-based entity embedding to better capture the type information of entities, which was applied to modeling contextual information of entities in knowledge bases. Recently, transformer-based models are also applied for entity linking in Chinese short texts. Cheng et al. [29] developed the BERT-ENE model based on BERT and treated the short textual entity-linking problem to a binary classification task. However, they ignored the assistance of keywords for entity linking in the context of candidate entities. Consequently, Zhan et al. [30] employed the TextRank keyword extraction technique in BERT to enhance the effectiveness of Chinese short-text entity linking. Nevertheless, these methods only utilized BERT models to extract semantic features that were not combined with other models to further explore the correlation between entities. Zhao et al. [31] proposed an approach to integrate BERT with a graph model, in which the graph model can capture the association between candidate entities with different mentions. In addition, Jiang et al. [32] applied the features extracted by BERT through a dual-channel network to further extract semantic information. However, excessive model superposition increases the computational complexity and sacrifices more computing resources while improving efficiency.

Further enhancements to the performance and application of transformer-based models have been proposed subsequently. First, Liu et al. [33] proposed a robust and optimized version of the RoBERTa model based on BERT, which outperformed BERT in multiple problems involving natural language processing. Secondly, Riemers et al. [34] found that sentence vectors generated by the transformer structure system could not be directly matched for semantic similarity. Subsequently, Li et al. [35] discovered that the BERT output sentence vector is anisotropic and adopted a flow model to modify the spatial representation of the sentence vector, so that the sentence vector can better represent the semantic information. In addition, some scholars have concentrated primarily on class representations to benefit text-based classification applications. Xiong et al. [36] developed a method to enhance the interaction between label and text to enhance the accuracy of BERT text classification. Inspired by the above research, a model based on RoBERTa sentence vector normalization and label embedding is proposed. The proposed model firstly employs the RoBERTa model to fully extract the text features and then feeds the sentence vectors output from RoBERTa into the flow model for post-processing to better represent the semantics of the sentence vectors. Additionally, the proposed model adopts label-embedding technique to improve the performance of NIL entity classification.

3. Our Proposed Entity-Linking Method

The purpose of the entity-linking task is to link entity mentions to the corresponding entities of the knowledge base. In our method, the linkable mentions are precisely linked to the knowledge base and the unlinkable mentions are further given a superordinate type. A description of our entity-linking method is given in the following four subsections. Firstly, the set of mentioned candidate entities is generated by utilizing the knowledge base to narrow the scope of subsequent links. Then, each module of our proposed model is introduced individually. Afterward, the entity-linking model is performed for linkable mentions, and the entity-typing model is further applied to unlinkable mentions. Finally, an algorithm is presented to describe the overall process of our proposed method. Details about our approach are presented in the following sections.

3.1. Candidate Entity Generation

Because of the large number of entities in the knowledge base, it is impossible to compare the mentions with each entity in the knowledge base, so a candidate search process is required to reduce the scope of subsequent links. This selection process is called candidate entity generation and aims to initially search for the set of entities related to the mentions. Therefore, entity linking first retrieves the set of candidate entities related to mentions from the knowledge base through candidate entity generation to reduce the link scope of subsequent models. There are two main methods for generating candidate enti-

ties: dictionary-based methods and search engine-based methods. The dictionary-based approach is employed by most entity-linking systems [37–39], which generates candidate entity sets by constructing a dictionary of mapping relationships for all possible linked entities. The search engine-based approach utilizes information on the web to generate candidate entities [40–42], which inputs the reference name into search engines and selects the top ranked pages to join the candidate set. To this end, according to the characteristics of the candidate entity generation method and the data sets utilized in this paper, the dictionary-based method is selected to generate the candidate entity set. The specific process of candidate entity generation in this paper is to match the mentions in the sentence with the keys in the dictionary that is generated and based on the knowledge base. If the keys of the dictionary meet the exact matching requirements, all the values corresponding to the keys will be added to the candidate entity set. To better demonstrate the process of candidate entity generation, Figure 1 illustrates an example of generating candidate entities sets. First, the mentions in the short text are precisely matched with the entities in the knowledge base, and then the matched entities are added to the set of candidate entities.

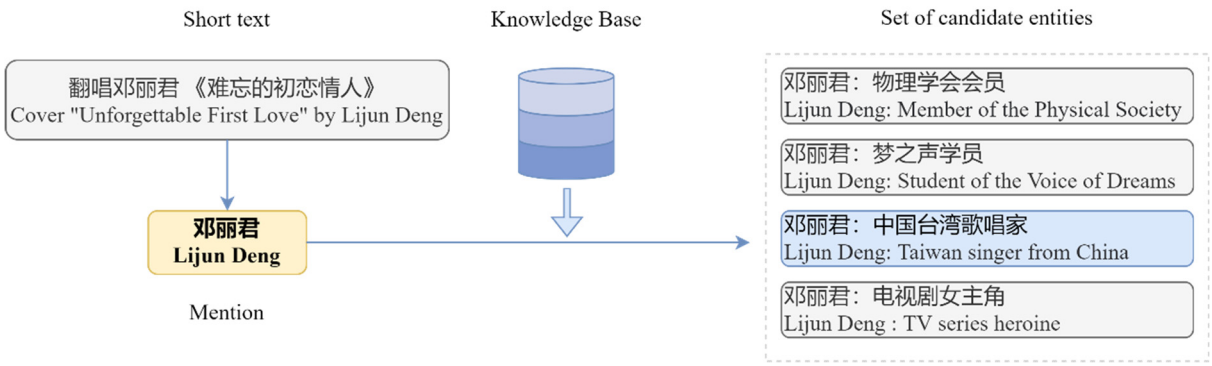

Figure 1. Example of generating a candidate entity set.

3.2. Modules of Our Proposed Model

In this part, the modules of our proposed method are described in detail. Firstly, the context semantic features are extracted from the sentence vector by RoBERTa. Then, the standard Gaussian function of the flow model is employed to correct the anisotropy of the sentence vector, which enables full representation of the semantic information in the sentence vectors. Finally, the label-embedding technique is adapted to further improve the classification performance of unlinkable entities.

3.2.1. Vector Representation of Context

BERT is a bi-directional language model built by stacking transformer [43] partial structures. Moreover, the training of BERT is different from the previous language representation models by combining the Masked Language Model (MLM) and Next Sentence Prediction (NSP). Specifically, the MLM task randomly selects 15% of the words in the input sequence to replace with (MASK), and then the context is exploited to predict these masked words. The NSP task is mainly employed to judge whether two sentences are consecutive.

The RoBERTa model inherits the advantages of the BERT model, which also adopts the encoder of a bi-directional transformer as the middle layer to extract information. In addition, the RoBERTa model provides better performance compared to the BERT model by improving the masking method and removing the NSP task. In the masking strategy, the BERT model utilizes a static mask that keeps the masking token unchanged. In comparison, RoBERTa adopts a dynamic mask where the mask position continuously changes during the training process, and this change enhances the randomness of the data and thus increases the learning capability for the model. After removing the NSP task, RoBERTa adopts consecutive full sentences and doc sentences as input and increases the length of in-

put sentences to 512 characters, which is much longer than the maximum 256 characters for the BERT model. In summary, the performance of the RoBERTa model is more prominent than BERT. Furthermore, some researchers [44] improve the single-character mask of original BERT and propose a whole-word mask scheme, and by combining the whole-word mask scheme with the RoBERTa model, a RoBERTa-wwm-ext model is further proposed. This mask scheme fully considers the traditional Chinese word segmentation operation in natural language processing and carries out the mask operation on the word granularity, which helps to capture the semantic features of the Chinese word-level and further enhances the performance of the transformer-based model on the Chinese dataset. To clearly demonstrate the difference between the whole-word mask and the single-character mask, Table 1 gives an example to compare the original text with single-word masks and full-word masks.

Table 1. Examples of different masks.

Illustration	Sample
Original text	水圬多的水就是 硬水 吗? Does water with a lot of masonry mean **hard water?**
Single character mask	水圬多的水就是**(MASK)** 水 吗? Does water with a lot of masonry mean **(MASK) water**?
Whole word mask	水圬多的水就是**(MASK) (MASK)** 吗? Does water with a lot of masonry mean **(MASK) (MASK)**?

In this paper, the RoBERTa-wwm-ext model is utilized to extract contextual semantic features, which combines the advantages of the full word masking technique and the RoBERTa model. Figure 2 displays the structure diagram for the RoBERTa-wwm-ext model, where all the words in the sentence are transformed into low-dimensional vectors, and then the text semantic information is captured by the 12-layer transformer, and finally, the trained sentence vectors and character vectors are outputted by the RoBERTa-wwm-ext model. Specifically, for the input content, the embedding representation of the text is the combination of token embedding, segment embedding, and position embedding; the equation for generating text embedding is shown as follows:

$$\mathbf{E}_t = \mathbf{E}_{token\_emb} + \mathbf{E}_{seg\_emb} + \mathbf{E}_{pos\_emb}, \tag{1}$$

where $\mathbf{E}_t$ represents the embedding representation of t-th character, $\mathbf{E}_{token\_emb}$ represents the token embedding of the character, $\mathbf{E}_{seg\_emb}$ represents the segment embedding of the character, and $\mathbf{E}_{pos\_emb}$ denotes the position embedding of the character.

Next, the embedding representation is fed into the middle layer of the RoBERTa model, which comprises an encoder in the 12-layer transformer to extract semantic information. The most critical module in the transformer is the multi-head attention mechanism, which consists of multiple self-attention layers. The corresponding output of a single self-attention mechanism is calculated by the dot product of attention, and multiple self-attention is obtained by concatenating all heads. A linear transformation is then applied to obtain the calculation for multi-head attention. Finally, self-attention and multi-head attention can be obtained by the following equations:

$$\mathbf{head}_i = Attention\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right), \tag{2}$$

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(\mathbf{head}_1, \mathbf{head}_2, \ldots, \mathbf{head}_n)\mathbf{W}^O, \tag{3}$$

where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$, and $\mathbf{W}^O$ are the weight matrices of the attention mechanism, and $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are the input vectors of the attention mechanism.

Finally, the sentence vector (CLS) in the output layer of the RoBERTa model contains global semantic and contextual information, which can be further utilized to determine whether the short text is the same semantic context as the description text of the candidate entity.

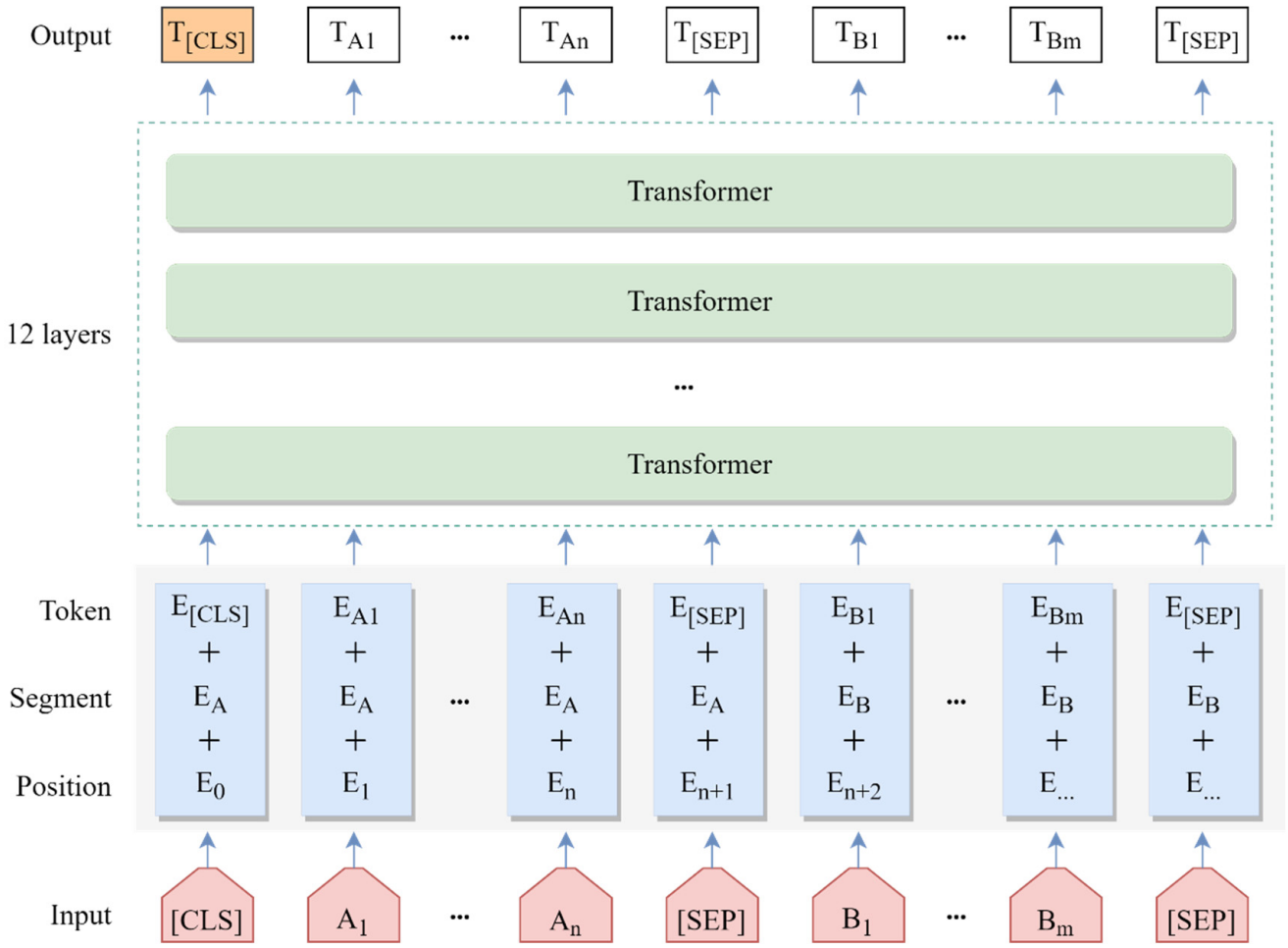

Figure 2. The model of RoBERTa.

3.2.2. Sentence Vector Normalization

The representation space of word embedding in the transformer-based model has an anisotropic conical spatial distribution [45–47], which limits the representational power of word embedding and thus leads to the underutilization of semantic information. In addition, in anisotropic representation space, the high-frequency words tend to concentrate tightly whereas the low-frequency words tend to disperse sporadically. The sparse distribution makes some spatial semantic expressions unclear and the unbalanced distribution between the high-frequency words and low-frequency words will mislead the space of contextual embedding. Moreover, the representation space of the sentence vector outputted from the pre-trained model based on the transformer also suffers from similar issues, which implies that there is also anisotropy in the sentence vector space extracted by the transformer-based model. Therefore, directly using the encoding based on the transformer model as context embedding cannot acquire sufficient semantic representation. To address the problem of anisotropy of sentence vectors output by the transformer-based model, non-smooth anisotropic sentence embedding can be transformed into an isotropic standard Gaussian distribution by utilizing the flow model [35], which enhances the semantic expression capacity of sentence vectors. Inspired by the above method, a layer of flow model is adopted in our proposed model, which improves the semantic representa-

tion of the sentence vectors by correcting the anisotropy problem. Specifically, the flow model is further employed to correct the representation space of the extracted sentence vectors after RoBERTa to improve the subsequent computational performance. The transformation of the flow model maximizes the possibility of generating RoBERTa sentence vectors from standard Gaussian latent variables and is able to correct the sentence vector space by utilizing standard Gaussian latent spaces. The equation of the flow model is described as follows:

$$\begin{aligned} \max_{\phi} \mathbb{E}_{\mathbf{u}} = \quad & \text{RoBERTa}(\text{sentence}), \text{sentence} \sim \mathcal{D} \\ & \log p_{\circledcirc}\left(f_{\phi}^{-1}(\mathbf{u})\right) + \log \left| \det \frac{\partial f_{\phi}^{-1}(\mathbf{u})}{\partial \mathbf{u}} \right|, \end{aligned} \tag{4}$$

where $\mathcal{D}$ denotes the dataset of sentences set, f_{ϕ} is an invertible mapping function that has been carefully designed to ensure the invertibility [48], $p_{\circledcirc}$ is a standard Gaussian distribution, the observed space $\mathbf{u}$ is the sentence vector extracted from RoBERTa, det is the determinant, ∂ is the partial derivative, and log is a function of logarithms.

3.2.3. Fusion Label Embedding

The thought of label embedding is to concatenate the original category text information and the short text to the pre-trained language model as input. Specifically, the interaction between the text embedding and the label embedding learns the relationship between the mention and the labels, and further enhances the performance of classification.

The structure of the RoBERTa with enhanced label embedding is illustrated in Figure 3; the original text is combined together with the labels in the pre-trained language model, and these two parts are embedded in different segments. In training the RoBERTa model, the original attention mechanism of the RoBERTa model is employed to implement the interaction between label embedding and text embedding. However, these categories are extremely abstract to the original text without the addition of label embedding: for example, for the following four categories: people, food, sports, architecture, and a short text "apples in Shanxi", where "apples" is the referent item to be classified. In the case of the model without label embedding, these four categories are all equivalent for the text "apples in Shanxi". When label embedding is introduced to the model, the input of RoBERTa becomes "people, food, sports, architecture + apples in Shanxi", which assists the model to understand the relationship between the text "apples" and "food". The interaction between labels and text can learn additional text-level relationships and enhances the performance of the classification task. Therefore, label embedding is employed in our model to further enhance the multi-classification performance of NIL entities.

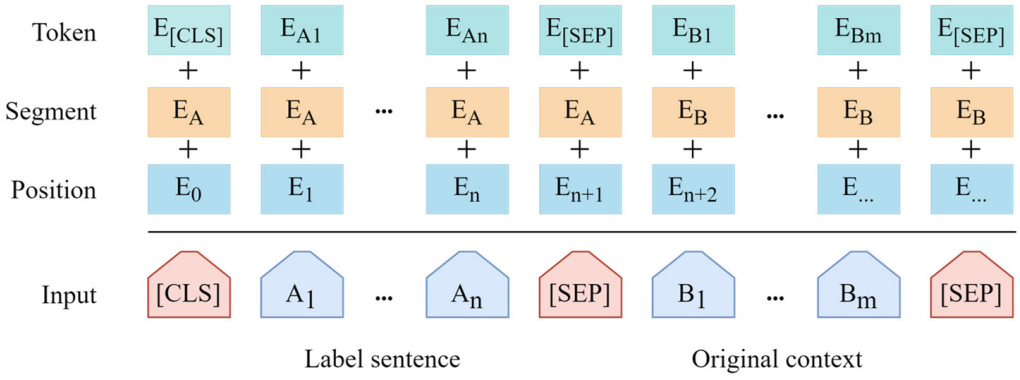

Figure 3. The structure of label embedding.

As shown in Figure 3, the input contains the texts of labels and the original short text separated by a (SEP) token; n is the total class of all labels and m is the total number of characters in the short text, $A_1, \ldots, A_n$ denotes the label token and $B_1, \ldots, B_m$ represents the original short-text token. The subsequent embedding representation and training process is the same as the original RoBERTa.

3.3. Entity Linking

The linkable entities are processed through the entity-linking model. As presented in Figure 4, the mention context and description information of the candidate entity are entered into the entity-linking model to obtain the relevance score. The proposed model of entity linking is roughly divided into three modules. Firstly, the RoBERTa module is employed to extract contextual features, then the flow module is utilized to further correct the contextual representation, and finally the fully connected layer is adapted to generate the output results. In specific, the mention context and description content of the candidate entity are combined with a special token and entered as input to RoBERTa. Next, the sentence vector [CLS] with extracted contextual features is fed to the flow model for post-processing. Later, the flow model outputs uniformly distributed and isotropic sentence vectors and feeds them into the fully connected layer, where the fully connected layer utilizes a sigmoid activation function. Finally, binary classification is performed for each pair of samples to output the prediction results.

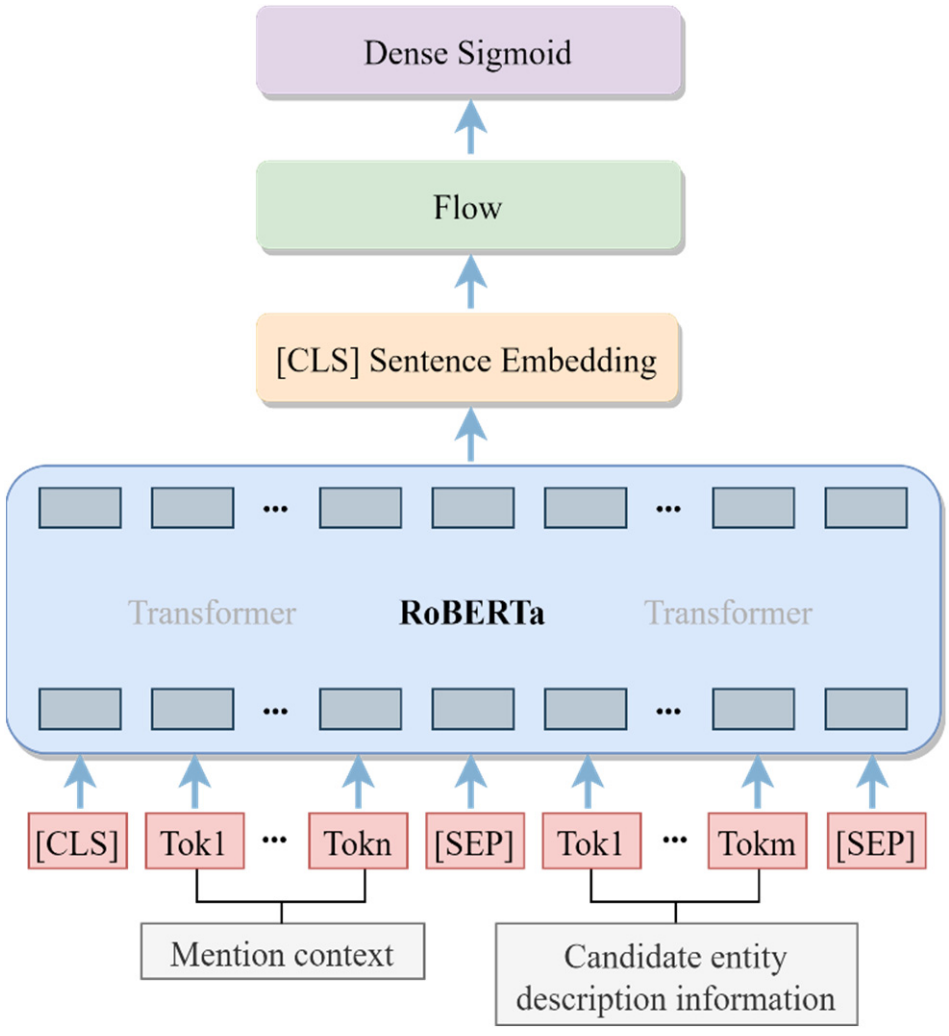

Figure 4. Entity-linking model.

3.4. Entity Typing

The entity-typing model is adopted for further classification of unlinkable entities, and superordinate type of the mention is obtained by entering the mention context and label sentence in the entity-typing model. As described in Figure 5, the model structure is likewise separated into three parts: RoBERTa layer, flow layer, and fully connected layers. To be specific, the label text and the mention context are linked by specific token as the input for RoBERTa, where the label text is composed of all category texts concatenated together. Next, the intermediate model is equivalent to the entity linking, which is to extract semantic features from the RoBERTa model and correct the anisotropy with the flow model. Finally, a softmax activation function is employed in the fully connected layer for each sample to generate prediction categories by multiple classifications.

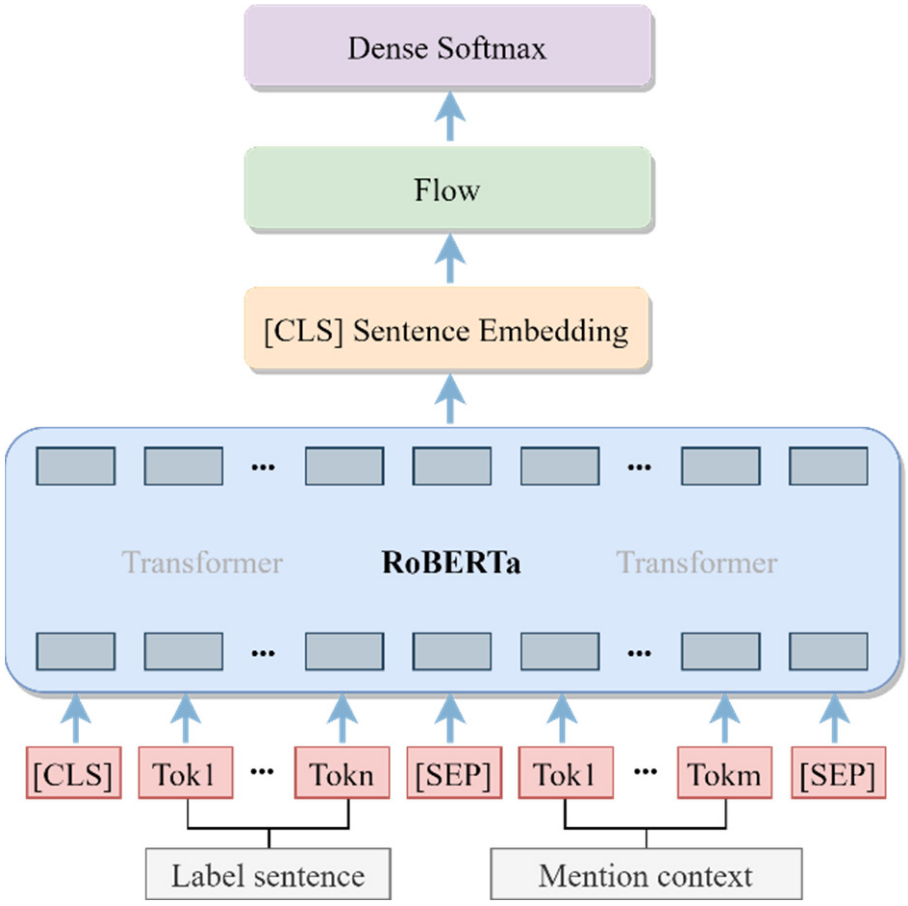

Figure 5. Entity-typing model.

3.5. Algorithm Description

Our proposed approach obtains the set of candidate entities and types from the knowledge base to link the mentions from the short text, and then the entity-linking model and entity-typing model are trained in sequence. In the final result of prediction, the candidate entity with the highest score and exceeding the threshold is regarded as the result of the mentioned link; otherwise, the mention is considered as a NIL entity and further assigned to the superordinate type. Finally, the entire process of our model is described in detail by Algorithm 1.

Algorithm 1 Our proposed model for entity linking of Chinese short text

Input: Chinese short text with mentions marked, knowledge base
Output: Linked entity *kb_id* or entity type *et*

Begin
Define *el_result* to preserve the relevance score of the entity linking model, *et_result* to store the superordinate type of the entity typing model, and K as the threshold for determining the relevance of the link;
Generate entity linking set $\mathcal{L}$ and and entity typing set $\mathcal{T}$ of Chinese short text with mentions marked in the knowledge base by the method of exact match;
for $i = 0$ *to* $|\mathcal{L}|$ **do**
    Obtain the sentence vector representation **V** of text and candidate entity information from RoBERTa;
    Transform the **V** into a uniformly distributed vector **F** by Equation (4);
    Use a fully connected layer combined with sigmoid function to predict the correlation of **F** and store the correlation score in *el_result*;
end for
for $i = 0$ *to* $|\mathcal{T}|$ **do**
    Obtain the sentence vector representation **V** of text and label information from RoBERTa;
    Transform the **V** into a uniformly distributed vector **F** by Equation (4);
    Use a fully connected layer combined with softmax function to predict the entity type of **F** and store the type result in *et_result*;
end for
for $i = 0$ *to* $|el\_result|$ **do**
    if $i \geq K$ **then**
        Output the linked entity *kb_id* with the highest score from *el_result*;
    else
        Output the prediction type *et* from *et_result*;
    end if
end for
End

4. Performance Analysis

In this part, the experimental procedure of our model is introduced in detail from three aspects. Firstly, it describes two entity-linking datasets of Chinese short text utilized for the experiments. Next, a detailed table and formula are utilized to describe the experimental setup and evaluation parameters. Finally, the superiority of our proposed approach is demonstrated by multiple sets of experiments.

4.1. Dataset

The experiment utilizes the annotated data and knowledge base provided by CCKS2020 and CCKS2019 entity linking for Chinese short text. To be specific, the annotation data are derived from real Internet page titles, video titles, and search queries, and the knowledge base is derived from the knowledge base of Baidu Baike. Table 2 shows examples of annotated data, and each entry contains the original short text and structured information of the mentions. Each knowledge base entry includes an entity, entity alias, entity type, and a set of attribute information associated with the entity. In experiments on both datasets, the proportion of training set to test set is 7:1. Moreover, there are 24 superordinate entity types in the CCKS2020 dataset for NIL entity classification.

Table 2. Experimental annotation data examples.

Text_Id	Text	Mention_Data
64	“《女教皇》电影完整版” “The Female Pope movie full version”	[{“kb_id”: “294192”, “mention”: “女教皇 (Female Pope)”, “offset”: “1”}, {“kb_id”: “60997”, “mention”: “电影 (movie)”, “offset”: “5”}]
37141	“成功需要运气吗” “Does success require luck?”	[{“kb_id”: “70091”, “mention”: “成功 (success)”, “offset”: “0”}, {“kb_id”: “329012”, “mention”: “运气 (luck)”, “offset”: “4”}]
25279	“吃番茄冒汗什么原因” “What is the reason of eating tomato sweat?”	[{“kb_id”: “305867”, “mention”: “番茄 (tomato)”, “offset”: “1”}, {“kb_id”: “144992”, “mention”: “冒汗 (sweat)”, “offset”: “3”}, {“kb_id”: “128398”, “mention”: “原因 (reason)”, “offset”: “7”}]

4.2. Evaluation Criteria

For a given input text of short text with n mention items: $M_n = \{m_1, m_2, m_3, \ldots, m_n\}$, each mention links to a knowledge base with entity id: $E_n = \{e_1, e_2, e_3, \ldots, e_n\}$, and the entity annotation system outputs the annotation results with entity id: $E'_n = \{e'_1, e_2, e'_3, \ldots, e'_n\}$. Moreover, equations for precision, recall, and F1 values of entity annotation are defined as follows:

$$P = \frac{\sum_{n \in N} |E_n \cap E'_n|}{\sum_{n \in N} |E'_n|}, \tag{5}$$

$$R = \frac{\sum_{n \in N} |E_n \cap E'_n|}{\sum_{n \in N} |E_n|}, \tag{6}$$

$$F1 = \frac{2 \times P \times R}{P + R}. \tag{7}$$

Since the entity mention is already given in the dataset, there are $\sum_{n \in N} |E'_n| = \sum_{n \in N} |E_n|$, in other words, $P = R = F1$. Therefore, the F1 value is employed as a comprehensive evaluation index in this paper.

4.3. Environment and Parameters

We base our model on the Pytorch framework, and the specific experimental environment of software and hardware is described in Table 3.

Table 3. Experimental environment settings.

Item	Environment
Operating system	Ubuntu 20.04.3 LTS
CPU	Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz
GPU	NVIDIA GeForce RTX 3080 Ti
Python version	3.8
Pytorch version	1.11.0

The parameters of our proposed model are detailed as follows: the RoBERTa-wwm-ext contains 12 layers of the transformer, the hidden-layer dimension is 768, the maximum sequence length is 384, the learning rate is 1×10^{-5} using the Adam optimization algorithm, the link threshold is 0.5, and the depth and level of the flow model are 3 and 2, respectively.

4.4. Experimental Results and Discussion

In this section, an extensive series of experiments are presented to investigate the validity of our proposed model. Firstly, the performance of RoBERTa is demonstrated by comparing the pre-trained language model. Secondly, the superiority of our proposed model is proved through multiple sets of comparison experiments and operational performance. Then, the enhanced performance of each module in our proposed model is analyzed with ablation experiments. Finally, the effects of learning rate and sentence vec-

tor strategy on model experimental results are investigated by comparing multiple groups. The specific experimental analysis is presented in the following subsections.

4.4.1. Comparison of Pre-Trained Model

To confirm the better performance of the RoBERTa model, a comparison of the pre-trained models is conducted on the CCKS2019 and CCKS2020 entity-linking datasets. Specifically, the pre-trained model is employed to extract text semantic information, and then the semantic information is sent to the fully connected layer for prediction. The experimental comparison results in Table 4 illustrate that the RoBERTa model performs more effectively than the BERT model. For the CCKS2020 and CCKS2019 datasets, the F1 score for RoBERTa is 1.08% and 0.97% higher than that of BERT, respectively.

Table 4. Results of different pre-trained models.

Model	F1/%(CCKS2020)	F1/%(CCKS2019)
BERT	87.61	88.10
RoBERTa	88.69	89.07

4.4.2. Comparison Methods

To verify the effectiveness of our proposed model, a comparison with some deep learning methods is conducted on the CCKS2020 dataset using the F1 score as the experimental measure, and the specific description of the comparative model as follows:

(1) BERT-TextRank: Zhan et al. [30] extracted semantic information by the BERT pre-trained model and employed the TextRank technique to increase the information of the entity description, where the number of keywords in the TextRank model is three.

(2) BERT-Flow-Label: The modules of our model are compared in the BERT-based model; to be specific, a layer flow model corrects the sentence vector output from BERT to fully represent semantic information, and label embedding is employed to further enhance the performance of multi-classification.

(3) RoBERTa-BiLSTM: BiLSTM is able to fully capture the contextual information by concatenating the forward and backward hidden-layer vectors, which is commonly applied in entity linking [49,50] to fully extract textual information. Inspired by these methods, the RoBERTa-BilSTM model is constructed to further extract textual information through the BiLSTM module.

(4) RoBERTa-Attention: Jia et al. [51] employed a layer of attention mechanism to further improve the representation of textual information by further focusing on important information and reducing the interference of irrelevant information. Hence, the RoBERTa-Attention model is created utilizing the aforementioned method as inspiration to further focus on the extracted textual information through the attention module.

(5) RoBERTa-TextRank: The Textrank technique is applied in RoBERTa to enhance text topics and facilitate the extraction of semantic information, where the number of keywords in the TextRank model is three.

The outcomes of these comparison experiments are presented in Table 5, and the following conclusions are derived from the analysis of results. To begin, the effectiveness of our flow and label-embedding modules is proved in the experiment of BERT-based model, and specifically, the F1 score of BERT-Flow-Label is 0.53% higher compared to BERT-TextRank. Then, our proposed model outperforms the RoBERTa model using the TextRank technique, in which the F1 score of RoBERTa-TextRank is 1.08% lower than our model. Finally, our proposed approach is superior to some deep learning models commonly utilized in entity linking. To be specific, the F1 score of our method is 0.98% and 0.91% higher than RoBERTa-Attention and RoBERTa-BiLSTM, respectively.

Table 5. Experimental results on CCKS2020 dataset.

Model	F1/%
BERT-TextRank	88.38
BERT-Flow-Label	88.91
RoBERTa-BiLSTM	89.11
RoBERTa-Attention	89.04
RoBERTa-TextRank	88.94
Our	90.02

In the similar way, the above experimental conclusions are also available in the CCKS2019 dataset as reported in Table 6. The difference from the CCKS2020 dataset is that there is no task of NIL entity classification in CCKS2019, the label-embedding module of our model is not utilized and only the performance of the flow module is demonstrated in the CCKS2019 dataset.

Table 6. Experimental results on CCKS2019 dataset.

Model	F1/%
BERT-TextRank	88.56
BERT-Flow	89.73
RoBERTa-BiLSTM	89.37
RoBERTa-Attention	89.70
RoBERTa-TextRank	89.33
Our_{wol}	90.40

4.4.3. Ablation Experiment

To analyze the effect of the flow module and the label-embedding module on the F1 value of our proposed model, ablation experiments are performed on the CCKS2020 dataset. The RoBERTa is employed as the baseline model and compared with adding flow and adding label embedding on the baseline, which are denoted as RoBERTa + flow and RoBERTa + label, respectively.

The results of the ablation experiments are illustrated in Table 7, and it is obvious that the flow module substantially improves the performance of model, while relatively small enhancement is observed with the label-embedding module. Specifically, the RoBERTa + Flow and RoBERTa + Label models outperform the baseline model by 1.18% and 0.10%, respectively.

Table 7. Results of the ablation experiment.

Model	F1/%
RoBERTa(baseline)	88.69
RoBERTa + Label	88.79
RoBERTa + Flow	89.87
Our	90.02

4.4.4. Operational Performance

To further demonstrate the performance advantages of our model, the parameter sizes and running times of several RoBERTa-based experiments are compared using the CCKS2020 dataset and trained on six GPUs with the specifications shown in Table 3. The parameter size and running time are benchmarked with RoBERTa, and a comparison of our model with RoBERTa-BiLSTM and RoBERTa-Attention is reported in Table 8. It is observed that our model has smaller parameter size and shorter running time, which validates the performance advantage of our model.

Table 8. Comparison of parameter size and running time.

Model	Total Parameter Size (MB)	Time/Epoch
RoBERTa(baseline)	409.074	43 m 10 s
RoBERTa-BiLSTM	439.797	58 m 16 s
RoBERTa-Attention	420.886	52 m 42 s
Our	418.869	51 m 13 s

4.4.5. Different Learning Rates

To explore the influence of different learning rates on the model, a comparison of three different learning rates is conducted on our proposed model. To be specific, the learning rates set on the CCKS2020 dataset are 1×10^{-6}, 1×10^{-5}, and 5×10^{-5}, respectively, and the learning rates set on the CCKS2019 dataset are 1×10^{-6}, 1×10^{-5}, and 2×10^{-5}, respectively.

The experimental results on both datasets with different learning rates are illustrated in Figures 6 and 7, respectively, and the following conclusions are obtained from the analysis of the results. Firstly, our model achieves the optimum performance with a learning rate of 1×10^{-5}, and the F1 values on CCKS2020 and CCKS2019 are 90.02% and 90.40%, respectively. Secondly, excessive learning rate leads to failure of model convergence. Specifically, when the learning rate increases to 5×10^{-5} and 2×10^{-5} on CCKS2020 and CCKS2019, the F1 value drops by 2.47% and 2.73%, respectively. Finally, a learning rate that is too small will cause model to fall into a local optimum rather than a global optimum. To be specific, the F1 values decrease by 3.91% and 0.85% when the learning rate decreases to 1×10^{-6} on CCKS2020 and CCKS2019, respectively.

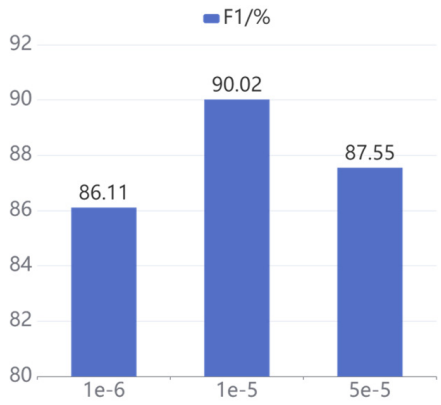

Figure 6. Comparison results for different learning rates on CCKS2020.

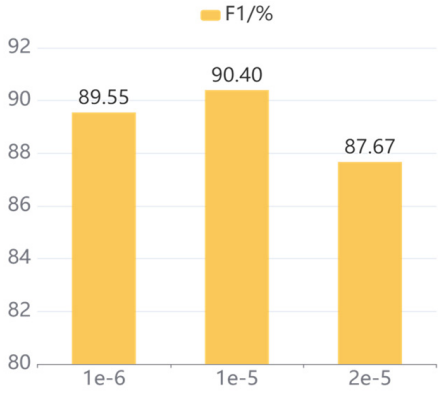

Figure 7. Comparison results for different learning rates on CCKS2019.

4.4.6. Different Sentence Vector Strategies

To investigate the effect of different sentence vector strategies on our proposed model, four commonly employed sentence vector strategies are set up in the experiment: CLS, All, Last, and First-Last. Specifically, CLS is adopted to represent the sentence-level information of the output in the pre-trained model, the average pooling of the hidden states is denoted by All, the mean pooling of the hidden states in the last layer is indicated by Last, and First-Last is applied to denote the average pooling of the first and last hidden states.

The comparison results for different sentence vector strategies on CCKS2020 and CCKS2019 are illustrated in Figures 8 and 9, respectively. It can be concluded that the CLS strategy achieves the best results on both CCKS2020 and CCKS2019 datasets, and the F1 score of other strategies is decreased with different degrees, so the CLS strategy is adopted as the sentence vector representation of our proposed model.

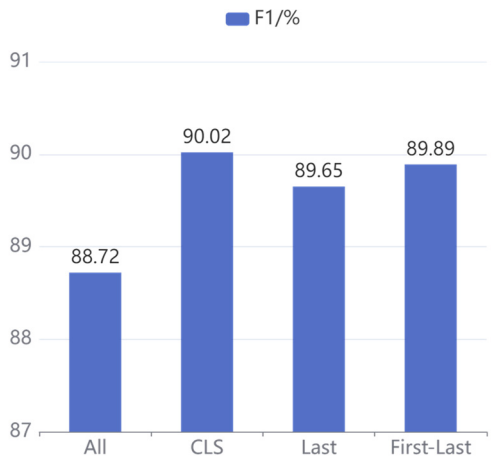

Figure 8. Comparison results for different sentence vectors on CCKS2020.

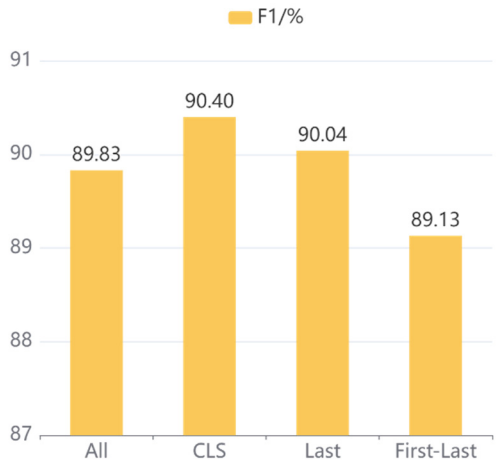

Figure 9. Comparison results for different sentence vectors on CCKS2019.

5. Conclusions and the Future Work

In this paper, we propose a model for entity linking in Chinese short text based on normalized RoBERTa sentence vectors and label embedding, which alleviates the semantic sparsity problem of Chinese short texts by making full use of semantic information. The proposed model employs the flow model to post-process the sentence vectors for fully utilizing the semantic information in the limited text. In addition, we further enhance the entity-linking performance by entity typing with label embedding. The experimental results on CCKS2020 and CCKS2019 datasets demonstrate that our model outperforms existing entity-linking methods and some commonly utilized deep learning approaches.

Although our model achieves advanced results, it still has some limitations. Since our model is based on the recognized short texts for entity linking, the effectiveness in practical applications will be limited by the accuracy of the recognized entities. In contrast, directly linking unstructured short texts can eliminate erroneous referents, which is the direction of our future work. In the future, we will combine named entity recognition with entity linking for joint learning to directly obtain the recognized referred and linked entities. Furthermore, our entity-linking method will be further applied to the expansion of knowledge graph and intelligent systems of question-answering.

Author Contributions: Conceptualization, L.G. and L.Z. (Lijuan Zhang); methodology, L.G.; software, L.G.; validation, L.G. and L.Z. (Lijuan Zhang); formal analysis, L.G.; investigation, L.G.; resources, L.Z. (Lei Zhang) and J.H.; data curation, L.G.; writing—original draft preparation, L.G.; writing—review and editing, L.Z. (Lijuan Zhang); visualization, L.G.; supervision, L.Z. (Lijuan Zhang) and J.H.; project administration, L.Z. (Lei Zhang) and J.H.; funding acquisition, L.Z. (Lei Zhang). All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by the Soft Science Research Program of Zhejiang Province under Grant [No.2021C25051] and Zhejiang Province Key Research and Development Project under Grant [No.2020C03071, No.2021C03145].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Jiang, H.; Gurajada, S.; Lu, Q.; Neelam, S.; Popa, L.; Sen, P.; Li, Y.; Gray, A. LNN-EL: A Neuro-Symbolic Approach to Short-text Entity Linking. *arXiv* **2021**, arXiv:2106.09795.
2. Gu, Y.; Qu, X.; Wang, Z.; Huai, B.; Yuan, N.J.; Gui, X. Read, retrospect, select: An MRC framework to short text entity linking. *arXiv* **2021**, arXiv:2101.02394.
3. Gupta, N.; Singh, S.; Roth, D. Entity linking via joint encoding of types, descriptions, and context. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 2681–2690.
4. Gillick, D.; Kulkarni, S.; Lansing, L.; Presta, A.; Baldridge, J.; Ie, E.; Garcia-Olano, D. Learning dense representations for entity retrieval. *arXiv* **2019**, arXiv:1909.10506.
5. Logeswaran, L.; Chang, M.-W.; Lee, K.; Toutanova, K.; Devlin, J.; Lee, H. Zero-shot entity linking by reading entity descriptions. *arXiv* **2019**, arXiv:1906.07348.
6. Ou, J.; Liu, N.N.; Kai, Z.; Yu, Y.; Yang, Q. Transferring Topical Knowledge from Auxiliary Long Texts for Short Text Clustering. In Proceedings of the 20th ACM Conference on Information & Knowledge Management, Glasgow, Scotland, UK, 24–28 October 2011; pp. 775–784.
7. Bunescu, R.; Pasca, M. Using encyclopedic knowledge for named entity disambiguation. In Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy, 3–7 April 2006.
8. Mann, G.; Yarowsky, D. Unsupervised personal name disambiguation. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, Edmonton, AB, Cannada, 27 May–1 June 2003; pp. 33–40.
9. Zhang, W.; Tan, C.L.; Sim, Y.C.; Su, J. NUS-I2R: Learning a Combined System for Entity Linking. In Proceedings of the 3th Text Analysis Conference, Gaithersburg, MD, USA, 15–16 November 2010.
10. Yupeng, J.; Hongxu, H.; Ping, Y. LSA-Based Chinese-Slavic Mongolian NER Disambiguation. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 703–708.
11. Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. Indexing by Latent Semantic Analysis. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 391–407.
12. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
13. Wang, P.; Xian, Y.; Guo, J. A novel method using word vector and graphical models for entity disambiguation in specific topic domains. *CAAI Trans. Intell. Syst.* **2016**, *11*, 366.
14. Ganea, O.-E.; Hofmann, T. Deep joint entity disambiguation with local neural attention. *arXiv* **2017**, arXiv:1704.04920.

15. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
16. McCann, B.; Bradbury, J.; Xiong, C.; Socher, R. Learned in Translation: Contextualized Word Vectors. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
17. Wu, C.; Luo, C.; Xiong, N.; Zhang, W.; Kim, T.H. A Greedy Deep Learning Method for Medical Disease Analysis. *IEEE Access* **2018**, *6*, 20021–20030.
18. He, Z.; Liu, S.; Li, M.; Zhou, M.; Zhang, L.; Wang, H. Learning entity representation for entity disambiguation. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Sofia, Bulgaria, 4–9 August 2013; pp. 30–34.
19. Cao, Y.; Hou, L.; Li, J.; Liu, Z. Neural collective entity linking. *arXiv* **2018**, arXiv:1811.08603.
20. Jia, N.; Cheng, X.; Su, S.; Ding, L. CoGCN: Combining co-attention with graph convolutional network for entity linking with knowledge graphs. *Expert Syst. Wiley Online Libr.* **2021**, *38*, e12606.
21. Wu, J.; Zhang, R.; Mao, Y.; Guo, H.; Soflaei, M.; Huai, J. Dynamic graph convolutional networks for entity linking. In Proceedings of the Web Conference 2020, Taipei, China, 20–24 April 2020; pp. 1149–1159.
22. Cheng, H.; Xie, Z.; Shi, Y.; Xiong, N. Multi-Step Data Prediction in Wireless Sensor Networks Based on One-Dimensional CNN and Bidirectional LSTM. *IEEE Access* **2019**, *7*, 117883–117896.
23. Zeng, W.; Tang, J.; Zhao, X. Entity linking on Chinese microblogs via deep neural network. *IEEE Access* **2018**, *6*, 25908–25920.
24. Hu, S.; Tan, Z.; Zeng, W.; Ge, B.; Xiao, W. Entity linking via symmetrical attention-based neural network and entity structural features. *Symmetry Multidiscip. Digit. Publ. Inst.* **2019**, *11*, 453.
25. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 2227–2237.
26. Onoe, Y.; Durrett, G. Fine-grained entity typing for domain independent entity linking. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 8576–8583.
27. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
28. Chen, S.; Wang, J.; Jiang, F.; Lin, C.Y. Improving entity linking by modeling latent entity type information. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 7529–7537.
29. Cheng, J.; Pan, C.; Dang, J.; Yang, Z.; Guo, X.; Zhang, L.; Zhang, F. Entity linking for Chinese short texts based on BERT and entity name embeddings. In Proceedings of the 2019 China Conference on Knowledge Graph and Semantic Computing (CCKS2019), Hangzhou, China, 24–27 August 2019.
30. Zhan, F.; Zhu, Y.; Liang, W.; Ji, X. Entity Linking Via BERT and TextRank Keyword Extraction. *J. Hunan Univ. Technol.* **2020**, *34*, 63–70.
31. Zhao, Y.; Wang, Y.; Yang, N. Chinese Short Text Entity Linking Based On Semantic Similarity and Entity Correlation. In Proceedings of the 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 9–11 November 2020; pp. 426–431.
32. Jiang, L.; Altenbek, G.; Wu, D.; Ma, Y.; Aierzhati, H. Chinese Short Text Entity Disambiguation Based on the Dual-Channel Hybrid Network. *IEEE Access* **2020**, *8*, 206164–206173.
33. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
34. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
35. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the sentence embeddings from pre-trained language models. *arXiv* **2020**, arXiv:2011.05864,.
36. Xiong, Y.; Feng, Y.; Wu, H.; Kamigaito, H.; Okimura, M. Fusing Label Embedding into BERT: An Efficient Improvement for Text Classification. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Bangkok, Thailand, 1–6 August 2021; pp. 1743–1750.
37. Phan, M.C.; Sun, A.; Tay, Y.; Han, J.; Li, C. NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1667–1676.
38. Gattani, A.; Lamba, D.S.; Garera, N.; Tiwari, M.; Chai, X.; Das, S.; Subramaniam, S.; Rajaraman, A.; Harinarayan, V.; Doan, A. Entity extraction, linking, classification, and tagging for social media: A wikipedia-based approach. *Proc. VLDB Endow. VLDB Endow.* **2013**, *6*, 1126–1137.
39. Urata, T.; Maeda, A. An entity disambiguation approach based on wikipedia for entity linking in microblogs. In Proceedings of the 2017 6th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), Hamamatsu, Japan, 9–13 July 2017; pp. 334–338.
40. Nozza, D.; Sas, C.; Fersini, E.; Messina, E. Word embeddings for unsupervised named entity linking. In Proceedings of the 12th International Conference on Knowledge Science, Engineering and Management, Athens, Greece, 28–30 August 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 115–132.

41. Tan, C.; Wei, F.; Ren, P.; Lv, W.; Zhou, M. Entity linking for queries by searching wikipedia sentences. *arXiv* **2017**, arXiv:1704.02788.
42. Munnelly, G.; Lawless, S. Investigating entity linking in early english legal documents. In Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, Fort Worth, TX, USA, 3–7 June 2018; pp. 59–68.
43. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.; Kaiser, L.; Polpsukhin, I. Attention is all you need. In Proceedings of the 31st Congerence on Neural Information Processing Systems, Long Brach, CA, USA, 4–9 December 2017.
44. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z. Pre-training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio Speech Lang. Process. IEEE* **2021**, *29*, 3504–3514.
45. Gao, J.; He, D.; Tan, X.; Qin, T.; Wang, L.; Liu, T.Y. Representation degeneration problem in training natural language generation models. *arXiv* **2019**, arXiv:1907.12009.
46. Wang, L.; Huang, J.; Huang, K.; Hu, Z.; Wang, G.; Gu, Q. Improving neural language generation with spectrum control. In Proceedings of the 7th International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
47. Ethayarajh, K. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. *arXiv* **2019**, arXiv:1909.00512.
48. Dinh, L.; Krueger, D.; Bengio, Y. NICE: Non-linear Independent Components Estimation. *arXiv* **2014**, arXiv:1410.8516.
49. Luo, A.; Gao, S.; Xu, Y. Deep semantic match model for entity linking using knowledge graph and text. *Procedia Comput. Sci. Elsevier* **2018**, *129*, 110–114.
50. Lu, W.; Zhou, Y.; Lu, H.; Ma, P.; Zhang, Z.; Wei, B. Boosting collective entity linking via type-guided semantic embedding. In Proceedings of the National CCF Conference on Natural Language Processing and Chinese Computing, Dalian, China, 8–12 November 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 541–553.
51. Jia, B.; Wu, Z.; Zhou, P.; Wu, B. Entity Linking Based on Sentence Representation. *Complex. Hindawi* **2021**, *2021*, 8895742.