applied sciences

MDPI

Article

Deep Contrast Learning Approach for Address Semantic Matching

Jian Chen, Jianpeng Chen, Xiangrong She, Jian Mao and Gang Chen *

Institute of Smart City Research (Wuhu), University of Science and Technology of China, Wuhu 241000, China; chenj@ustc.win (J.C.); chenjp@ustc.win (J.C.); shexr@ustc.win (X.S.); nh146@163.com (J.M.)
* Correspondence: cheng@ustc.win; Tel.: +86-173-5294-4872

Abstract: Address is a structured description used to identify a specific place or point of interest, and it provides an effective way to locate people or objects. The standardization of Chinese place name and address occupies an important position in the construction of a smart city. Traditional address specification technology often adopts methods based on text similarity or rule bases, which cannot handle complex, missing, and redundant address information well. This paper transforms the task of address standardization into calculating the similarity of address pairs, and proposes a contrast learning address matching model based on the attention-Bi-LSTM-CNN network (ABLC). First of all, ABLC use the Trie syntax tree algorithm to extract Chinese address elements. Next, based on the basic idea of contrast learning, a hybrid neural network is applied to learn the semantic information in the address. Finally, Manhattan distance is calculated as the similarity of the two addresses. Experiments on the self-constructed dataset with data augmentation demonstrate that the proposed model has better stability and performance compared with other baselines.

Keywords: address matching; smart city; contrast learning; neural networks; data augmentation

Citation: Chen, J.; Chen, J.; She, X.; Mao, J.; Chen, G. Deep Contrast Learning Approach for Address Semantic Matching. *Appl. Sci.* **2021**, *11*, 7608. https://doi.org/10.3390/app11167608

Academic Editors: Alexei Gvishiani and Boris Dzeboev

Received: 21 July 2021
Accepted: 17 August 2021
Published: 19 August 2021

1. Introduction

Geographical addresses are the most important basic data resources in the construction of a smart city. How to dig out potential associations between address texts and use the result to serve for standardization construction is a key issue that directly affects the level of smart city construction.

The early research methods on address mainly focused on text similarity. The literal similarity between the two geographical addresses was calculated from a certain measurement dimension and the threshold was manually set [1]. Specifically, the edit distance [2–4] is a traditional way which defines the similarity as the minimum number of character editing operations required to convert one string to another string, which is very easy to be applied in real work. Subsequently, Jaccard [5] brought up a new way which obtains a more accurate effect on short address by calculating the local similarity of two addresses, but it does not work well for long addresses. Afterwards, the N-gram approach based on vector space was proposed [6], which converts addresses to vector representations in the same vector space, and then calculates the similarity using mathematical methods for example cosine similarity [7]. Compared with previous methods, the N-gram approach improves the effect and obtains a better performance. Nowadays, all the traditional methods mentioned above are still inadequate.

More recently, with the diversification of addresses and the higher requirement to process a large number of addresses than before, traditional address matching methods obviously cannot meet the requirements. A new method based on address structure and address element extraction is proposed, which uses the hierarchical syntax tree to identify address and then do further address matching work [8]. Basically, the way of acquiring address element is mainly by dividing into word segmentation with dictionaries, probability distributions, such as conditional random fields, hidden Markov models [9,10], or natural language word segmentation tools (Jieba, THULAC, etc.). Some scholars have put

forward corresponding solutions to the extraction of address elements: some rule-based and fuzzy Chinese address coding methods are raised to establish a standard address database and generate matching rules [11–13]. However, this method relies on the completeness of address database, and it is difficult to formulate all the rules as the growth of database. Tian proposed an optimized Chinese address matching model and provided a coding service with this model [14]. Zhang used Bert as pretrained embedding model and applied CRF algorithm to extract address element and semantic features [15]. Comber embed word2vec into CRF to convert address elements into a fixed dimension semantic representation vector [16].

Nevertheless, semantic information cannot be obtained effectively when dealing with longer address records and the distribution of information density is uneven. To address this kind of problem, follow-up research begins to apply a neural network to do this task, such as using CNN or RNN [17]. Santos proposed a multi-model fusion approach which apply RNN and GRU as address semantic information modeling [18]. This model has achieved a good performance improvement compared to single similarity measurement-based models and some supervised learning methods. Next, Lai combined the advantages of RNN and CNN models, proposed the RCNN model, which uses a bidirectional structure and embeds BiRNN structure into convolutional layer [19]. This kind of structure effectively reduces the network noise and maximizes the ability to extract the context information of addresses. In the field of smart cities, there are also many researchers trying to introduce deep learning to cope with city development issues, such as the cities expansion and personalized POI recommendation [20,21]. Karimzadeh proposed a geographic analysis system for NER recognition, which can efficiently sort out geographic analysis problems [22]. After that, the deep learning methods are verified to solve the spatial data and the urban geographic problems, and prove that deep learning ways are the most effective way and have broad application prospects [23]. Grekousis have analyzed more than 140 articles about using artificial neural networks to settle the urban geographic problems. In summary, artificial neural networks have obvious advantages over traditional methods [24].

The semantic representation of addresses based on deep learning is essentially an NLP problem [25,26]. In the field of NLP, contrastive learning algorithms have recently been widely proposed [27,28]. Contrastive learning algorithm is a subset of deep learning. Its goal is to bring the enhanced new samples as close as possible in the embedding space and make different samples as far away as possible. How to construct examples is an important issue in contrastive learning. For the translation task, Yang changed the number of omitted words, word frequency, and part of speech according to the actual translation, designed different types of negative examples to realize data augmentation [29]. Wu and Meng proposed to use word deletion, reordering, and substitution to achieve it [30,31]. However, due to the inherent discrete characteristics of Chinese addresses, it is extremely difficult to implement data augmentation simply through text processing.

In summary, the approaches are difficult to cope with addresses that contain complex structures or contain redundant information. The reason is that these methods are lacking in terms of understanding the semantics of the address, and they also cannot extract the semantic features of the address well. At the same time, these models often focus on dataset built with specific conditions that do not provide effectively help on enhancing generalization ability, which cannot fundamentally improve the model performance.

To address these problems, this paper proposes a contrast learning address matching algorithm. First, ABLC use the Trie syntax tree structure to construct a standard address tree to extract address elements, then uses Bi-LSTM and CNN models to embed the address into vectors with semantic information. Following that, we introduce the attention mechanism to get position-aware information from the context, so as to further improve the accuracy of semantic representation. In the end, the corresponding Manhattan distance is calculated between two semantic vectors of address pair, which can be considered as the similarity of two addresses. Furthermore, we introduced a data augmentation method to extend

the existing address dataset, and then the model has been significantly improved in the stability and the ability to perceive similar addresses is competitive.

The contributions of this paper are as follows: (1) Propose a contrast learning address matching algorithm that captures similarities and differences between the input address pairs so as to achieve the judgment of similarity and dissimilarity of address pairs. (2) Propose a semantic-based address representation model with a hybrid neural network that incorporates an attention mechanism. The model extracts local and global features of the input data in addition to giving higher weights to important information in the address, so as to more effectively capture key information from addresses. (3) Propose an address data augmentation method to improve the performance of the model. By constructing an address enhancement dataset based on the uniqueness of addresses and combining the dropout strategy to achieve data enhancement, the overall performance of the model is improved with better generalization capability.

2. Materials and Methods

In this section, we propose a semantic-based address matching framework according to the characteristics of the address. We first use the Trie syntax tree to build a standard address model and apply it to extract address elements. Additionally, then we create a contrast learning model which is based on a hybrid neural network, to perform semantic representation of the address. Finally, the similarity between address pair is obtained by calculating the Manhattan distance. Furthermore, the data augmentation method is introduced to construct address datasets, which improves the accuracy of address matching and the performance of the model. The address matching framework is referred to as the ABLC model and the algorithm description is as below Algorithm 1 shown.

Algorithm 1 The ABLC algorithm

Input: address set {A}, address text pair $(a_i \in \mathrm{A}, a_j \in \mathrm{A})$
Output : similarity of two address text $sim(a_i, a_j)$

```
Initialize sepResult with null
divisionTree ← BuildTree(A)
for ele in [a_i, a_j] do
  for node in divisionTree do
    if headof(ele, len(node)) == node:
      sepList ← node
      ele.delete(node)
    if node == LastNode(A):
        sepList ← ele
    sepResult ← set_List
similarity ← ABLC(sepResult [0], sepResult [1])
sim(a_i, a_j) ← similarity
```

2.1. Problem Definition

We define address matching in this paper according to the below description: assume D_{sa} containing N address datasets $D_{sa} = \{sa_1, sa_2, \ldots sa_n\}$, for a certain element sa_i from D_{sa}, the task goal of this paper is to find an address pair $\{sa_i, sa_j\}$ and satisfy: $similarity(sa_i, sa_j) \geq \eta$,where $sa_i \in D_{sa}, sa_j \in D_{sa}$ and $sa_i \neq sa_j$, η is the set threshold.

2.2. Address Model

The particularity of the Chinese language leads to the particularity of Chinese addresses, which is mainly reflected in the following aspects: (1) Multiple: An address contains multiple place names; (2) Hierarchical: Address description is usually in sequence from large area to small area; (3) Detailed: The standard address contains the place name of each level. The Chinese address is composed of multiple address elements and a valid address element should include one of different level address names, such as the admin-

istrative division, the street, the neighborhood, the door, the landmark, and the point of interest. Several address description patterns commonly used now are: administrative division + street (road or lane) + house number; administrative division + community (natural village) + house number; administrative division + (street, road and lane) + point of interest (marker). Administrative divisions can be divided into provinces, cities, districts (counties), streets (towns), and communities (administrative villages).

The Trie syntax tree is a kind of hash-tree structure. Generally, it is used to store and sort a large number of strings. Unlike a binary tree, the key point of Trie syntax tree is that the string is not directly stored in the node, but is determined by the position of the node in the tree. Its advantage is to minimize unnecessary string comparisons and improve the query efficiency. All descendants of node have the same prefix, which is the string corresponding to this node. Additionally, the root node corresponds to an empty string. Basically, not all nodes have corresponding values, only the leaf nodes and some internal nodes have relevant values. This paper constructs the Trie syntax address tree, as shown in Figure 1, which is used to extract address elements.

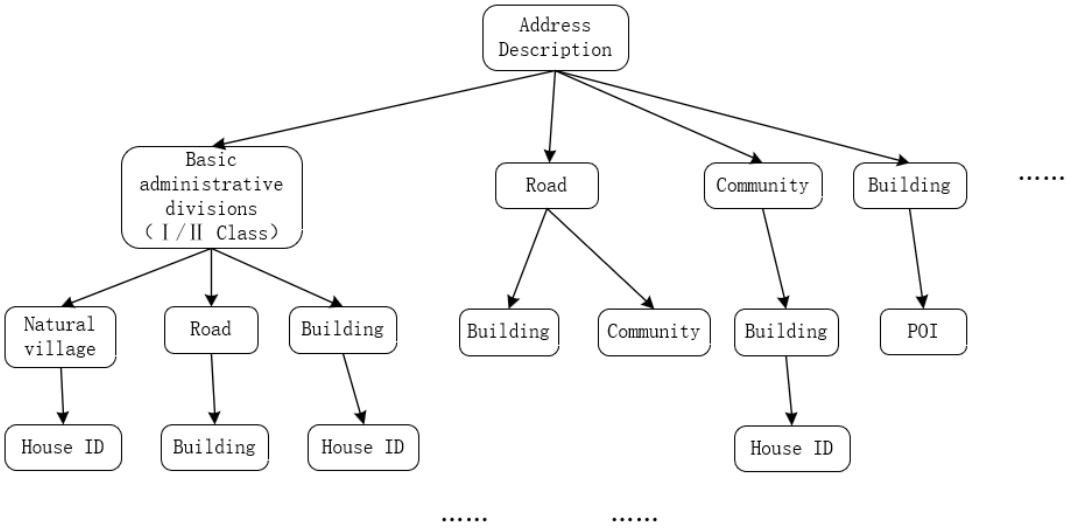


Figure 1. Trie syntax address tree.

2.3. Address Semantic Contrast Learning Model

This section introduces a semantic-based address contrast learning model which is fused with attention mechanism, Bi-LSTM and CNN network. The model is established based on the characteristics of the Chinese address and advantages of each sub-network in the hybrid neural network model. It accepts the input of the address pair, and, respectively, generates the semantic vector representation of the address, and finally determines whether the address pair is similar by calculating the Manhattan distance. The overall structure of the model is shown in Figure 2. The contrast learning model contains embedding stage, Bi-LSTM stage, CNN stage, attention stage, and semantic distance calculation stage. The specific details of each stage are explained as below.

2.3.1. Embedding

The embedding stage mainly focuses on converting the Chinese address into vectors, that is, maps the input address into a fixed m × n matrix. Chinese address is actually a special language description which the words have no formal delimiters, such as blank space. Therefore, the address needs to be segmented before word embedding and we should pay more attention to dividing the place name address into various address elements. Each address element is equivalent to a word in Chinese. This paper adopts Jieba's word segmentation algorithm and loads a custom word segmentation database to split address.

The construction of the custom database is based on the particularity of city place names and addresses to supplement the correct segmentation of unidentified names by Jieba.

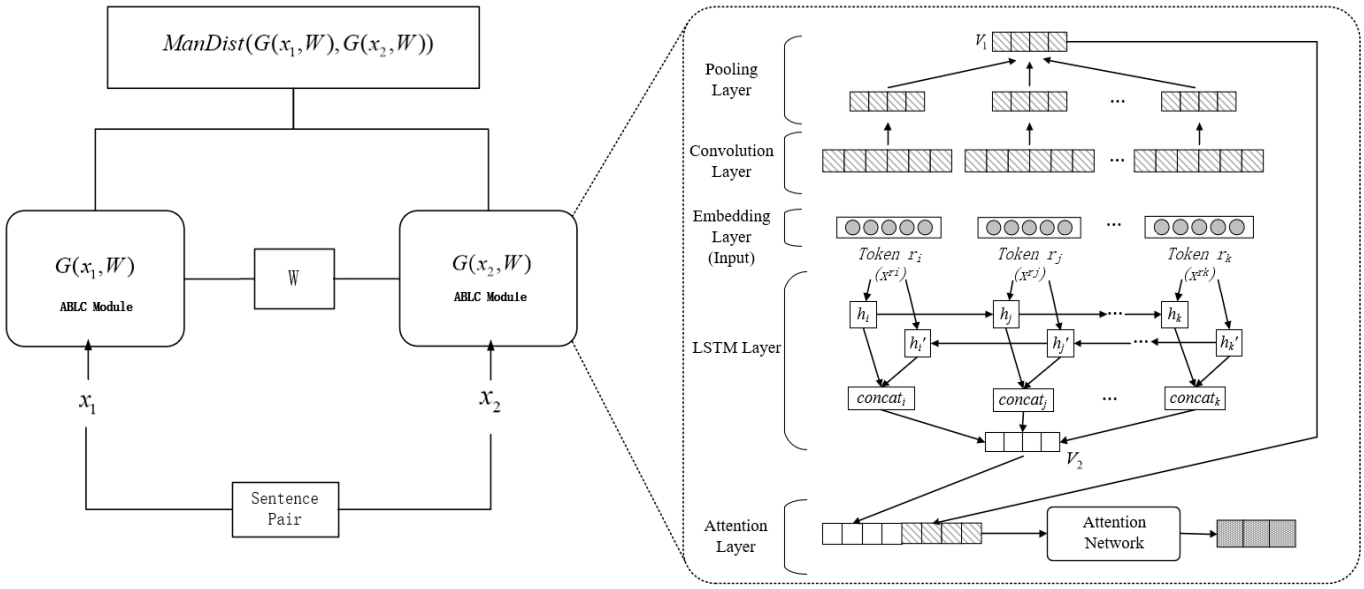


Figure 2. (**a**) Overall framework of the contrast learning model. (**b**) The structure of ABLC model.

Suppose the address A is composed of N words, namely $A = \{a_1, a_2, \ldots, a_N\}$. For each word in address A, you can use the word vector dictionary $D^w \in \mathrm{R}^{d^w|V|}$. Where V is the number of the vocabulary and d^w is the dimension of the vocabulary. The word vector dictionary D^w is obtained through learning, and the dimension of the word vector d^w is set according to requirements. Therefore, the vector of words a_i in address A is:

$$e_i = D^w V^i \tag{1}$$

where V^i is a vector of length $|V|$, and its value is 1 at e_i and 0 at the rest position. In this way, the vector of address A can be expressed as $e = \{e_1, e_2, \ldots, e_T\}$.

This paper limits the maximum length N = 20 after word segmentation for each address A. The size of the vocabulary is 10 W, and the dimension of the word vector is 300, that is, each address is mapped into a 20 × 300 vector after the embedding layer, which is used as the input of the subsequent stage.

2.3.2. Bi-LSTM

LSTM is a kind of RNN, mainly to solve the problem of gradient disappearance and gradient explosion in the training process. LSTM has better performance in long sequences [32]. The LSTM neural network uses three gate structures: input gate, forget gate and output gate to maintain and update the increase and decrease in information in the cell. However, a one-way LSTM can only process information in one direction, and cannot process information in another direction. The bidirectional LSTM is a further extension to solve the defects of LSTM. This paper uses bidirectional LSTM to extract feature information to learn address features fully. Specifically, two different LSTM neural network layers are used to traverse from the front and the back of the Chinese address, respectively, so that the address information of the two directions can be saved. Compared with the one-way LSTM, Bi-LSTM cannot only save the previous context address information, but also consider the future context address information. Therefore, the semantic representation is extracted more completely.

First, the forget gate generates a value f_t between 0 and 1 based on the output h_{t-1} from the previous memory unit and input data x_t, to determine how much information is lost in the last long-term state. h_{t-1} and x_t through the input gate to determine the update information to i_t, and in addition, through a tan h layer to get the new candidate memory unit information C_t'. Additionally, the last long-term status C_{t-1} is updated to C_t through the operation of the forget gate and the input gate. Finally, the judgment is obtained from the output gate, to multiply the value o_t between -1 and 1. The multiply result h_t is used to determine which state characteristics of the current memory cell are output. As shown in the following formula:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$C_t' = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * C_t' \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

This model uses LSTM to solve long-term dependence, and combines the complementary information of the positive and negative directions of Bi-LSTM to fully learn the address text characteristics as shown in Figure 3. In this experiment, the number of hidden neurons is 100, and the dropout parameter is set to 0.5.

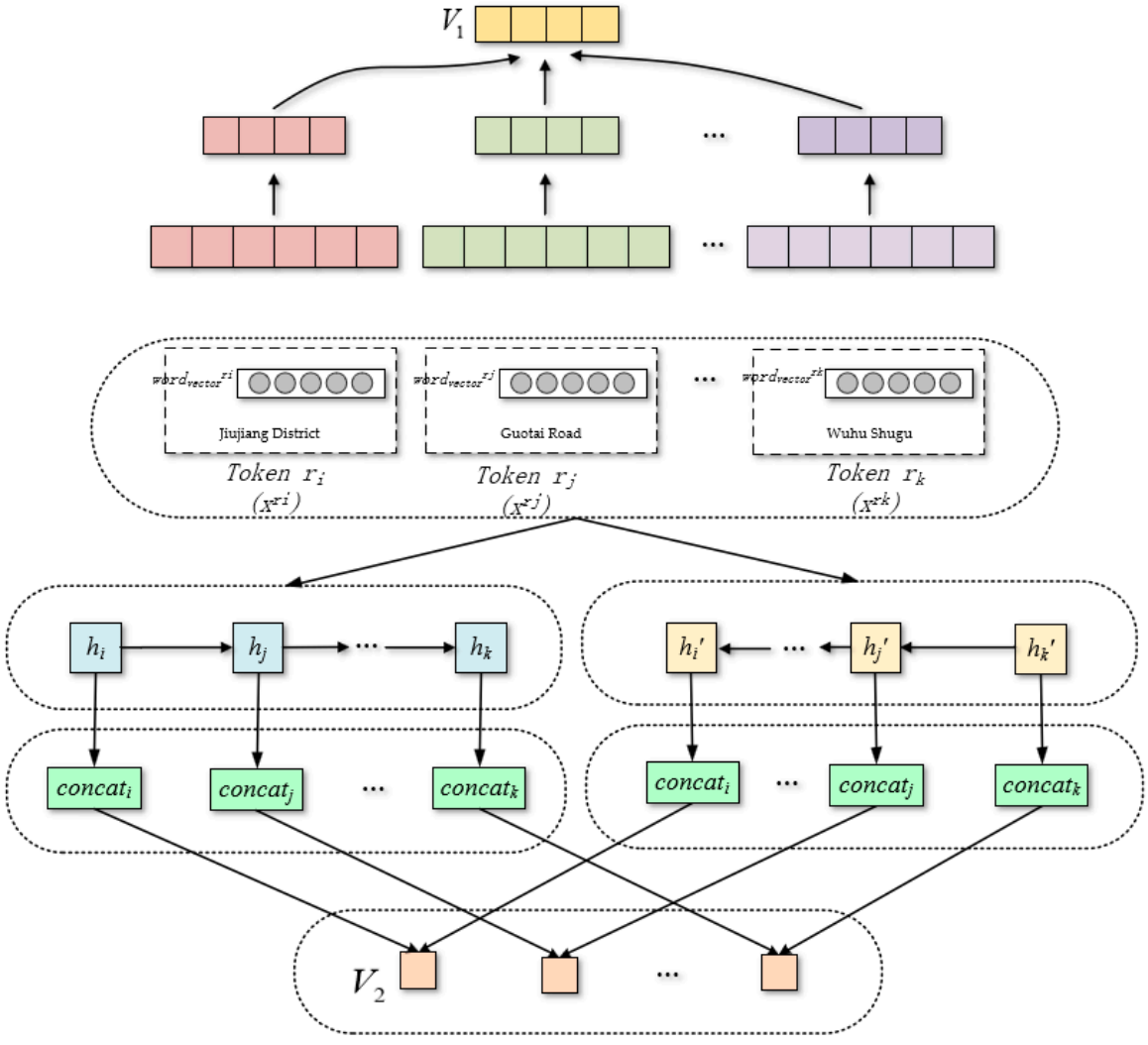


Figure 3. The Structure of Bi-LSTM module.

2.3.3. CNN

Convolutional neural network CNN has achieved good results in the field of computer vision [33], and the convolution kernel pooling is actually a process of feature extraction. The idea of CNN is to localize the overall data, use the convolution kernel function to extract the features in each local data, and then reconstruct all the fragmented features. Finally, the extraction of the overall information is realized under the guidance of the objective function.

Address text has multi-name and hierarchical property, that is, it is a text composed of a series of geographical entities, such as "Wuhu Shugu A Block 6 (POI), Guotai Road No. 2 (jieluxiang), Jiujiang District (District/County), Wuhu City (City), Anhui Province (Province)". The changes in the different levels of the Chinese address are consistent with the application scenarios of the CNN window. Based on this, the core convolution form based on CNN is used to extract the features of the address-level data. This paper uses 1-dimensional Convolution1D for convolution. The specific convolution structure is shown in the Figure 4: First, ZeroPadding1D is used to fill the edges of the input word vector matrix with zero values, and then 100 filters with a length of 5 convolution kernels are used for convolution. It is equivalent to using a $100 \times 5 \times 300$ convolution kernel to perform a convolution operation on the output matrix of the embedding layer. After the convolution operation, the extractable size is $20 \times 5 \times 300$. Then, select MaxPooling1D with pool_size of 2 to sample the convolved features, that is, take the maximum value of the convolved local area, and finally the output dimension is 20×100, as the input of the next stage.

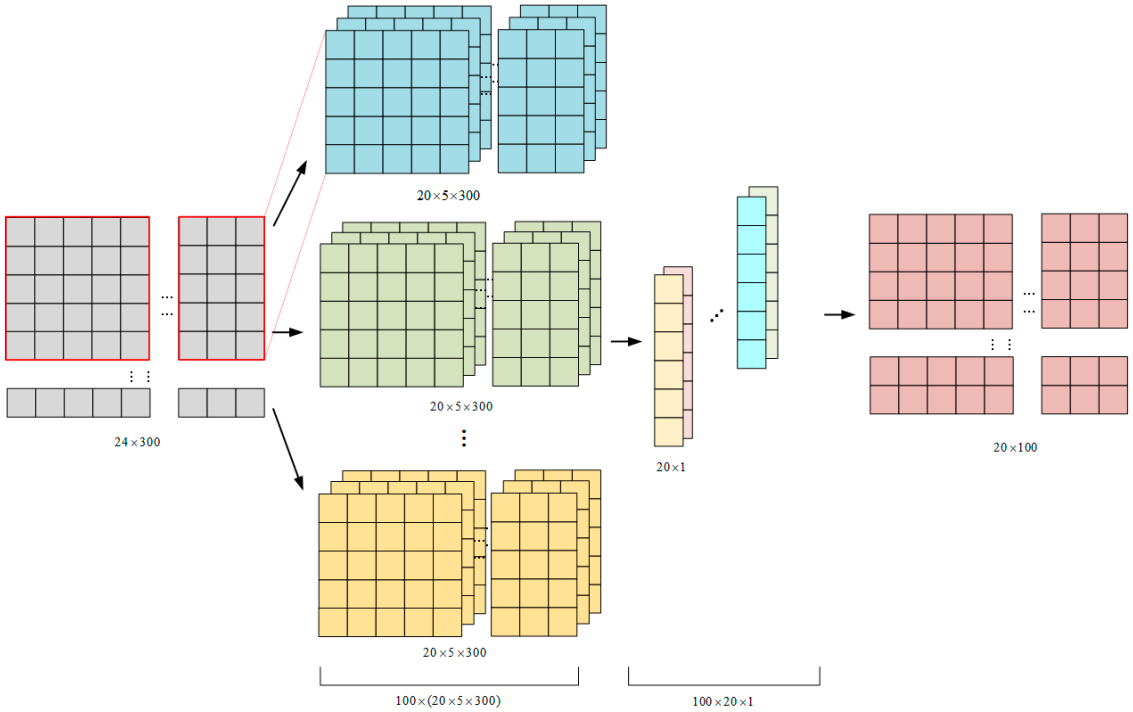


Figure 4. The structure of CNN module.

2.3.4. Attention

The human visual perception to the external world is not the full range, but focuses on a specific part according to the purpose [34]. In the field of NLP, self-attention simulates this learning process of humans. For a specific character, a certain weight is assigned to the character based on the whole text, and then integrates all the weights to determine the semantic representation of the character. According to the habit of describing addresses in Chinese, it is customary to put meaningful words or words of specific addresses in front of the expression, so different weights should be assigned to each word. For example: "1st

village,1st village group", "No. 6, 1st floor, 1st community", "No. 1, building 11, district 4, 1st community", "No. 5–6, Zone E, 1st mall", "1 Building No. 11 Facade, 1st road, 1st community ". In this part, we propose to use the attention mechanism to represent the semantic information of the address, so that the semantic vector can express richer semantic information by assigning different weights.

The definition: H is the input vector containing $[h_1, h_2, \ldots, h_T]$, where T is the length of the sentence. The input vector at this stage is derived from the weighted output of the CNN and Bi-LSTM. The related formulas are described as follows:

$$A' = \tan h(H) \tag{8}$$

$$\alpha = \text{softmax}(\mathrm{W}^{\mathrm{T}} A') \tag{9}$$

$$A'' = H\alpha^T \tag{10}$$

where $H \in \mathrm{R}^{d^w \times T}$, d^W is the dimension of the word vector, W is obtained through training, and W^{T} is transposition, A'' is the vector representation after the attention stage.

Then, the final representation of each address vector is:

$$A = \sum_{axis=1, i=1}^{d^w} A'' \tag{11}$$

Among them, each row vector of the matrix is added to obtain the final vector.

2.3.5. Manhattan Distance

This paper applies Manhattan distance to calculate the similarity between a pair of addresses. The definition $A^{left} = (A^l{}_1, A^l{}_2, \ldots, A^l{}_n)$ and $A^{right} = (A^r{}_1, A^r{}_2, \ldots, A^r{}_n)$ vectors are, respectively, semantic representation of the address pair after attention stage, then the Manhattan distance of A^{left} and A^{right} can be expressed as:

$$Md = \sum_{i=1}^{n} \left| A^{left}{}_i - A^{right}{}_i \right| \tag{12}$$

Use the sigmoid function to predict the final similarity y value:

$$y = sig\text{mo}id(Md) \tag{13}$$

3. Results

3.1. Dataset

In order to evaluate the stability of the model proposed in this paper, we leverage a standard address library to construct an address data sets containing 195,405 pairs of address, and then employs manual marking to mark whether the two addresses are similar or not. An example of address pair is shown in Table 1. From the address pair dataset, we select 10% of the address pairs as the test sets, which contains 13,027 pairs of similar and 6513 pairs of non-similar. The ratio of positive and negative samples is around 2:1. For the remaining address dataset, we use a ten-fold cross-validation strategy for training and verification. In the data preprocessing stage, we use the third-party tool Jieba to segment the addresses. Considering that the address, as a short text with a special structure, may contain a large number of unique vocabularies of place names, we used a custom stop vocabulary list when segmenting words:

Table 1. Address pairs on custom dataset.

Address 1	Address 2	Similarity
Dormitory of Xinhua Bookstore, Chaowu Road, Jinhe Community, Wuwei City	Interior of Xinhua Bookstore, Chaowulu, Jinhe Community, Wucheng Town, Wuwei County, Anhui Province	1
No. 1, Wuteng Village, Xinwu Economic Development Zone, Wuhu County, Wuhu	Xiaocun Nature Village, Zhongyao Village Villagers Committee, Liulang Town, Wuhu County, Anhui Province	0

3.2. Data Augmentation

Data augmentation is an effective way to expand the sample sets by way of changing the training data. The larger the data size and the better the quality, the trained model is able to get better predictive and generalization capabilities. Address pairs on data augmentation are constructed as Table 2. It is different from the image field through the introduction of noise or cropping to achieve data enhancement [35–37]. In the field of NLP, small changes in string may lead to huge deviations in meaning, so it is hard to perform simple transformations on data. In text classification tasks, scholars have proposed several text enhancements based on noise which is synonym substitution, random insertion, random exchange, and random deletion [38]. Aiming at the particularity of the address, this paper adopts a data enhancement method based on the dropout strategy. Essentially, data enhancement is implemented by two ways. One way is to concatenate address with itself as a positive sample, or with a random sample from the rest addresses as a negative sample. The other way is to send same sample to dropout structure twice [39]. Specifically, assuming that the address A is input with the dropout semantic representation model, the vector obtained is $h^{(0)}$, and then the same address A is input into the semantic representation model (in this case, another random dropout) to obtain the vector $h^{(1)}$. We treat $h^{(0)}$, $h^{(1)}$ as a pair positive example.

Table 2. Address pairs on data augmentation.

Address 1	Address 2	Similarity
Xinhua Bookstore Dormitory of Xinhua Bookstore, Chaowu Road, Jinhe Community, Wuwei City, Interior of Xinhua Bookstore, Xinhua Bookstore, Chaowu Road, Jinhe Community, Wucheng Town, Wuwei County, Anhui Province	The interior of Xinhua Bookstore, Chaowulu Xinhua Bookstore, Jinhe Community, Wucheng Town, Wuwei County, Anhui Province, Dormitory of Xinhua Bookstore, Chaowu Road, Jinhe Community, Wuwei City	1
No. 1, Wuhu Wuteng Village, Xinwu Economic Development Zone, Wuhu County, Wuhu, Xiaocun Nature Village, Zhongyao Village Villagers Committee, Liulang Town, Wuhu County, Anhui Province	Xiaocun Nature Village, Zhongyao Village Villagers Committee, Liulang Town, Wuhu County, Anhui Province, No. 1, Wuhu Wuteng Village, Xinwu Economic Development Zone, Wuhu County, Wuhu	1
Dormitory of Xinhua Bookstore, Chaowu Road, Jinhe Community, Wuwei City, Interior of Xinhua Bookstore, Chaowu Road, Jinhe Community, Wucheng Town, Wuwei County, Anhui Province	No. 1, Wuhu Wuteng Village, Xinwu Economic Development Zone, Wuhu County, Wuhu, Xiaocun Nature Village, Zhongyao Village Villagers Committee, Liulang Town, Wuhu County, Anhui Province	0

3.3. Experiment

In this study, the word2vec model is used as the semantic representation model. After the address pairs are indexed as a predefined vocabulary list, the sentences are embedded as a list of word indexes. Lists that less than 20-dimensional are padded with 0 to 20-dimensional coding. As for the setting of hyperparameters, considering the possible length of the address, the output dimension of each word in the semantic embedding layer

is set to 768 dimensions, and the overall semantic representation dimensions of each address in address pair are both set to 100. After the semantic representation, two semantic vectors are obtained separately and taken as the input of the next network layer.

Considering the size of the dataset, during the network training process, the batch size of training set is adjusted to 1024. The model also used a two-layer Bi-LSTM network and CNN layer to obtain global context information and local context information. To enhance the difference of two address, ABLC used a dropout structure and probability is set to 0.5. The output is fused into a $X \in \mathbb{R}^{25\times100}$ feature matrix and sent to the self-attention network to get more position-aware information in address descriptions. Finally, two 100-dimensional representation vectors are used as output of the semantic representation to calculate the Manhattan distance. After four layers of full connection compression, the output of last layer is seen as the similarity of the two addresses.

In order to judge the prediction result of the model, we select accuracy, precision, recall, and F1 score as evaluation indicators. The accuracy reflects the model accurate to judge of "similar/dissimilar" and the F1 score reflects the overall performance of the model.

3.3.1. Parameter Experiment Analysis

The relevant parameters in this work are shown in Table 3. In order to verify the stability of the model parameters used in this paper, we have constructed a number of comparative experiments to prove it. The experiment contains multiple models with different batch sizes and learning rates. The model design is shown in the Table 4:

Table 3. Parameter name and corresponding value.

Parameter Name	Parameter Value
epoch	25
batch_size	1024
optimizer	Adam
learning_rate	0.01
dropout	0.5

Table 4. Parameter comparative experiment.

Model No.	Model Setting
1	learning_rate = 0.1
2	learning_rate = 0.001
3	learning_rate = 0.0001
4	batch_size = 512, learning_rate = 0.001
5	batch_size = 1500, learning_rate = 0.1

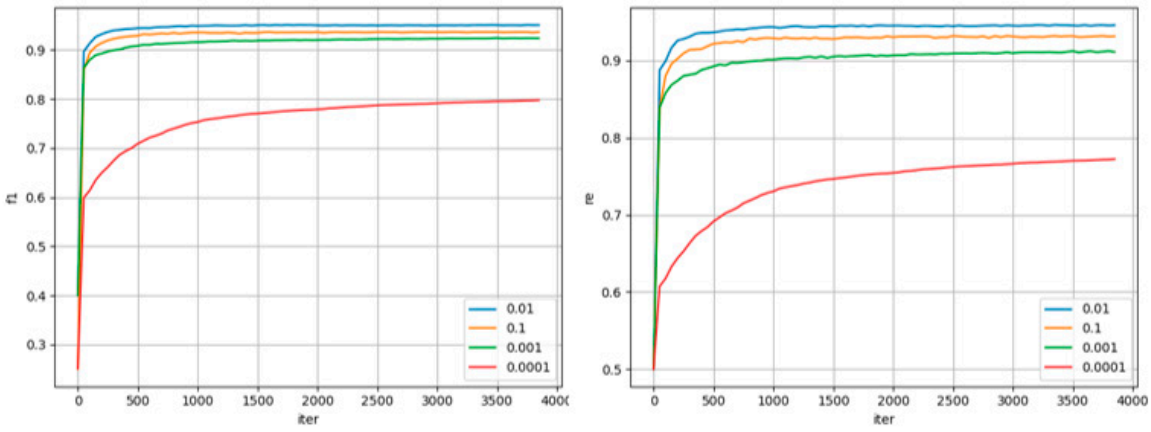
The experiment is carried out under the same training set, and the comparison results of the training indicators are shown in the Table 5:

Table 5. Parameter experiment comparison results.

Model	F1 Score	Accuracy	Recall	Precision
ABLC	0.9504	0.9563	0.9460	0.9552
1	0.9362	0.9439	0.9315	0.9413
2	0.9234	0.9343	0.911	0.9402
3	0.8926	0.8435	0.9798	0.8197
4	0.9263	0.9362	0.9137	0.9436
5	0.9381	0.9458	0.9356	0.9407

The specific analysis of the influence of each parameter element on the model prediction results is as follows:

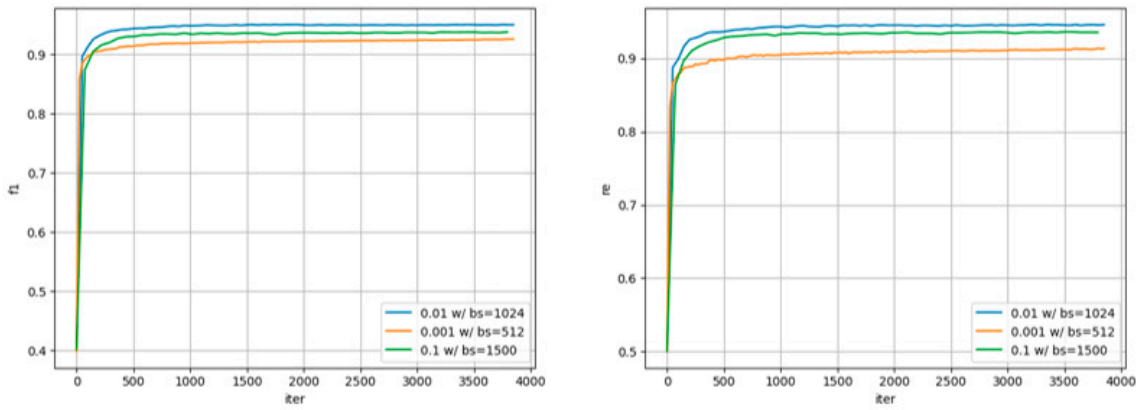
As shown in Figure 5, when the learning rate is set to 0.01, the model converges and achieves a good convergence effect at the same time after 25 epochs. After the learning rate is set to a lower number like 0.0001, the overall average F1 score drops about 15%. The explanation of this experimental result can be expressed as that high learning rate will make the parameter update amplitude large in each iteration of the model. So that the model fails to converge and misses the extreme value during the iteration process. If the learning rate is too small, the convergence rate will be low. Moreover, the minimum point may not be reached and the convergence quality also will be poor.



(**a**) (**b**)

Figure 5. (**a**) F1 score at different learning rates on the training set. (**b**) Recall at different learning rates on the training set.

The experimental results from Figure 6 show that large batches could enable the model to obtain potential information in the datasets more quickly, but the overall gradient update times will be reduced accordingly. Because of that, the model often fails to reach the minimum value, and a small batch will give the model more opportunities to update parameters. Additionally, the result also shows that the adjustment of learning rate can profoundly affect the results of model prediction, and the gap can hardly be closed by changing the batch size.



(**a**) (**b**)

Figure 6. (**a**) F1 score at different learning rate and batch size on the training set. (**b**) Recall at different learning rate and batch size on the training set.

A too large or too small epoch number cannot lead this model to optimal results. When the training rounds are insufficient, the model cannot obtain enough information, and the performance of the trained model is poor. However, at the same time, too high training rounds will cause two problems. First, the model tends to overfit and the results

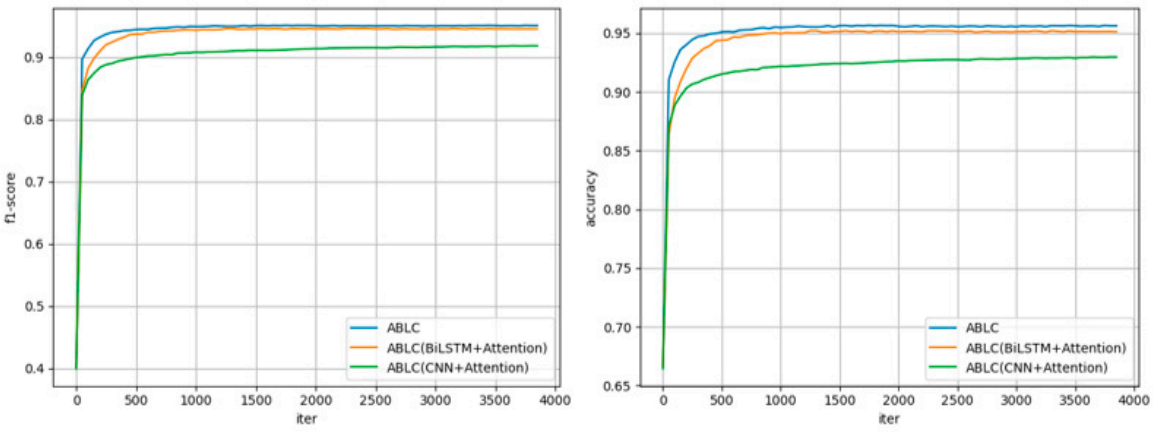
between the training set and the test set are quite different. Secondly, the model may learn a large number of non-representative features and lead the prediction results to a worse direction. According to these conclusions, we select 25 as the best epoch number.

3.3.2. Analysis of Ablation Experiments

In order to verify the stability of the proposed module that integrates context and location information in this study, we designed a number of models with removing partial model structure for comparison experiments. The specific model design and the experimental results of the three models are shown in Table 6. Taking every 50 training samples as a round of iteration, the recall rate and F1 score are recorded. The changes of these two metrics with the training process are shown in Figure 7.

Table 6. Experiment results on ablation analysis.

Model Name	F1	Accuracy	Recall	Precision
ABLC	0.9504	0.9563	0.9460	0.9552
ABLC (BiLSTM + attention)	0.9448	0.9512	0.9428	0.9468
ABLC (CNN + attention)	0.9178	0.9297	0.9020	0.9413



(**a**) (**b**)

Figure 7. (**a**) F1 score of different models on ablation analysis. (**b**) Accuracy of different models on ablation analysis.

As shown in Figure 7, in terms of and F1 score and accuracy, the model proposed in this paper has the best overall performance and stable performance. Compared with the model performance after ablation, the F1 score is improved by about 3–10%. This result proves the overall performance of the model decline when only considering the global context information obtained by Bi-LSTM or the local information obtained by CNN. The decline indicates that the model cannot effectively capture part of the key information in the address. At the same time, the prediction effect of the model has been effectively improved after combining the contextual global information and the local information related to the location. In addition, the F1 score proves that the accuracy of the model is not affected by the ratio of positive and negative examples in the dataset, but the learning ability of the model is indeed enhanced.

3.3.3. Comparative Experiment Analysis

In order to prove that the model proposed in this paper can achieve better results, we select some baseline models as the reference for performance comparison. Considering that the address similarity calculation problem is simplified into a judging problem of "similar" and "non-similar", it can be regarded as a disguised binary classification problem. We compare the approach proposed in this paper with multiple mainstream text classification

approaches, including deep learning methods and machine learning methods. We utilized a random forest and SVM as comparison baseline models (2011) [40]. Additionally, we compared ESIM proposed by Kang (2020) for address semantic matching based on deep learning [41]. In addition, we introduce FastText (2016) and TextRCNN (2015) algorithms as comparison approaches [42,43]. The prediction result is shown in the following table, and the comparison of the change trend of some indicators during the training process is shown in the figure.

From the score shows in the Table 7, it can be concluded that the ABLC model has better improvement from several dimensions compared to other baseline models. From the semantic information point of view, the accuracy improvement of the ABLC model is round 4–10% than other models. The improvement proves that our model does have certain advantages in the classification results.

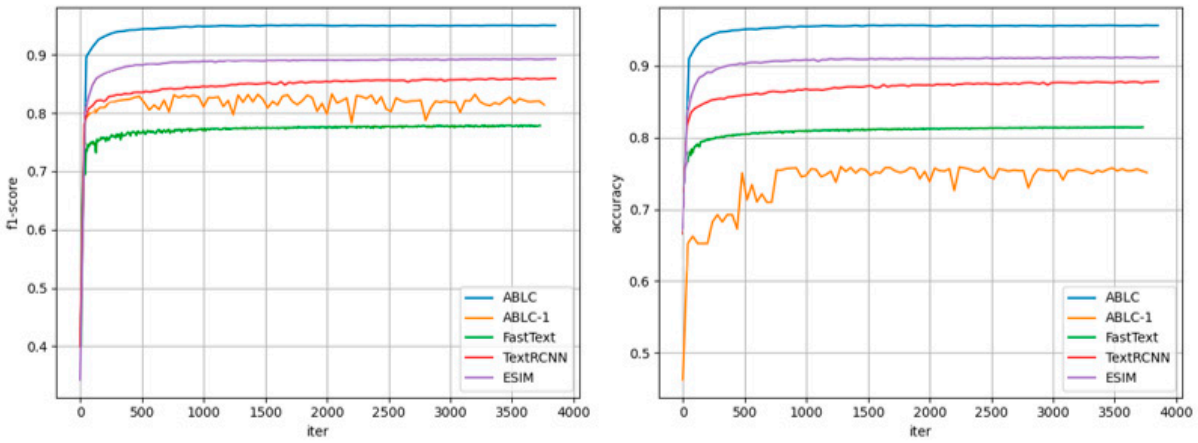
Table 7. Experiment results on comparison experiment.

Model Name	F1 Score	Accuracy	Recall	Precision
ABLC	0.9504	0.9563	0.9460	0.9552
ESIM	0.8992	0.9146	0.9051	0.9020
SVM	0.7267	0.7782	0.7125	0.7662
FastText	0.6763	0.812	0.6132	0.7569
TextRCNN	0.8062	0.8774	0.7733	0.8424
ABLC-1(Xlnet)	0.8142	0.7515	0.8348	0.7947

4. Discussion

Further detailed analysis, in terms of the convergence speed during the training process, FastText has a relatively simple and clear structure, so the convergence is more effective and faster than other models. Due to the fact that TextRCNN uses bidirectional RNN structure to obtain context information, it has certain advantages over FastText on the overall information acquisition, so the result shows TextRCNN has obvious performance strengths compare to FastText. This result presents two conclusions. First of all, the semantic extraction approach that using single sentence has certain advantages over using whole sentence pair to obtain context information as input of the network. Although bidirectional RNN can make up for the deficiencies on calculating distribution of the important words, it will ignore the comparative information from the sentence pair, so it is inferior to the ABLC model in performance. Secondly, the effect difference between the TextRCNN and FastText is not very large, indicating that the additional position-wise information introduced by attention mechanism has a certain improvement, but the effect is relatively limited. This conclusion can be explained as that the address is a special text information based on certain rules. Additionally, then the distribution of semantic information related to its position is often more fixed. Therefore, even though the position information is referred, the semantic gap between addresses is small, and the overall performance improvement of model is not very obvious.

Compared with ABLC model, ABLC-XLNet does not give an effective promotion, and has a certain amount of decline, as shown in Figure 8. As a possible explanation to this, address is a special branch of Chinese string, which contains proper nouns that refer to different places. XLNet has the ability to improve the performance on many downstream tasks, but for Chinese address descriptions, the model may lack of responsiveness to proper nouns of place. Because of this, the embedding performance cannot get a significant enhancement, and the prediction performance metrics of the model have a certain range of oscillations [44,45].



(**a**) (**b**)

Figure 8. (**a**) F1 score of different models compare with ABLC. (**b**) Accuracy of different models compare with ABLC.

Except the ABLC model proposed in this paper, the ESIM model has the best overall performance because ESIM uses local inference and inference synthesis techniques to achieve information extraction and can capture both local and global features of information. However, compared with the ABLC model proposed in this paper, its performance is slightly weaker, probably because the ABLC contrast learning model gives higher weight to the key information in addition to acquiring local and global features of information, and is good at capturing the similarities and differences between inputs via using contrast learning algorithm, thus having better classification capabilities.

A special case shown in the Table 8, even though this paper uses the semantic similarity to do the task of matching address, the ABLC model determines that the relationship between address 1 and address 2 is "not similar", whereas, the subject in address 1 and the subject in address 2 refer to the same building. One possible explanation is the training set does not contain such related information, or the lack of relevant external knowledge to supplement, so that the model cannot find out some subjects with related relationships.

Table 8. A special case of failure determination.

Address 1	Address 2	Similarity
No. 51, Changjiang Middle Road, Fanluoshan Street, Jinghu District, Wuhu City	Human Resources Security Bureau, Jinghu District, Wuhu City, Anhui Province	0

5. Conclusions

Aiming at the current problem of unrecognizable redundant information in Chinese addresses, this paper proposes a contrast learning address matching model based on attention-Bi-LSTM-CNN network. The model first extracts the address elements using Trie syntax tree according to the characteristics of Chinese addresses, followed by using Bi-LSTM to obtain the sentence-level information of addresses, as well as using CNN to obtain the word-level information in addresses, and combining with the attention mechanism to focus on the key information in addresses and assign higher weights. After the complete extraction of semantic information of the addresses, the final comparison of address similarity is achieved using the Manhattan distance. In addition, data augmentation is applied to construct the address augmentation dataset, which is combined with the dropout strategy to achieve data augmentation. The comparison with various benchmark models shows that our proposed model has better performance. For the next step we will consider that for one thing to study the association between addresses and geographic entities, for another thing to try to introduce information such as geographic information maps to enhance the

accuracy of recognition. In addition, the generalization ability to unknown address needs to do further research.

Author Contributions: J.C. (Jian Chen) and G.C. proposed the original idea and conducted the organization of the content. J.C. (Jianpeng Chen) carried out the experiments and analysis the results. J.C. (Jian Chen) wrote this paper. X.S. and J.M. put forward some suggestions on the modifications of this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Key Research and Development Plan of Anhui Province in 2021(202104a05020071).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, B.H.Y.; Waddell, P.; Wang, L.; Pendyala, R.M. Reexamining the influence of work and nonwork accessibility on residential location choices with a microanalytic framework. *Environ. Plan. A* **2010**, *42*, 913–930. [CrossRef]
2. Zhang, Z.; Hadjieleftheriou, M.; Ooi, B.C.; Srivastava, D. Bed-tree: An all-purpose index structure for string similarity search based on edit distance. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, Indianapolis, IN, USA, 6–10 June 2010; pp. 915–926.
3. Levenshtein, V.I. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Phys. Doklady* **1966**, *10*, 707.
4. Bilenko, M.; Mooney, R.J. Adaptive Duplicate Detection Using Learnable String Similarity Measures. In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; Association for Computing Machinery: New York, NY, USA, 2003; pp. 39–48.
5. Jaccard, P. Nouvelles Recherches Sur la Distribution Florale. *Bull. Soc. Vaudoise Sci. Nat.* **1908**, *44*, 223–270.
6. Banerjee, S.; Pedersen, T. *The Design, Implementation, and Use of the Ngram Statistics Package*; Springer: Berlin/Heidelberg, Germany, 2003.
7. Li, B.; Han, L. Distance weighted cosine similarity measure for text classification. In Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning, Salamanca, Spain, 10–12 September 2013; Springer: Berlin/Heidelberg, Germany, 2013; pp. 611–618.
8. Kang, M.; Du, Q.; Wang, M. A New Method of Chinese Address Extraction Based on Address Tree Model. *Acta Geod. Cartogr. Sin.* **2015**, *44*, 99–107.
9. Laferty, J.D.; McCallum, A.; Pereira, F.C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In Proceedings of the 18th International Conference on Machine Learning, San Francisco, CA, USA, 18–24 July 2001; Morgan Kaufmann Publishers Inc.: Burlington, MA, USA, 2001; pp. 282–289.
10. Rabiner, L.; Juang, B. An introduction to hidden Markov models. *IEEE ASSP Mag.* **1986**, *3*, 4–16. [CrossRef]
11. Sun, Z.; Qiu, A.G.; Zhao, J.; Zhang, F.; Zhao, Y.; Wang, L. Technology of fuzzy Chinese-geocoding method. In Proceedings of the 2013 International Conference on Information Science and Cloud Computing, Guangzhou, China, 7–8 December 2013; IEEE Computer Society: Washington, DC, USA, 2013; pp. 7–12.
12. Xueying, Z.; Guonian, L.; Boqiu, L.; Wenjun, C. Rule-based approach to semantic resolution of Chinese addresses. *J. Geo-Inf. Sci.* **2010**, *12*, 9–16.
13. Cangxiu, C.; Bin, Y. A rule-based segmenting and matching method for fuzzy Chinese addresses. *Geogr. Geo-Inf. Sci.* **2011**, *27*, 26–29.
14. Tian, Q.; Ren, F.; Hu, T.; Liu, J.; Li, R.; Du, Q. Using an optimized Chinese address matching method to develop a geocoding service: A case study of Shenzhen, China. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 65. [CrossRef]
15. Zhang, H.; Ren, F.; Li, H.; Yang, R.; Zhang, S.; Du, Q. Recognition Method of New Address Elements in Chinese Address Matching Based on Deep Learning. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 745. [CrossRef]
16. Comber, S.; Arribas-Bel, D. Machine learning innovations in address matching: A practical comparison of word2vec and CRFs. *Trans. GIS* **2019**, *23*, 334–348. [CrossRef]
17. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Association for Computational Linguistics: Stroudsburg, PA, USA, 2014; pp. 1746–1751.
18. Santos, R.; Murrieta-Flores, P.; Calado, P.; Martins, B. Toponym matching through deep neural networks. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 324–348. [CrossRef]
19. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent convolutional neural networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; p. 29.

20. He, J.; Li, X.; Yao, Y.; Hong, Y.; Jinbao, Z. Mining transition rules of cellular automata for simulating urban expansion by using the deep learning techniques. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2076–2097. [CrossRef]
21. Ding, R.; Chen, Z. RecNet: A deep neural network for personalized POI recommendation in location-based social networks. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 1631–1648. [CrossRef]
22. Karimzadeh, M.; Pezanowski, S.; MacEachren, A.M.; Wallgrün, J.O. GeoTxt: A scalable geoparsing system for unstructured text geolocation. *Trans. GIS* **2019**, *23*, 118–136. [CrossRef]
23. Du, P.; Bai, X.; Tan, K.; Xue, Z.; Samat, A.; Xia, J.; Li, E.; Su, H.; Liu, W. Advances of four machine learning methods for spatial data handling: A review. *JGSA* **2020**, *4*, 1–25.
24. Grekousis, G. Artificial neural networks and deep learning in urban geography: A systematic review and meta-analysis. *Computers Environ. Urban Syst.* **2019**, *74*, 244–256. [CrossRef]
25. Young, T.; Hazarika, D.; Poria, S.; Cambria, E. Recent trends in deep learning based natural language processing. *IEEE Comput. Intell. Mag.* **2018**, *13*, 55–75. [CrossRef]
26. Zhang, L.; Wang, S.; Liu, B. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Mining Knowl. Discov.* **2018**, *8*, e1253. [CrossRef]
27. Rumelhart, D.E.; Hintont, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
28. Klein, T.; Nabi, M. Contrastive self-supervised learning for commonsense reasoning. *arXiv* **2020**, arXiv:2005.00669.
29. Yang, Z.; Cheng, Y.; Liu, Y.; Sun, M. Reducing word omission errors in neural machine translation: A contrastive learning approach. *Proc. ACL* **2019**, 6191–6196.
30. Meng, Y.; Xiong, C.; Bajaj, P.; Tiwary, S.; Bennett, P.; Han, J.; Song, X. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *arXiv* **2021**, arXiv:2102.08473.
31. Wu, Z.; Wang, S.; Gu, J.; Khabsa, M.; Sun, F.; Ma, H. Clear: Contrastive learning for sentence representation. *arXiv* **2020**, arXiv:2012.15466.
32. Karim, F.; Majumdar, S.; Darabi, H.; Chen, S. LSTM fully convolutional networks for time series classification. *IEEE Access* **2017**, *6*, 1662–1669. [CrossRef]
33. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M. A guide to convolutional neural networks for computer vision. *Synth. Lect. Computer Vision* **2018**, *8*, 1–207. [CrossRef]
34. Shaw, P.; Uszkoreit, J.; Vaswani, A. Self-attention with relative position representations. *arXiv* **2018**, arXiv:1803.02155.
35. Shijie, J.; Ping, W.; Peiyi, J.; Siping, H. Research on data augmentation for image classification based on convolution neural networks. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 January 2017; IEEE: New York, NY, USA, 2017; pp. 4165–4170.
36. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
37. Sakamoto, T.; Yokozawa, M.; Toritani, H.; Shibayama, M.; Ishitsuka, N.; Ohno, H. A crop phenology detection method using time-series MODIS data. *Remote Sens. Environ.* **2005**, *96*, 366–374. [CrossRef]
38. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
39. Bouthillier, X.; Konda, K.; Vincent, P.; Memisevic, R. Dropout as data augmentation. *arXiv* **2015**, arXiv:1506.08700.
40. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
41. Lin, Y.; Kang, M.; Wu, Y.; Du, Q.; Liu, T. A deep learning architecture for semantic address matching. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 559–576. [CrossRef]
42. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. Fasttext. zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
43. Zhou, X.; Chen, X.; Song, J.; Zhao, G.; Wu, J. Team Cat-Garfield at TREC 2018 Precision Medicine Track. In Proceedings of the TREC, Gaithersburg, MD, USA, 14–16 November 2018.
44. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.X. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5754–5764.
45. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.