International Journal of Geo-Information

MDPI

Article

Bidirectional Gated Recurrent Unit Neural Network for Chinese Address Element Segmentation

Pengpeng Li [1,2], An Luo [2,3,*], Jiping Liu [1,2], Yong Wang [1,2], Jun Zhu [1], Yue Deng [4] and Junjie Zhang [3]

1 Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 610031, China; lipengpeng@my.swjtu.edu.cn (P.L.); zhujun@swjtu.edu.cn (J.Z.)
2 Research Center of Government GIS, Chinese Academy of Surveying and Mapping, Beijing 100830, China; liujp@casm.ac.cn (J.L.); wangyong@casm.ac.cn (Y.W.)
3 School of Marine Technology and Geomatics, Jiangsu Ocean University, Lianyungang 222005, China; 2018224060@jou.edu.cn
4 School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China; 2018102050017@whu.edu.cn
* Correspondence: luoan@casm.ac.cn

Received: 10 September 2020; Accepted: 21 October 2020; Published: 26 October 2020

Abstract: Chinese address element segmentation is a basic and key step in geocoding technology, and the segmentation results directly affect the accuracy and certainty of geocoding. However, due to the lack of obvious word boundaries in Chinese text, the grammatical and semantic features of Chinese text are complicated. Coupled with the diversity and complexity in Chinese address expressions, the segmentation of Chinese address elements is a substantial challenge. Therefore, this paper proposes a method of Chinese address element segmentation based on a bidirectional gated recurrent unit (Bi-GRU) neural network. This method uses the Bi-GRU neural network to generate tag features based on Chinese word segmentation and then uses the Viterbi algorithm to perform tag inference to achieve the segmentation of Chinese address elements. The neural network model is trained and verified based on the point of interest (POI) address data and partial directory data from the Baidu map of Beijing. The results show that the method is superior to previous neural network models in terms of segmentation performance and efficiency.

Keywords: Chinese address element; Bi-GRU neural network; address segmentation; Viterbi

1. Introduction

With the rapid development of technologies such as the internet and big data and the emergence of location-based services [1], the public's demand for location data is increasing rapidly. According to statistics, approximately 70% of the world's web pages contain location information. However, most of this information is expressed in the form of text [2], which leads to a lack of spatial coordinate information. Therefore, we urgently need tools to convert textual addresses into spatial coordinates and to provide the public with location-based services and big data analysis capabilities based on spatial locations. To date, geocoding is the most effective and commonly used method to establish a connection between textual addresses and spatial coordinates [3].

Geocoding is a coding method based on spatial positioning technology, which provides a way to map unstructured text addresses to geographic coordinates [4]. We can use geocoding technology to establish the relationship between nonspatial information and spatial information in the reference scope of geospatial space. Therefore, geocoding has a wide range of applications in the field of urban spatial positioning and spatial analysis, such as disaster emergency response and disaster

management [5], disease investigation and prevention [6], and crime scene location [7]. It realizes the space coordinate conversion process of textual addresses through address element segmentation, address standardization, address matching, and space positioning [8]. Among these steps, Chinese address element segmentation refers to the process of splitting unstructured Chinese addresses into address elements [9]. The segmentation process references the process of Chinese word segmentation, which divides Chinese addresses into words and marks them according to the expression characteristics of address elements to achieve the purpose of address element segmentation. This is a basic and critical step in geocoding, and the segmentation results directly affect the accuracy and certainty of geocoding [10]. Unfortunately, due to the diversity and complexity in the way Chinese addresses are expressed, Chinese address element segmentation is an extremely complicated process. The specific reasons are as follows:

(1) Due to China's vast land area, numerous nationalities, and large geographic and cultural differences, Chinese addresses exist in a state of disorder, lacking uniform standards and causing confusion. Therefore, to date, China has not established an authoritative and reliable unified Chinese address naming standard that covers the whole country. Currently, with the advent of the big data era, Chinese address expression has become more complex and diverse [11]. This exacerbates the abovementioned disorder and confusion. This situation complicates the segmentation of Chinese address elements.
(2) Unlike English sentences, where spaces are present as natural dividing lines between words, Chinese sentences can be defined only by various punctuation marks. The Chinese address segmentation step is much more difficult than for most other languages where spaces are natural delimiters.
(3) Due to China's unique place-name and address management system and the diversity and complexity of Chinese addresses [12], government departments at all levels, planning departments, and transportation departments have lacked reliable, standard, and unified address information resources for a long time. For example, in the address naming management of Beijing, the naming of administrative divisions at all levels is managed by the Civil Affairs Department of Beijing, the naming of roads is managed by the Traffic and Planning Department of Beijing, and the naming plates for house, buildings, units, and household (rooms) plates is managed by the Beijing Public Security Bureau. It is difficult to achieve unified address expressions due to the different address naming standards and management oversight of multiple departments [13]. This also makes the management of addresses more difficult, and the accuracy of segmentation for different types of address elements varies greatly.

In recent years, with the rapid development of deep learning and natural language processing technologies, Chinese word segmentation based on neural networks has also achieved satisfactory results. In particular, the recurrent neural network (RNN) and its improved network have achieved substantial breakthroughs in the accuracy and segmentation efficiency of Chinese word segmentation [14,15]. Among them, the gated recurrent unit (GRU) neural network has the characteristics of a simple network structure and easy training processes [16]. Therefore, referring to the Chinese word segmentation model based on a neural network, this paper proposes a method of Chinese address element segmentation based on the bidirectional GRU (Bi-GRU) neural network. This method first performs Chinese word segmentation on the input Chinese address sequence. Secondly, the word feature vector is generated through text vectorization as the input of the neural network. Then, each tag feature vector is generated through the neural network. Finally, the Viterbi algorithm is used for tag inference and word segmentation. This study provides basic support for the conversion of spatial information and nonspatial information and the application of location-based services. At the same time, this study provides a new idea and a new method for address element segmentation.

The remainder of this paper is organized as follows: Section 2 reviews the related work and introduces the gated recurrent unit (GRU) neural network, Jieba word segmentation, and Chinese

address elements. Section 3 introduces the segmentation model and segmentation process of Chinese address elements. Section 4 introduces the experimental data, evaluation criteria, and experimental results. Section 5 discusses the experimental results and explains related issues. Section 6 concludes with a summary of this paper and introduces the next research direction.

2. Related Works

2.1. Previous Works

Chinese address element segmentation is one of the applications of Chinese word segmentation in place-name addresses. However, because Chinese expressions have no obvious word boundaries, Chinese word segmentation is not as easy as English and Spanish word segmentation [17]. Second, Chinese, like other languages, presents the problem of polysemy due to different contexts, which makes Chinese word segmentation more difficult. With the continuous development of natural language-processing technology, substantial breakthroughs have been made in Chinese word segmentation technology. Currently, the most commonly used Chinese word segmentation methods are as follows:

(1) The dictionary-based string-matching method matches the strings to be segmented with a dictionary library one by one according to a certain strategy. According to the different directions of string matching, the matching strategies can be divided into three types: forward maximum matching, reverse maximum matching, and bidirectional maximum matching. This method is easy to implement, has high segmentation accuracy, and is fast, so it is the most widely used. Many scholars have improved the word segmentation performance of this method by improving the dictionary structure. For example, Wang et al. [18] used the double-array trie tree algorithm to preferentially process nodes with more branch nodes to improve the efficiency of the dictionary search and reduce the necessary data storage space. Li et al. [19] and Mo et al. [20] used a double-character hash indexing mechanism to improve the accuracy of word segmentation and shorten the segmentation time. However, the dictionary-based string-matching method fails to address the identification of ambiguous words and unregistered words.

(2) The word segmentation method is based on semantic rules. The main idea of this method is to use the principles of word formation, part-of-speech features, and semantic databases to construct specific rules based on semantics. For example, Li et al. [21] proposed a Chinese address segmentation method based on a combination of rules and statistics. This method uses statistical methods to extract address information from the original address corpus and uses a rule-based method to segment Chinese addresses. Zhang et al. [22] constructed an address element feature database based on different types of address elements and performed Chinese address segmentation based on a database of these character features. The advantage of this method is that it is strongly pertinent to and has high segmentation accuracy for certain types of words. However, artificial tagging and feature extraction are required, and there are some problems, such as sparse features and poor adaptability.

(3) The word segmentation method is based on conventional machine learning. The main idea of this method is to train a model on the tagging of characters and consider the frequency and the contextual information of words. Therefore, this method offers a favorable learning ability and performs well regarding the recognition of ambiguous words and unregistered words. Among the most commonly used models are the maximum entropy model [23,24], hidden Markov model (HMM) [25,26], and conditional random field (CRF) model [27,28]. The above three models require large amounts of artificial tagging data for training and are easily affected by manual feature selection.

(4) The word segmentation method is based on a neural network. The main idea is to simulate the operation of the human brain, distribute processing, and establish a numerical calculation model. This process stores the implicit method for understanding word segmentation into the neural network and achieves the correct word segmentation result through the self-learning and training

of internal weights. The neural network can automatically learn features and avoid the limitations of conventional manual feature extraction. For example, Chen et al. [29] used a long short-term memory (LSTM) neural network, because a general neural network cannot learn long-distance dependence relationships. Chen et al. [30] proposed a gated recursive neural network (GRNN) that fuses the complex features of contextual characters and a supervised layered training method to achieve a better word segmentation model.

All the above Chinese word segmentation methods or Chinese address element segmentation methods take a single character as the minimum segmentation unit. They recombine character sequences into word sequences or address elements according to certain specifications. Since Chinese characters represent the smallest semantic unit that constitutes Chinese words, Chinese word segmentation can only be based on characters as the smallest segmentation unit [31]. This also makes the Chinese word segmentation method based on the neural network use characters as the model input. However, Chinese address elements are different from Chinese words, which usually consist of one or more words. For example, if we use "莲花池 (Lianhuachi)" and "西路(West Road)" to directly represent address elements, the practical meaning is lost. Only a combination of the two words forms a complete address element. These words that constitute the address elements are exactly the word sequences generated by Chinese word segmentation. Therefore, based on Chinese word segmentation, this paper proposes a method of Chinese address element segmentation based on a Bi-GRU neural network.

2.2. GRU Neural Network

An RNN is a kind of neural network with a memory function that is suitable for sequence data modeling. However, classic RNNs experience the issues of gradient explosion and gradient disappearance, and these algorithms cannot handle long-distance dependence problems. For this reason, in 1997, Hochreiter et al. [32] proposed an LSTM neural network, a special RNN that performs better on longer sequences. Chao et al. [33] proposed the GRU neural network based on LSTM neural networks. They combined the forget gate and input gate in an LSTM neural network into a single "update gate" and combined the cell state and hidden state. The GRU neural network is a circular network structure that determines the current output information through the input information at the current moment and the output information at the previous moment. Therefore, the output information at each moment in the GRU neural network depends on past information. Therefore, its chain attribute is closely related to the sequential labeling problem and is applied to the Chinese word segmentation task.

A GRU neural network has two control gates, a reset gate and an update gate, as shown in Figure 1. The reset gate determines how much information needs to be forgotten in the hidden state of the previous moment. When the value of the reset gate is closer to 0, the information of the previous moment is forgotten. When the value is closer to 1, the hidden information of the previous moment is retained in the current memory information. The update gate determines how much information in the hidden state at the previous moment will be brought into the current hidden state. When the value of the update gate is closer to 0, the information in the hidden state at the previous moment is forgotten. When the value is closer to 1, the information is retained in the current hidden state.

In Figure 1, z_t is the update gate, r_t is the reset gate, $\tilde{h}_t$ is the candidate hidden state of the currently hidden node, h_t is the current hidden state, x_t is the input of the current neural network, and h_{t-1} is the hidden state at the previous moment. The detailed calculation formula is as follows:

$$z_t = \sigma(w_{zx}x_t + u_{zh}h_{t-1}) \tag{1}$$

$$r_t = \sigma(w_{rx}x_t + u_{rh}h_{t-1}) \tag{2}$$

$$\tilde{h}_t = \tan(w_{hx}x_t + r_t \odot u_{hh}h_{t-1}) \tag{3}$$

$$h_t = (1 - z_t) \odot \tilde{h} + z_t \odot h_{t-1} \tag{4}$$

where σ is the activation function *sigmoid*, which ranges from 0 to 1, $\odot$ is the Hadamard product of the matrix, w and u are the weight matrices that need to be learned, and z_t and r_t range from 0 to 1. In Chinese address element segmentation, the reset gate acts on h_{t-1} to record all important information, also known as memory content. As shown in Formula (3), the reset gate is composed of vectors from 0 to 1. Therefore, after the Hadamard product is obtained, the reset gate determines how much of the hidden state in the previous time should be forgotten in the current memory content. The current input information is then added and placed in the activation function. Therefore, $\tilde{h}_t$ records all important information through the reset gate and input information. The update gate determines the currently hidden state h_t by acting on h_{t-1} and $\tilde{h}_t$ and passes it to the next unit. As shown in Formula (4), the first term through $(1 - z_t)$ determines what information needs to be forgotten, and the corresponding information in the memory content is updated at this time. The second term of the formula determines how much information of h_{t-1} is retained in the current hidden state. Therefore, h_t decides to collect the required information at $\tilde{h}_t$ and h_{t-1} through the update gate.

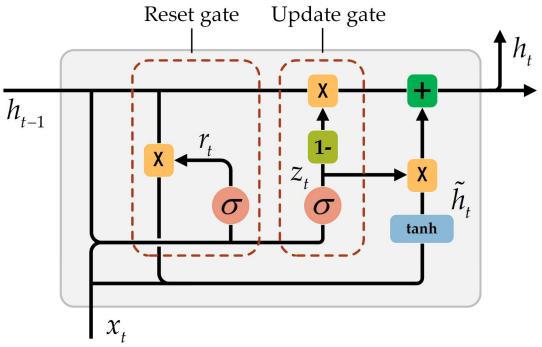

Figure 1. Gated recurrent unit (GRU) neural network unit structure. x_t is the current input, z_t is the update gate, r_t is the reset gate, $\tilde{h}_t$ is the candidate hidden state of the currently hidden node, h_t is the current hidden state, x_t is the input of the current neural network, and h_{t-1} is the hidden state at the previous moment. σ is the activation function *sigmoid*. $\tilde{h}_t$ records all important information through the reset gate and input information.

Each hidden layer in the neural network has a separate update gate and reset gate. The layers produce different dependent relationships based on the current input information and the information from the previous moment.

2.3. Jieba Word Segmentation

With the development of natural language processing technology, increasingly numerous Chinese word segmentation tools have become available. Jieba word segmentation is widely used due to its active community, simplicity, and rich functionality. This method achieves word segmentation by first scanning the sentence with an efficient word map based on the prefix dictionary and then generating all possible word formations in the sentence. Next, a directed acyclic graph is constructed according to the segmentation position. Then, dynamic programming is used to find the path with the maximum probability, and the maximum segmentation combination based on the word frequency is found. Finally, for unregistered words, an HMM based on the ability of Chinese characters to form words is used for segmentation.

2.4. Chinese Address Element

Chinese addresses necessitate abstract coding methods with a description of the spatial location through the organization of natural language and address expression models [8]. This can be expressed as:

$$A = \left\{ x_i \in A | P\left(x_i, x_j\right) \neq \varnothing, x_i \neq x_j \right\} \tag{5}$$

where A is the Chinese address, x_i is the Chinese address element, and $P(x_i, x_j)$ is the spatial constraint relationship between the address elements and cannot be empty. The spatial constraint relationship refers to the topological constraint between the geographical entities corresponding to the address elements.

Chinese address elements, as the smallest semantic unit in a Chinese address, describe a certain area or geographic entity. According to the Chinese address structure and address mode, address elements can be divided into three types. The administrative division elements mainly include the five administrative divisions of province, city, county (district), town, and village (community); the detail address elements can include roads, house numbers, communities, building numbers, unit numbers, etc.; and supplemental address elements can include the names of various institutions or words that indicate spatial orientation. Each type has specific word-formation rules, which can be expressed as "proprietary words plus general words" [22]. Among them, general words are "feature words" that indicate the level or type of address elements, usually at the end of the address elements. Proprietary words are the remaining part of the address element after removing the general words. For example, "Lianhuachi" in "Lianhuachi West Road" is a proprietary word, and "West Road" is a general word. Therefore, in the segmentation of Chinese address elements, using words as the smallest segmentation unit conforms to the word-formation rules of address elements. This article summarizes the "feature words" of various types of address elements through the statistical analysis of a large amount of address data, as shown in Table 1.

Table 1. Feature words for address element types.

Type	Feature Words
Administrative division	Province/City/District/County/Town/Village/Community/etc.
Detail address	Road/Street/Alley/Hutong/No./Yard/Auxiliary Road/etc.
Supplemental address	Square/Building/Hotel/Park/Intersection/East/Near/Inside/etc.

3. Chinese Address Element Segmentation

The Chinese word segmentation task is usually considered a sequence tagging task. This task uses corresponding tags for each character in Chinese sentences and performs word segmentation based on these tags [34]. Currently, the mainstream tag sets in sequence tagging tasks are two-tag sets, three-tag sets, four-tag sets, and six-tag sets, as shown in Table 2.

Table 2. Four cited mainstream tagging methods.

Tagging Method	Tag Set
two-tag set	{B, I}
three-tag set	{B, I, O}
four-tag set	{B, M, E, S}
six-tag set	{B, M1, M2, M, E, S}

B represents the beginning of the word, I represents other parts except the beginning of the word, M, M1, and M2 represent the middle of the word, E represents the end of the word, S and O represent single-character words.

Generally, the more characters in the tag sets, the more accurate the tagging is. However, words consisting of five or more characters in short texts similar to Chinese addresses account for very few words. Common Chinese words are composed of four or fewer characters [35]. Therefore, it is difficult for tags M1 and M2 to play a role in the tagging of the six-tag set, and many unnecessary feature functions are generated [36]. Therefore, in this article, a four-tag set is chosen to label the address elements. Among them, tag B indicates that the word is the beginning word of the address element, tag M indicates that the word is the middle word of the address element, tag E indicates that the word is the end word of the address element, and tag S indicates that the word is a complete

address element. Additionally, the transition weight matrix A and the initial tag vector π are used to better represent the dependency between tags and the initialization of tags.

In this paper, based on the Chinese word segmentation neural network model, the Bi-GRU neural network is used for the segmentation of Chinese address elements. The model mainly includes the following four steps, as shown in Figure 2.

(1) Chinese word segmentation: The Chinese address sequence is divided into several independent words by using the Chinese word segmentation tools and the characteristics of the address elements. These independent words are used as the input to text vectorization.
(2) Text vectorization: The feature vector representation of each word is found through a lookup table, and these feature vectors are used as the input of the neural network.
(3) Neural network: The Bi-GRU neural network is used to generate the tag feature representation of each word and serve as the input of the tag inference.
(4) Tag inference: The Viterbi algorithm is used to find the maximum possible tag combination as the final tag sequence.

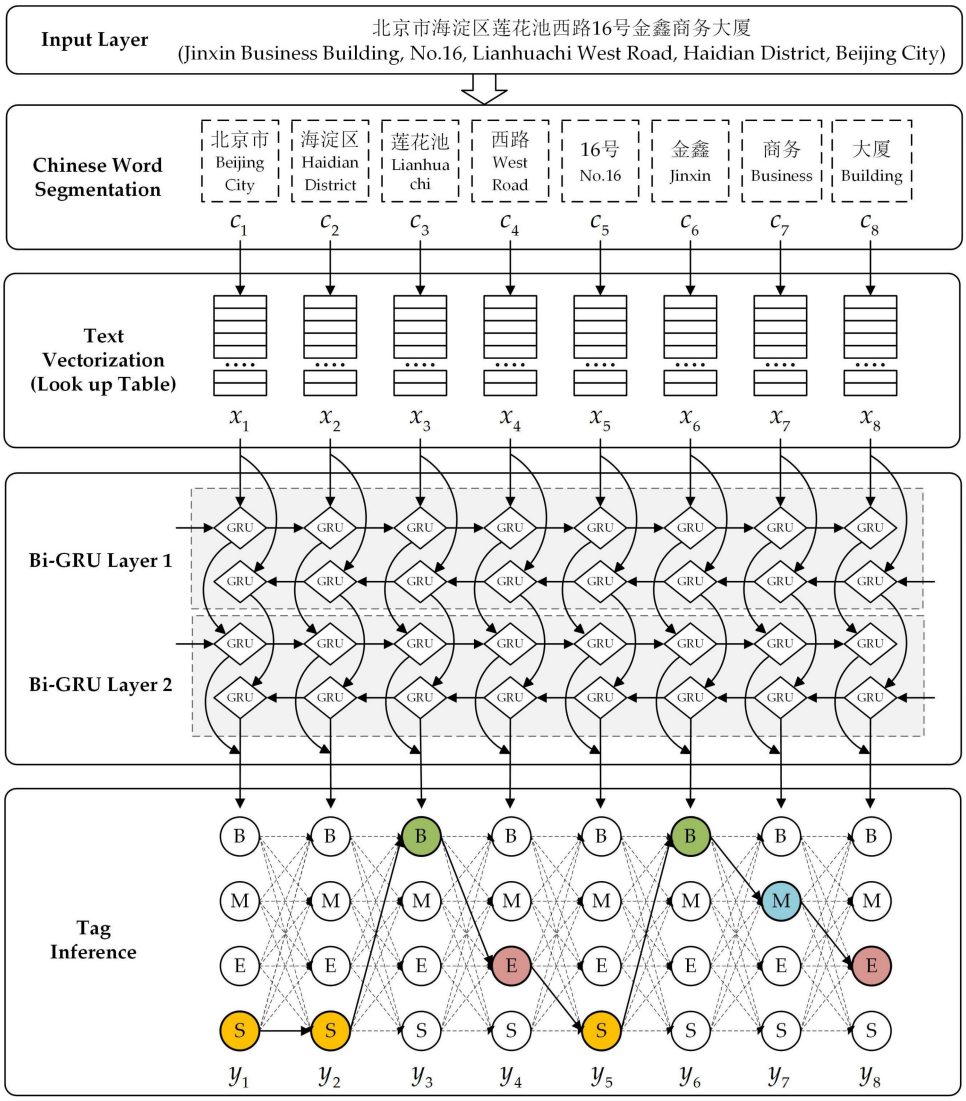

Figure 2. Chinese address element segmentation model based on the bidirectional GRU (Bi-GRU) neural network. The colored tag represents the final tag of the input word. B represents the beginning of the address element, M represents the middle of the address element, E represents the end of the address element, and S represents the single word address element.

3.1. Chinese Word Segmentation

In addition to Chinese characters in Chinese addresses, some non-Chinese characters, such as numbers, English letters, and special characters, often appear. Therefore, based on Jieba word segmentation, the following treatments are carried out for these characters:

(1) Numbers generally indicate information such as the house number, building number, unit number, floor number, and room number. The processing method combines the number with the prefix and suffix to modify it into one word, such as "甲2号 (No.2 A)" in "前门大街甲2号 (No.2 A Qianmen Street)" as a word.

(2) English letters are generally combined with prefixes and suffixes to indicate a specific geographic entity, such as "望京SOHO中心 (Wangjing SOHO Center)". The processing method combines all adjacent English letters into one word.

(3) Special characters are generally expressed as additional descriptive information or some default information. In "东城区天坛路1号(天坛公园北门) (No. 1 Tiantan Road, Dongcheng District (The north gate of the temple of Heaven Park))", the information in brackets is the additional description of the previous address information. In the processing method, for special characters that bracket information, the brackets and the information inside are deleted, and special symbols other than brackets are treated as words.

3.2. Text Vectorization

Neural network model word segmentation first requires a feature vector of a specific dimension to represent characters. This feature vector can describe the semantic relevance between characters and becomes the input of the neural network as a character feature. We perform Chinese word segmentation on all addresses in the training dataset and generate a dictionary D of size $|D|$. Then, we use the bidirectional encoder representation from the transformers (BERT) [37] model for each word $c \in D$ to generate the corresponding feature vector $V_c \in R^d$, where d is the dimension of the feature vector (the hyperparameters we need to set). Then, we stack the feature vectors of all the words into an embedding matrix $M \in R^{d \times |D|}$.

Formally, assuming a given word sequence $c_{[1,n]}$, for each word $c_i (1 \le i \le n)$ with an associated index in the embedding matrix, the corresponding feature vector $V_c \in R^d$ can be retrieved through a lookup table. The text vectorization layer in the model can be regarded as a simple projection layer, where the feature vector representation of each word can be retrieved in the lookup table through its index.

3.3. Bi-GRU Neural Network

A Bi-GRU neural network is a GRU neural network enhanced with a two-layer structure. This two-layer structure provides the output layer with the complete contextual information of the input information at every moment. The basic idea of the Bi-GRU neural network is that the input sequence is passed through a forward neural network and a backward neural network, and then, the outputs of the two are connected in the same output layer. Figure 3 shows the two-layer Bi-GRU neural network used in this article in a time series expansion form.

Among them, in the Bi-GRU neural network of each layer, the forward layer calculates the output of the hidden layer at each time from forward to backward, and the backward layer calculates the output of the hidden layer at each time from backward to forward. The output layer superimposes and normalizes the output results of the forward layer and backward layer at each moment:

$$\overrightarrow{h_t^1} = f\left(w_{\underset{xh^1}{\rightarrow}} x_t + w_{\underset{h^1h^1}{\rightarrow\rightarrow}} \overrightarrow{h_{t-1}^1} + b_{\underset{h^1}{\rightarrow}}\right) \tag{6}$$

$$\overleftarrow{h_t^1} = f\left(w_{\underset{xh^1}{\leftarrow}} x_t + w_{\underset{h^1h^1}{\leftarrow\leftarrow}} \overleftarrow{h_{t+1}^1} + b_{\underset{h^1}{\leftarrow}}\right) \tag{7}$$

$$\overrightarrow{h_t^2} = f\left(w_{\substack{\rightarrow\rightarrow\\ h^1h^2}}\overrightarrow{h_t^1} + w_{\substack{\rightarrow\rightarrow\\ h^2h^2}}\overrightarrow{h_{t-1}^2} + b_{\substack{\rightarrow\\ h^2}}\right) \tag{8}$$

$$\overleftarrow{h_t^2} = f\left(w_{\substack{\leftarrow\leftarrow\\ h^1h^2}}\overleftarrow{h_t^1} + w_{\substack{\leftarrow\leftarrow\\ h^2h^2}}\overleftarrow{h_{t+1}^2} + b_{\substack{\leftarrow\\ h^2}}\right) \tag{9}$$

$$y_t = g\left(w_{\substack{\rightarrow\\ h^2y}}\overrightarrow{h_t^2} + w_{\substack{\leftarrow\\ h^2y}}\overleftarrow{h_t^2} + b_y\right) \tag{10}$$

where $\overrightarrow{h_t^1} \in R^H$ and $\overrightarrow{h_t^2} \in R^H$ are the output vectors of the hidden layer of the forward layer in the first and second layers of the Bi-GRU neural network at time t, H is the number of units in the GRU cell, $\overleftarrow{h_t^1} \in R^H$ and $\overleftarrow{h_t^2} \in R^H$ are the output vectors of the hidden layer of the backward layer in the first and second layers of the Bi-GRU neural networks at time t, $y_t \in R^T$ is the score of the corresponding word on each label at time t, T is the number of tags, x_t is the neural network input at time t, $f(\cdot)$ is GRU neural network processing, $g(\cdot)$ is the activation function, where $g(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}}$, and w and b are the weight matrices that need to be learned.

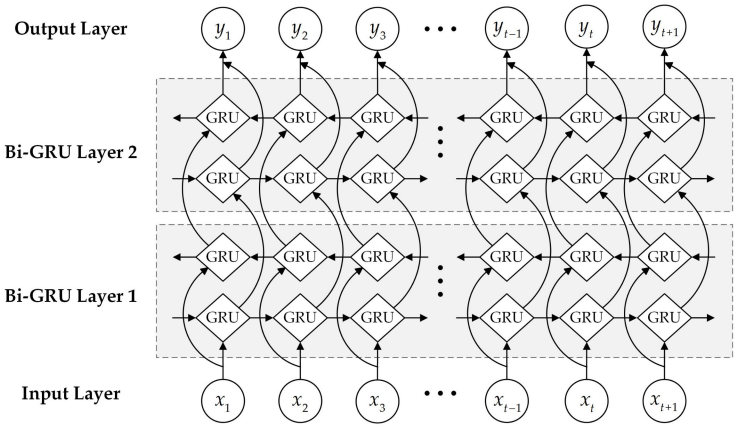

Figure 3. The Bi-GRU neural network unit structure.

3.4. Tag Inference

The tag inference is based on the tag score and the tag transition matrix to find the most likely group of all tag combination sequences. As the cost of an exhaustive search is extremely high, we use Viterbi-based thinking for searching. We find the most likely sequence combination by calculating the local optimal probability and backtracking the backward pointer.

For an input sequence $c_{[1,n]}$ of the neural network model defined at time $t(1 \le t \le n)$, the local optimal probability of tag i is $\delta_t(i)$, and the backward pointer is $\varphi_t(i)$. The calculation formula is as follows:

When $t = 1$:

$$\delta_t(i) = \pi_i + y_{(1,i)} \tag{11}$$

When $1 < t \le n$:

$$\delta_t(i) = \max_j\left(\delta_{t-1}(j) + a_{ji} + y_{(t,i)}\right) \tag{12}$$

$$\varphi_t(i) = \operatorname*{argmax}_j\left(\delta_{t-1}(j) + a_{ji}\right) \tag{13}$$

where π_i is the probability that tag i is the first tag of the sequence, a_{ji} is the transition probability that tag j transfers to tag i, where the larger the value of a_{ji} is, the greater the probability that tag j transfers to tag i, and $y_{(t,i)}$ is the score of word c_t on tag i at time t. $\varphi_t(i)$ points to a certain tag j at the previous time that generated the optimal tag i at time t.

From this, the optimal path to the current time can be determined, and the pointer can be pointed. Therefore, the final tag i_t at time t can be inferred from the local optimal probability at time n. The derivation formula is as follows:

$$i_t = \begin{cases} \text{argmax}(\delta_t(i)) & t = n \\ \varphi_{t+1}(i_{t+1}) & 1 \le t < n \end{cases} \quad (14)$$

4. Experiments

The main parameters of the experimental environment are as follows: the CPU is an Intel(R) Xeon(R) CPU E5-1620 v4 @ 3.50 GHz with 16 GB memory, the deep-learning framework is TensorFlow 1.2.1, the development language is Python 3.6, the Chinese word segmentation tool is the precise mode in the Jieba word segmentation tool, and the text vectorization tool is the BERT service bert-as-service provided by Tencent AI Lab.

The experimentation in this paper is mainly divided into two groups. The first group uses single characters as the input of the neural network. The second group uses words as the input of the neural network. We compare the differences between the segmentation performance and efficiency of four neural networks: Bi-GRU, Bi-LSTM [14], GRU [15], and LSTM [29].

4.1. Datasets

The experimental datasets used in this article were acquired by the use of web crawler technology to obtain point of interest (POI) data and part of the directory data from the Baidu map (https://map.baidu.com) of Beijing, China. The POI data were preprocessed after they were obtained. First, we removed the POI data of repeated addresses and non-Chinese addresses and then performed manual address element segmentation on the addresses in the POI data, as shown in Tables 3 and 4. A total of 189,305 pieces of sample data were ultimately generated (approximately 1.34% of addresses contained misspelled address elements). Among them, 80% of the data was used as model training data (20% of the training data was used for cross-validation data in hyperparameter settings), and 20% of the data was used as model testing data.

Table 3. Examples of point of interest (POI) address data.

Chinese Address	English Address
北京市西城区车公庄大街甲4号	No.4 A Chegongzhuang Street Xicheng District Beijing City
北京市朝阳区亮马桥路31号院附近	Near Yard 31 Liangmaqiao Road Chaoyang District Beijing City
北京市海淀区颐和园路5号北京大学内	Inside Peking University No. 5 Yiheyuan Road Haidian District Beijing City
北京幸福北里29号楼	Building 29 Xingfu Beili Beijing

Table 4. Examples of manual segmentation of POI address elements.

Chinese Address	English Address
北京市/西城区/车公庄大街/甲4号	No.4 A, Chegongzhuang Street, Xicheng District, Beijing City
北京市/朝阳区/亮马桥路/31号院/附近	Near, Yard 31, Liangmaqiao Road, Chaoyang District, Beijing City
北京市/海淀区/颐和园路/5号/北京大学/内	Inside, Peking University, No. 5, Yiheyuan Road, Haidian District, Beijing City
北京/幸福北里/29号楼	Building 29, Xingfu Beili, Beijing

Regarding the directory data, we used keywords such as roads, residential areas, office buildings, parks, and schools to obtain directory data for some roads, residential areas, office buildings, parks, and schools from the Baidu map of Beijing. Each record in the directory data corresponds to an address element in Chinese addresses. Therefore, manual segmentation is not required, and the information can be directly used as a training sample for the model. Additionally, the administrative division is an

essential address element in each address, and this element is relatively stable and does not change frequently. Therefore, each administrative division of Beijing can also be used as a supplementary training sample. The above directory data and administrative division data total 28,262.

4.2. Evaluation Criteria

For experimental evaluation criteria, the classification evaluation method of machine learning is used. This method uses three indicators, the precision (P), recall (R), and F1-score (F1), to evaluate the segmentation results. The precision refers to the proportion of correctly segmented address elements with respect to all segmented address elements, the recall refers to the proportion of correctly segmented address elements to all standard address elements, and the F1-score refers to the weighted harmonic average of the precision and recall. The calculation formula of each indicator is as follows:

$$P = \frac{|A \cap B|}{|B|} \tag{15}$$

$$R = \frac{|A \cap B|}{|A|} \tag{16}$$

$$F1 = \frac{2P * R}{P + R} \tag{17}$$

where A is the standard set of address elements, and B is the segmented set of address elements.

4.3. Hyperparameters

To obtain a favorable performance from a neural network, the setting of the hyperparameters is extremely important. The main hyperparameters of this experiment are shown in Table 5. In the text vectorization layer of the model, the window size is the number of characters contained in the longest address in the address datasets. The word vector dimension is a commonly used character embedding dimension [38,39]. In the neural network layer, the number of hidden units and the number of neural network layers represent a compromise between the model training speed and model performance [40]. The dropout rate and batch size are selected through comparative experiments to determine the optimal value. In the label inference layer, after tag statistics are acquired for the character training dataset and the word training dataset, the tag transition matrix A and the vector initialization vector π are obtained, as shown in Equations (18) and (19).

$$A_1 = \begin{array}{c} \\ B \\ M \\ E \\ S \end{array} \begin{array}{c} \begin{array}{cccc} B & M & E & S \end{array} \\ \begin{bmatrix} 0 & 0.90 & 0.10 & 0 \\ 0 & 0.40 & 0.60 & 0 \\ 0.96 & 0 & 0 & 0.04 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad \pi_1 = \begin{array}{c} \\ B \\ M \\ E \\ S \end{array} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

$$A_2 = \begin{array}{c} \\ B \\ M \\ E \\ S \end{array} \begin{array}{c} \begin{array}{cccc} B & M & E & S \end{array} \\ \begin{bmatrix} 0 & 0.23 & 0.77 & 0 \\ 0 & 0.23 & 0.77 & 0 \\ 0.71 & 0 & 0 & 0.29 \\ 0.48 & 0 & 0 & 0.52 \end{bmatrix} \end{array} \quad \pi_2 = \begin{array}{c} \\ B \\ M \\ E \\ S \end{array} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{19}$$

where A_1 and π_1 are the tag transition matrix and initialization vector when single characters are the input. A_2 and π_2 are the tag transition matrix and initialization vector when words are the input.

The dropout rate is intended to prevent overfitting by discarding some hidden layer nodes in the network [41]. From Figure 4, we can see that, during the network model training, the F1 value of each neural network increases with the increasing dropout rate. The convergence speed of the network model is faster with the increasing dropout rate. To verify the overfitting effect of each

dropout rate, we cross-validate the model during the training process, and the results obtained are shown in Table 6. Table 6 shows that when the dropout rate is not set—that is, when the dropout rate is 1—each neural network achieves its lowest F1 value on the testing set. When the dropout rate is 0.7, each neural network achieves its best F1 value on the testing set: that is, the overfitting effect is the best. In summary, the dropout rate of the text experiment is set to 0.7.

Table 5. Hyperparameter settings.

Hyperparameters	Value
Window size	32
Word vector dimension	200
Hidden unit number	128
Neural network layers	2
Batch size	2000
Dropout rate	0.7

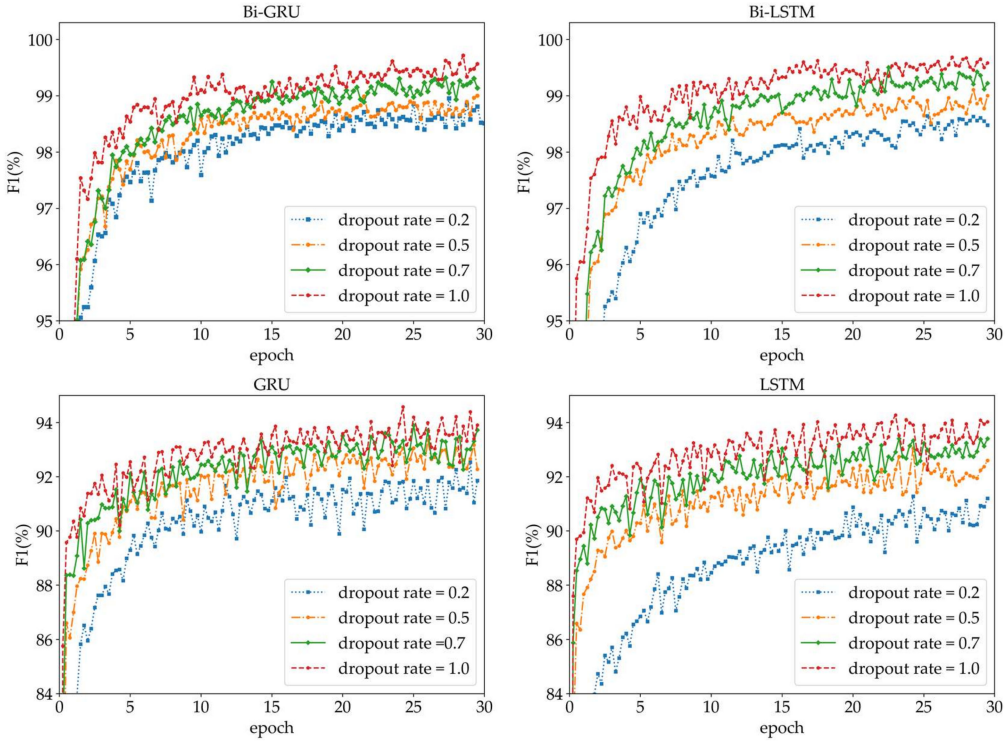

Figure 4. The comparison of different dropout rate hyperparameters for each neural network. LSTM: long short-term memory.

Table 6. The results for different dropout rates in each neural network on the cross-validation data. LSTM: long short-term memory.

Dropout Rate	Neural Network			
	Bi-GRU	Bi-LSTM	GRU	LSTM
0.2	96.57%	95.85%	89.73%	90.20%
0.5	98.34%	99.03%	90.52%	91.21%
0.7	**99.20%**	**99.07%**	**91.73%**	**92.16%**
1.0	90.82%	91.34%	85.76%	89.54%

The bold font indicates the optimal dropout rate for different neural networks.

The batch size is the number of samples selected for one iteration of training of the neural network, and its size affects the optimization and speed of the model. From Figure 5, we can see that, when the batch sizes are 500 and 1000, the convergence speed of each network model is the fastest. However, the training error of the model is high, and the oscillations are large. When the batch size is 3000, the training error of the network model decreases. However, a local optimal situation occurs during the training process, and the F1 indicator peaks at the end of each round of training. When the batch size is 2000, although the convergence speed is slower than those of the 500 and 1000 batch sizes, the descent direction is accurate, the oscillations are small, the training error becomes lower, and there are no local optima. Therefore, the batch size of the text experiment is set to 2000.

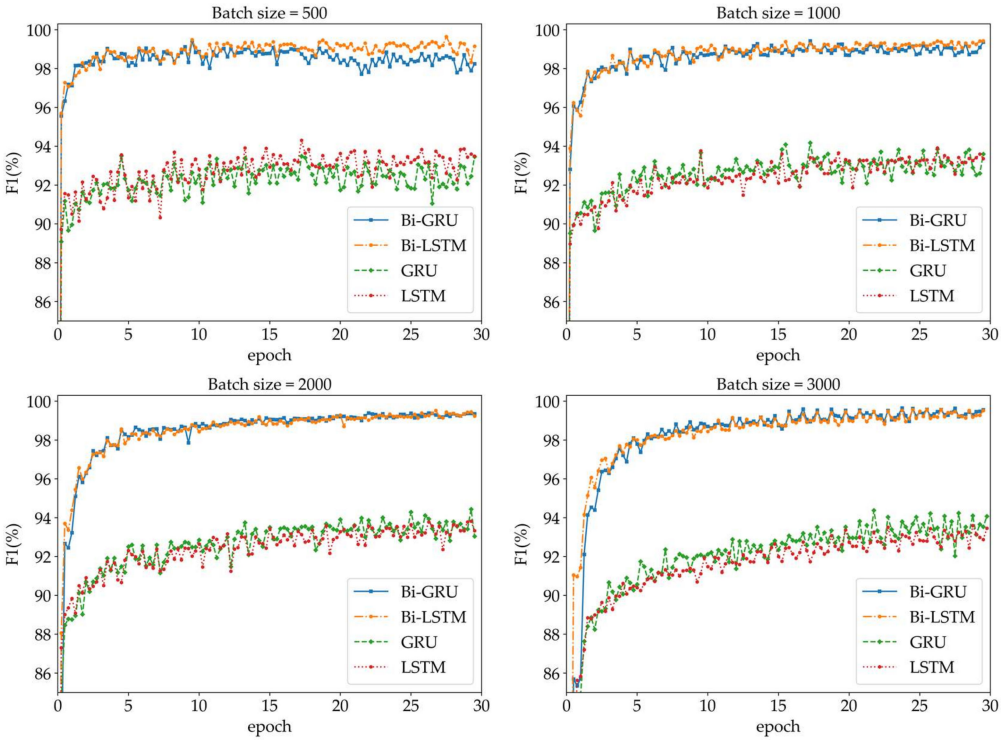

Figure 5. Comparison of different batch size hyperparameters for each neural network.

The same parameters are used for the Bi-GRU, Bi-LSTM, CRF, and LSTM neural networks so that the training speed and performance of the four models can be compared.

4.4. Results

Through experiments, the results of the segmentation of the address elements and the training times of the Bi-GRU, Bi-LSTM, GRU, and LSTM neural networks for different inputs are shown in Tables 7 and 8. We compare the segmentation indicators of all neural networks, as shown in Figure 6.

Table 7. The segmentation results of each neural network are compared for different input models. Precision (P), recall (R), and F1-score (F1).

Neural Network	Character Input			Word Input		
	P	**R**	**F1**	**P**	**R**	**F1**
Bi-GRU	**97.81%**	97.69%	97.75%	**99.22%**	99.10%	**99.16%**
Bi-LSTM	97.69%	**97.88%**	**97.78%**	99.14%	**99.11%**	99.12%
GRU	91.15%	85.81%	88.40%	94.62%	91.87%	93.22%
LSTM	90.65%	86.56%	88.56%	93.99%	91.92%	92.94%

The bold font indicates the neural network with the largest value of segmentation indicators for different input models.

Table 8. The training time of each neural network is compared for different input models.

Neural Network	Character Input	Word Input
Bi-GRU	**517 s/epoch**	**521 s/epoch**
Bi-LSTM	562 s/epoch	567 s/epoch
GRU	**269 s/epoch**	**270 s/epoch**
LSTM	292 s/epoch	293 s/epoch

The bold font in the first row indicates the one with a shorter training time for the two bidirectional neural networks (Bi-GRU and Bi-LSTM) in different input models. The bold font in the third row indicates the one with shorter training time for the two unidirectional neural networks (GRU and LSTM) in different input models.

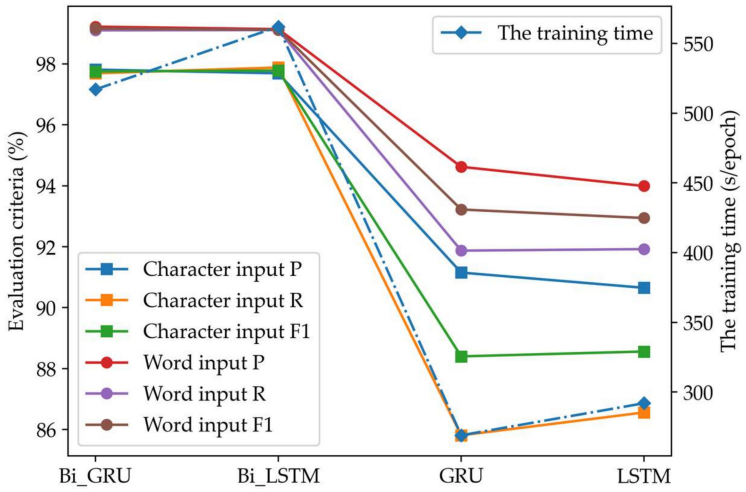

Figure 6. Segmentation results of each neural network for different input models. Precision (P), recall (R), and F1-score (F1).

- Single character input

From Table 7, we can see that of the network models with single character input, the segmentation precision based on the Bi-GRU neural network is the highest at 97.81%. The segmentation recall and F1-score based on the Bi-LSTM neural network are the highest at 97.88% and 97.78%, respectively. However, the difference between the F1-scores of the two bidirectional neural networks is only 0.03%, so the segmentation performance of the two can be considered to be almost the same. The segmentation F1-scores of the GRU and LSTM neural networks are both lower than 89%, only 88.40% and 88.56%, respectively. As can be seen from Figure 6, the segmentation indicators based on the unidirectional neural network GRU and LSTM are lower than those based on the bidirectional neural network Bi-GRU and Bi-LSTM.

From the segmentation efficiency in Table 8, we can see that of the network models with single character input, the Bi-GRU neural network is 8.70% faster than the Bi-LSTM neural network on average per round of training. According to the 30 rounds of training in this paper, the Bi-GRU neural network is approximately 22 minutes faster than the Bi-LSTM neural network. The GRU neural network is 8.55% faster than the LSTM neural network per round of training on average. The unidirectional GRU and LSTM neural networks are nearly twice as fast as the corresponding bidirectional neural networks Bi-GRU and Bi-LSTM on average per round of training.

- Word input

Table 7 shows that, of the network models with word input, the precision and F1-score based on the Bi-GRU neural network are the highest at 99.22% and 99.16%, respectively. The segmentation recall based on the Bi-LSTM neural network is the highest at 99.11%. Similarly, the F1-score difference of the two bidirectional neural networks is only 0.04%, so the segmentation performance of the two can be

considered to be almost the same. The segmentation indicators based on the GRU and LSTM neural networks exceed 90%, among which, F1 reached 93.22% and 92.94%, respectively. However, it can be seen from Figure 6 that their segmentation indicators are still lower than the bidirectional neural networks Bi-GRU and Bi-LSTM.

From the segmentation efficiency in Table 8, we can see that, of the network models with word input, the Bi-GRU neural network is, on average, 8.83% faster than the Bi-LSTM neural network per round of training. The GRU neural network is, on average, 8.52% faster than the LSTM neural network per round of training. Similarly, the unidirectional neural networks GRU and LSTM are, on average, much faster than the corresponding bidirectional neural networks Bi-GRU and Bi-LSTM per round of training.

- Comparison

In terms of segmentation performance, the network models with word input perform better in various indexes of segmentation compared to those of the network models with single character inputs. The precision values of the Bi-GRU, Bi-LSTM, GRU, and LSTM neural networks increase by 1.41%, 1.45%, 3.47%, and 3.34%, respectively. The recall values increase by 1.41%, 1.23%, 6.06%, and 5.36%, respectively. The F1-scores increase by 1.41%, 1.34%, 4.83%, and 4.39%, respectively.

In terms of segmentation efficiency, the two input models are almost the same. Among them, the Bi-LSTM neural network yields the largest difference in average training time per round, which is only five seconds. The Bi-GRU neural network generates a difference of four seconds. The LSTM and the GRU neural networks both yield a difference of one second. Therefore, the network models with single character inputs and the network models with word inputs offer almost equivalent segmentation efficiency.

5. Discussion

5.1. Analysis of the Experimental Results

From the above experimental results, we formulated the following conclusions:

(1) Regardless of whether the network models utilize single-character input or word input, the bidirectional neural networks Bi-GRU and Bi-LSTM offer better segmentation performances than the unidirectional neural networks GRU and LSTM, because a bidirectional neural network can calculate the state before and after each moment from forward and backward directions, respectively. Therefore, this type of network can better consider the past information and future information of the address sequence.

(2) The network models with word input of the above four kinds of neural networks demonstrate better segmentation performances than the corresponding network models with single-character inputs. Chinese address elements are composed of words and are the results of Chinese word segmentation. Therefore, using words as the input of the neural network model conforms to the word-formation rules of Chinese address elements.

(3) In the case of the same segmentation performance, the Bi-GRU neural network is faster than the Bi-LSTM neural network in segmentation efficiency, because the Bi-GRU neural unit directly transmits the hidden state to the next neural unit, whereas the Bi-LSTM neural unit needs to use the memory cell state to package the hidden state and transmit it to the next neural unit. Additionally, when calculating the current hidden state value, the Bi-GRU neural unit needs to output only the value of one parameter at the previous time in addition to the current input, whereas the Bi-LSTM neural unit requires two parameters, the memory cell state value, and the output value of the hidden node at the previous moment.

(4) The network models with word inputs of the above four kinds of neural networks have better segmentation performances and the same segmentation efficiency as the corresponding network models with single-character inputs, because, regardless of whether single-character input or

word input is used, the calculation in the neural network is a feature vector generated after text vectorization. Therefore, the dimensionality of the vector affects the efficiency of segmentation, not the input characters and words.

5.2. Segmentation Issues

Although the method in this paper achieved good segmentation results in the experiment, some segmentation problems still exist. There are two main reasons for these problems: the smallest segmentation unit and address spelling errors. The former accounted for approximately 35.2% of all incorrectly segmented addresses, and the latter accounted for approximately 52.6%. The remaining 12.2% of problems were attributed to many other reasons, including special characters, small sample sizes, and human error generating erroneous samples.

The segmentation method of Chinese address elements proposed in this paper is based on the results of Chinese word segmentation and uses words as the smallest segmentation unit. Therefore, the Chinese word segmentation results directly affect the accuracy of the address element segmentation. Our commonly used Chinese word segmentation tools are based on the principle of "Chinese word formation", and the segmentation of address elements can be regarded as the recombination of word sequences. Therefore, the result of Chinese word segmentation is not the word contained in the address element. For example, the result of the Chinese word segmentation of "甘家口西白堆子(Ganjiakou, Xi, Baiduizi)" is "甘家口/西白/堆子(Ganjiakou, Xibai, Duizi)", and the correct address element is "甘家口/西/白堆子(Ganjiakou, Xi, Baiduizi)". Therefore, the words "Xibai" and "Duizi", as the smallest segmentation unit, cannot be recombined into the address element "Baiduizi".

Incorrect address spellings and address abbreviations in Chinese address expressions also affect the correct segmentation of address elements. Common misspellings of addresses are mostly homophonic spelling errors, such as "广安门(Guanganmen)" being incorrectly spelled as "光安门(Guanganmen)" and "箭厂胡同(Jianchang Hutong)" being incorrectly spelled as "建厂胡同(Jianchang Hutong)". Common address abbreviations are mostly for administrative divisions and roads. For example, "海淀区(Haidian District)" and "海淀街道(Haidian Street)" in administrative divisions are abbreviated as "海淀(Haidian)", and "莲花池西路(Lianhuachi West Road)" is abbreviated as "莲池西路(Lianchi West Road)". These incorrect address element descriptions are not only prone to ambiguity, but they also generate inaccurate model training features.

6. Conclusions

Geocoding technology, as a bridge between spatial information and nonspatial information, plays a crucial role in today's era of big data. The segmentation of Chinese address elements is one of the core techniques of geocoding. Focusing on the problems of the existing segmentation methods and the characteristics of Chinese address elements, this paper proposes a Chinese address element segmentation method based on a Bi-GRU neural network. Chinese word segmentation technology is used to perform Chinese word segmentation on address sequences in advance to generate the input of the neural network model. The experiment is based on the POI address data and some directory data in the Baidu map of Beijing for training and testing. The experimental results show that the bidirectional neural network is superior to the unidirectional neural network in segmentation performance. Moreover, when the Bi-GRU and Bi-LSTM neural networks have the same segmentation performances, the segmentation efficiency of the Bi-GRU neural network is higher, while the segmentation efficiency of the Bi-LSTM neural network is relatively low. Additionally, each neural network demonstrated a better segmentation performance with word input than with single-character input.

In Chinese address expressions, according to the order in which these address elements appear in the address, there is a strong spatial constraint relationship between them—namely, a hierarchical subordination relationship. However, this article starts from the perspective of natural language, thus ignoring the spatial constraint relationship between address elements. In this paper, unstructured

Chinese addresses are segmented into independent and complete address elements. However, they do not have any semantic information, so it is impossible to determine the composition and meaning of their representation in the address. Therefore, the semantic annotation of address elements is an area that necessitates further study.

Author Contributions: Data curation, Yong Wang and Junjie Zhang; Methodology, Pengpeng Li and An Luo; Project administration, Jiping Liu; Resources, Jiping Liu; Writing—original draft, Pengpeng Li; Writing—review & editing, Jun Zhu and Yue Deng. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Plan of China (under project numbers 2017YFB0503601 and 2017YFB0503502) and the Basic Business Cost Project for Central-level Scientific Research Institutes (AR2011).

Acknowledgments: The authors would like to thank the anonymous reviewers and the editor for their constructive comments and suggestions for this paper.

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

References

1. Dhar, S.; Varshney, U. Challenges and business models for mobile location-based services and advertising. *Commun. ACM* **2011**, *54*, 121–128. [CrossRef]
2. Cong, G.; Jensen, C.S. Querying Geo-Textual Data: Spatial Keyword Queries and Beyond. In Proceedings of the 2016 International Conference on Management of Data, San Francisco, CA, USA, 26 June–1 July 2016; pp. 2207–2212.
3. Melo, F.; Martins, B. Automated Geocoding of Textual Documents: A Survey of Current Approaches. *Trans. GIS* **2017**, *21*, 3–38. [CrossRef]
4. Davis, C.A.; Fonseca, F.T. Assessing the Certainty of Locations Produced by an Address Geocoding System. *Geoinformatica* **2007**, *11*, 103–129. [CrossRef] [PubMed]
5. Laylavi, F.; Rajabifard, A.; Kalantari, M. A multi-element approach to location inference of twitter: A case for emergency response. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 56. [CrossRef]
6. Rushton, G.; Armstrong, M.P.; Gittler, J.; Greene, B.R.; Pavlik, C.E.; West, M.M.; Zimmerman, D.L. Geocoding in Cancer Research: A Review. *Am. J. Prev. Med.* **2006**, *30*, S16–S24. [CrossRef] [PubMed]
7. Ratcliffe, J.H. Geocoding crime and a first estimate of a minimum acceptable hit rate. *Int. J. Geogr. Inf. Sci.* **2004**, *18*, 61–72. [CrossRef]
8. Zandbergen, P.A. A comparison of address point, parcel and street geocoding techniques. *Comput. Environ. Urban Syst.* **2008**, *32*, 214–232. [CrossRef]
9. Song, Z. Address matching algorithm based on Chinese natural language understanding. *J. Remote Sens.* **2013**, *17*, 788–801.
10. Kuai, X.; Guo, R.; Zhang, Z.; He, B.; Zhao, Z.; Guo, H. Spatial Context-Based Local Toponym Extraction and Chinese Textual Address Segmentation from Urban POI Data. *ISPRS Int. Geo-Inf.* **2020**, *9*, 147. [CrossRef]
11. Shan, S.; Li, Z.; Yang, Q.; Liu, A.; Zhao, L.; Liu, G.; Chen, Z. Geographical address representation learning for address matching. *World Wide Web* **2020**, *23*, 1–18. [CrossRef]
12. 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.
13. 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. Geo-Inf.* **2016**, *5*, 65. [CrossRef]
14. Yao, Y.; Huang, Z. Bi-directional LSTM recurrent neural network for Chinese word segmentation. In Proceedings of the International Conference on Neural Information Processing, Kyoto, Japan, 16–21 October 2016; pp. 345–353.
15. Li, X.; Duan, H.; Xu, M. A gated recurrent unit neural network for Chinese word segmentation. *J. Xiamen Univ.* **2017**, *56*, 237–243.
16. Lu, Q.; Zhu, Z.; Xu, F.; Zhang, D.; Guo, Q. Bi-GRU Sentiment Classification for Chinese Based on Grammar Rules and BERT. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 538. [CrossRef]
17. Zhang, M.; Yu, N.; Fu, G. A simple and effective neural model for joint word segmentation and POS tagging. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 1528–1538. [CrossRef]

18. Wang, S.; Zhang, H.; Wang, B. Research of Optimization on Double-Array Trie and its Application. *J. Chin. Inf. Proc.* **2006**, *20*, 24–30.
19. Li, Q.; Chen, Y.; Sun, J. A New Dictionary Mechanism for Chinese Word Segmentation. *J. Chin. Inf. Proc.* **2003**, *4*, 13–18.
20. Mo, J.; Zheng, Y.; Shou, Z.; Zhang, S. Improved Chinese word segmentation method based on dictionary. *Comput. Eng. Desig.* **2013**, *34*, 1802–1807.
21. Li, L.; Wang, W.; He, B.; Zhang, Y. A hybrid method for Chinese address segmentation. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 30–48. [CrossRef]
22. Zhang, X.; Lv, G.; Li, B. Rule-based Approach to Semantic Resolution of Chinese Addresses. *J. Geo-Inf. Sci.* **2010**, *1*, 9–16. [CrossRef]
23. Low, J.K.; Ng, H.T.; Guo, W. A maximum entropy approach to Chinese word segmentation. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, 14–15 October 2005; pp. 161–164.
24. Zhang, L.; Qin, M.; Zhang, X.; Ma, H. A Chinese word segmentation algorithm based on maximum entropy. In Proceedings of the International Conference on Machine Learning and Cybernetics, Qingdao, China, 11–14 July 2010; pp. 1264–1267.
25. Zhang, H.; Liu, Q.; Cheng, X.; Zhang, H.; Yu, H. Chinese lexical analysis using hierarchical hidden markov model. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, 11–12 July 2003; pp. 63–70.
26. Asahara, M.; Goh, C.L.; Wang, X.; Matsumoto, Y. Combining segmenter and chunker for Chinese word segmentation. In Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, 11–12 July 2003; pp. 144–147.
27. Peng, F.; Feng, F.; McCallum, A. Chinese segmentation and new word detection using conditional random fields. In Proceedings of the 20th International Conference on Computational Linguistics, Geneva, Switzerland, 23–27 August 2004; pp. 562–568.
28. Lafferty, J.; McCallum, A.; Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning, Williamstown, MA, USA, 28 June–1 July 2001; pp. 282–289.
29. Chen, X.; Qiu, X.; Zhu, C.; Liu, P.; Huang, X. Long short-term memory neural networks for chinese word segmentation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 1197–1206.
30. Chen, X.; Qiu, X.; Zhu, C.; Huang, X. Gated recursive neural network for chinese word segmentation. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Beijing, China, 26–31 July 2015; pp. 1744–1753.
31. Huang, C.R.; Chen, K.J.; Chen, F.Y.; Chang, L.L. Segmentation Standard for Chinese Natural Language Processing. In Proceedings of the International Conference on Computational Linguistics, Taipei, Taiwan, 1–4 August 1997; pp. 47–62.
32. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
33. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
34. Zheng, X.; Chen, H.; Xu, T. Deep learning for Chinese word segmentation and POS tagging. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 647–657.
35. Yu, C.; Wang, S.; Guo, J. Learning Chinese Word Segmentation Based on Bidirectional GRU-CRF and CNN Network Model. *IJTHI* **2019**, *15*, 47–62. [CrossRef]
36. Liu, Z.; Ding, D.; Li, C. Chinese word segmentation method for short Chinese text based on conditional random fields. *J. Tsinghua Univ. Nat. Sci. Ed.* **2015**, *55*, 906–910.
37. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 8 June 2019; Volume 1, pp. 4171–4186.

38. Chen, X.; Xu, L.; Liu, Z.; Sun, M.; Luan, H. Joint learning of character and word embeddings. In Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015; pp. 1236–1242.
39. Lu, Y.; Zhang, Y.; Ji, D.H. Multi-prototype Chinese character embedding. In Proceedings of the Tenth International Conference on Language Resources and Evaluation, Portorož, Slovenia, 23–28 May 2016; pp. 855–859.
40. Fan, T.; Zhu, J.; Cheng, Y.; Li, Q.; Xue, D.; Munnoch, R. A New Direct Heart Sound Segmentation Approach using Bi-directional GRU. In Proceedings of the 2018 24th International Conference on Automation and Computing, Newcastle, UK, 6–7 September 2018; pp. 1–5.
41. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Neural Information Processing Systems Conference, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.