International Journal of Geo-Information

MDPI

Article

A Coarse-to-Fine Model for Geolocating Chinese Addresses

Chunyao Qian [1], Chao Yi [1], Chengqi Cheng [2], Guoliang Pu [2,*] and Jiashu Liu [3]

[1] School of Earth and Space Sciences, Peking University, Beijing 100871, China; qianchunyao@pku.edu.cn (C.Q.); pkuyichao@pku.edu.cn (C.Y.)
[2] College of Engineering, Peking University, Beijing 100871, China; ccq@pku.edu.cn
[3] Center for Data Science, Peking University, Beijing 100871, China; liujiashu@pku.edu.cn
* Correspondence: pgl@pku.edu.cn; Tel.: +86-010-6275-5390

Received: 17 October 2020; Accepted: 23 November 2020; Published: 25 November 2020

Abstract: Address geolocation aims to associate address texts to the geographic locations. In China, due to the increasing demand for LBS applications such as take-out services and express delivery, automatically geolocating the unstructured address information is the key issue that needs to be solved first. Recently, a few approaches have been proposed to automate the address geolocation by directly predicting geographic coordinates. However, such point-based methods ignore the hierarchy information in addresses which may cause poor geolocation performance. In this paper, we propose a hierarchical region-based approach for geolocating Chinese addresses. We model the address geolocation as a Sequence-to-Sequence (Seq2Seq) learning task, that is, the input sequence is a textual address, and the output sequence is a GeoSOT grid code which exactly represents multi-level regions covered by the address. A novel coarse-to-fine model, which combines BERT and LSTM, is designed to learn the task. The experimental results demonstrate that our model correctly understands the Chinese addresses and achieves the highest geolocation accuracy among all the baselines.

Keywords: address geolocation prediction; Chinese addresses; GeoSOT grid code; deep neural network; BERT; sequence-to-sequence

1. Introduction

Addresses, as natural language descriptions of geographic locations, are often used by humans in daily life. In China, there are increasing demands for various LBS applications, such as take-out services, express delivery, online car-hailing services, etc. While the unstructured addresses are easy for humans to understand and locate, they are difficult for computers to operate with. To enable the use of unstructured address information by these applications, one prerequisite is automatically assigning the correct geographic locations for the addresses. This process is commonly called address geolocation.

Accurate estimation of address location is an important factor for LBS applications. In recent years, deep learning models have been explored for the textual geolocation prediction task. Numerous studies adopted deep neural networks to predict coordinates (i.e., longitudes and latitudes) of text data including blogs, tweets, Wikipedia articles, etc. However, such point-based geolocation methods ignore the hierarchy information in address descriptions (e.g., country, province, city, etc.), which has been shown to be very effective in previous studies [1,2]. In addition, recent work [3] also demonstrates that predicting coarse-grained areas is much easier than predicting fine-grained areas.

Motivated by these studies, we consider a hierarchical region-based approach for Chinese address geolocation. To be more specific, our goal is to predict multiple regions (from coarse grain to fine grain) covering the target address location while the coarse-grained region prediction can guide the fine-grained region prediction. However, two technical challenges stand in our way to reach this goal. First, it is challenging to understand the meaning of the addresses. Polysemy is a very common phenomenon in Chinese addresses. An example is shown in Figure 1. The characters in red wireframe are the same but represent a completely different level of the hierarchy and meaning of the address. Second, determining a series of hierarchical regions covering the target location and predicting each of them one by one is an intricate task. How to construct a hierarchy of the related regions and how to predict the fine-grained region under the supervision of the coarse-grained region prediction remain unclear.

Figure 1. Polysemy in Chinese.

To address these challenges, we propose a novel coarse-to-fine model for geolocating Chinese addresses. Our proposed model is based on an encoder–decoder framework augmented with an attention mechanism [4]. We address the first challenge by taking the state-of-the-art language model BERT (Bidirectional Encoder Representations from Transformers) as the encoder, which is capable of extracting different embeddings of the Chinese characters according to the different contexts. We tackle the second challenge by adopting a multilevel subdivision scheme for the earth's surface, known in the literature as GeoSOT (Geographical coordinate Subdividing grid with one dimension integer coding on a 2^n Tree) [5]. Based on GeoSOT, we first build hierarchies of regions related to target address locations in the training phase. Since each region has a unique identification code, we then train an LSTM (Long Short-Term Memory) [6] network based decoder to predict each region's code by attending to the address semantic meaning.

We make the following contributions in this work: (1) We creatively model the Chinese address geolocation as a Seq2Seq learning task in which the input is the textual Chinese address and the output is a GeoSOT grid code. (2) A novel coarse-to-fine model is proposed, which takes the BERT as encoder and an LSTM model as a decoder. (3) We demonstrate the effectiveness of our coarse-to-fine geolocation model by conducting detailed experiments. It significantly outperforms the baseline methods.

2. Related Work

This section introduces prior studies that are most relevant to our work, including text-based geolocation prediction and neural language modeling.

2.1. Textual Geolocation Prediction

Textual geolocation aims at locating the textual addresses with language modeling techniques. According to the predicting targets, prior studies can be divided into two categories: for coordinates and for regions.

Coordinate-oriented approaches view the geolocation task from the regression perspective and directly predict the longitude and latitude of text data. They are widely used in social media data

geolocation, such as tweets, blogs, social images, etc. Considerable literature has developed various geolocators by leveraging different features of text content and users, such as location indicative words, metadata, user profiles, and friendship graphs. For example, as a pioneering work, Fink et al. [7] presented a method that uses the place name mentions in a blog to determine the blog's location. Chi et al. [8] integrated location indicative words, city/country names, hashtags, and mentions and trained a multinomial Naive Bayes classifier to predict the locations of tweets. Liu et al. [9] proposed a unified framework to predict geolocations for Flickr images, which combines the information from both image tags and the user profile. Rahimi et al. [10] proposed GCN, a multiview geolocation model based on graph convolutional networks that uses both text and network context. Miura et al. [11] unified text, metadata, and user network representations with a neural network for geolocation prediction. However, most studies are conducted on social media data like Twitter, where metadata and external gazetteers are needed. By contrast, our geolocation model only relies on textual features.

Region-oriented approaches take the geolocation prediction as a classification task by first partitioning the regions into discrete subregions using regular grids, adaptive grids or city-level regions. These approaches treat the resulting discrete regions as either a flat list [12–18] or a nested hierarchy [2,3]. For example, Wing and Baldridge [12] was the first to use the n-gram statistical language model and a discrete, regular grid division of the earth's surface to predict the grids belonging to the document. It was extended by Roller et al. [13] with additional considerations of data distribution, the authors defined an alternative grid construction using k-d trees that more robustly adapt to data. Rout et al. [16] uses an SVM classifier and a number of features that reflect different aspects and characteristics of Twitter user networks to predict city-level location. Dredze et al. [18] adopted a supervised learning approach, training a multiclass classifier to identify the city of a tweet. Foregoing taking discrete regions as a flat list, the other research thread tried to predict text geolocation hierarchically by treating the discrete regions as a nested hierarchy. Mahmud et al. [1] developed a two-level hierarchical location classifier which first predicts time zone or state, and then predicts the city label. Wing and Baldridge [2] constructed a grid hierarchy. The probability of the final fine-grained location can be computed recursively from the leaf node up to the root. Recently, Kulkarni et al. [19] proposed a multilevel geocoder (MLG) for geolocating tweets. MLG exploits the natural hierarchy of the geographic locations by jointly predicting at different levels of granularity. However, with the deepening hierarchy, such classification-based geolocation methods can hardly handle the classification because the output space is too large. To overcome this limitation, we propose the coarse-to-fine model (CFM) to achieve multilevel geolocation in a Seq2Seq fashion. To the best of our knowledge, our method is the first deep learning-based neural network which models the geolocation prediction as a Seq2Seq task.

2.2. Neural Language Modeling

Language modeling aims to learn the joint probability of word sequences in a language. The first neural language model was proposed by Bengio et al. [20], who proposed to represent each word by a continuous real-vector and leverage a feedforward neural network to learn the distributed representation of each word. Compared with the traditional statistical language model, the neural language model substantially ameliorates the curse of dimensionality and exhibits better generalization ability. With the rapid development of deep learning technologies, the feedforward neural network-based language model was later extended to recursive neural networks [21] and convolutional neural networks based language models [22]. However, these models for learning word embeddings only allow a single context-independent representation for each word. In other words, they can hardly handle polysemy. To solve this problem, the concept of pretraining word embeddings was proposed [23] and widely adopted in ELMo (Embeddings from Language Models) [24], GPT (Generative Pre-Training) [25] and BERT [26]. ELMo is a two-layer bidirectional LSTM model. It learns the representation for each word depending on

the entire context in which it is used. Therefore, even the same word will have different representations if the context is different. It has been proven that ELMo functions well for the word disambiguation task. GPT uses the Transformer [27] decoder (uni-directional) instead of the LSTM as the language model to better capture long-distance word relations. Moreover, fine-tuning the language model is taken as a training target together with downstream tasks. BERT integrates the advantages of ELMo and GPT, which takes the transformer encoder (bidirectional) as the language model. It achieves great success in a wide range of NLP tasks. Recently, a lite BERT (i.e., ALBERT) [28] is proposed to decrease memory consumption and increase the training speed of BERT. In this work, we leverage the pretrained BERT to extract the character representations in Chinese addresses.

3. Methodology

3.1. Problem Statement

We model the coarse-to-fine geolocation as a Seq2Seq task. The given address V can be viewed as a sequence of n Chinese characters $\{v_1, v_2, \ldots, v_n\}$. The output of the model is the GeoSOT grid code C , which is a sequence of digits containing p quaternary digits $\{c_1, c_2, \ldots, c_p\}$ with c_t being the digit at time t. We formulate the geolocation as the inference over a probabilistic model. The goal of the inference is to generate a code sequence $c^*_{1:p}$ which maximizes $P(c_{1:p}|v_{1:n})$:

$$c^*_{1:p} = \arg\max_{c_{1:p}} \prod_{t=1}^{p} P\left(c_t | c_{0:t-1}, v_{1:n}\right) \tag{1}$$

3.2. Overall Architecture

Figure 2 illustrates the overall architecture of our coarse-to-fine geolocation model. Essentially, it follows an encoder–decoder framework with an attention mechanism. The encoder is used to learn the location-specific information implied in the input address V, and the decoder is used to generate the GeoSOT code sequence C.

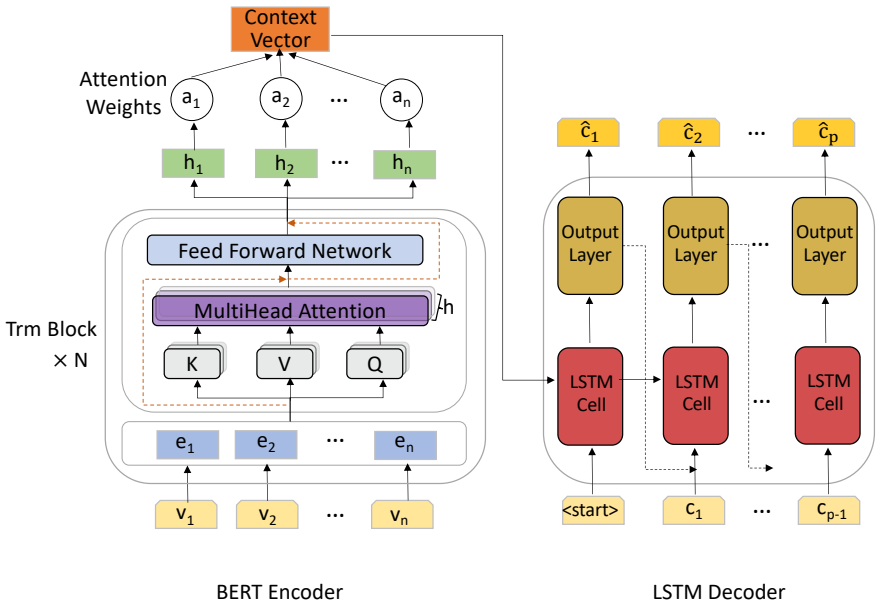


Figure 2. The overall architecture of CFM.

As Chinese addresses are inherently difficult for a machine to understand, we leverage the advanced language model BERT as the encoder to capture the complex relationships between Chinese characters in the address. It consists of N identical Transformer (abbr. **Trm**) blocks. The encoder takes the Chinese address as input and outputs the feature representation h_i for each Chinese character. Considering that the hierarchical locations covered by the given address are represented by a GeoSOT grid code, a LSTM-based decoder is used to predict the code digit one by one. The probability for each digit is computed after a character-level attention layer. The details of our model are elaborated in the following sections.

3.3. GeoSOT Subdivision Scheme

GeoSOT (Geographical coordinate Subdividing grid with One dimension integer coding on a 2^n Tree) is a geo-referencing and coding framework [5]. Taking the intersection of the prime meridian and equator as the central point, GeoSOT recursively divides the surface of the earth into four grid cells. It finally constructs a hierarchical quadtree with 32 levels spanning from the global to the centimeter scale. Table 1 shows the grid size at each level.

Table 1. GeoSOT grid size of different levels.

Level	Grid Size	Level	Grid Size	Level	Grid Size	Level	Grid Size
1	-	9	128 km	17	512 m	25	2 m
2	-	10	64 km	18	256 m	26	1 m
3	-	11	32 km	19	128 m	27	0.5 m
4	-	12	16 km	20	64 m	28	25 cm
5	-	13	8 km	21	32 m	29	12.5 cm
6	1024 km	14	4 km	22	16 m	30	6.2 cm
7	512 km	15	2 km	23	8 m	31	3.1 cm
8	256 km	16	1 km	24	4 m	32	1.5 cm

Grid cells at each level are indexed using a Z-order filling curve [29]. Each cell can be represented as a single string containing quaternary numbers such as '0', '1', '2' and '3'. The longer the GeoSOT code length, the finer the grid granularity. The subdivision and coding method is shown in Figure 3. The advantages of GeoSOT codes are two-fold: (1) uniqueness, in which each geographical region on the Earth has only one unique GeoSOT code; (2) recursiveness, which is the lower-level grids that are subdivided by the upper-level grids. The GeoSOT grid code can represent the geospatial hierarchies at various levels without relying on external metadata.

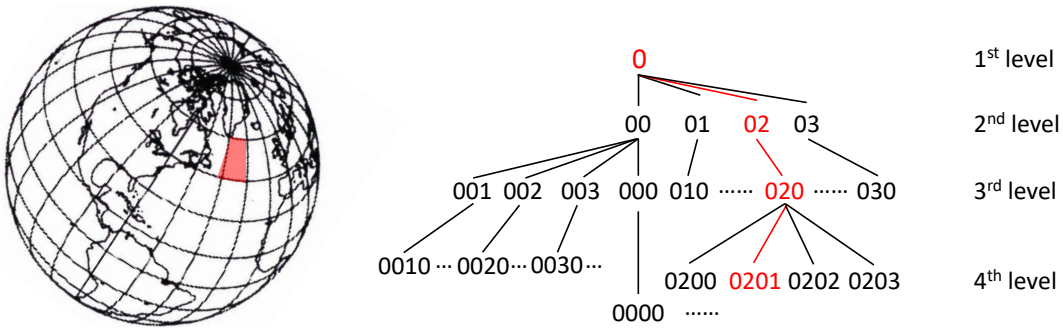


Figure 3. GeoSOT subdivision model.

3.4. Representing Chinese Textual Addresses

Given the raw Chinese addresses, we first aim to transform them into a computer-operable form, then extract geographical features (e.g., location or spatial relations) that can be understood by computers to support the subsequent geolocation prediction.

3.4.1. Input Processing

Tokenization. is the process of splitting the raw text into smaller pieces. Different from nonlogo syllabary languages, such as English, the Chinese language is formed by a stream of characters with no white space to separate them. In addition, there are a huge number of word-level combinations in Chinese, which means that building a word-level vocabulary is more likely to encounter out-of-vocabulary situations in the testing phase. By contrast, the number of Chinese characters is relatively limited, and we can easily exhaust all Chinese characters to construct the vocabulary. Based on the above observations, we consider performing the character-level tokenization for Chinese address texts in this work.

Input Embedding. In the previous step, we obtained a sequence of character-level tokens for each address. To further transform them into a computer-operable form, we take advantage of the word embedding technique. Word embeddings are the distributed representations of words, which encode each word into a unique real-valued vector [30,31]. Compared to the traditional one-hot representations, word embeddings are able to overcome the sparsity of training data and greatly reduce trainable parameters.

In our work, the embedding vector e_i for each character-level token v_i is directly retrieved from an embedding matrix E by a lookup operation. Moreover, the token positions are added to the initial input to record the location information. Similarly, we transform each token's position into an embedding, called position embedding p_i, which is retrieved from another embedding matrix P. Both E and P are trainable. For each character, we sum the token embedding e_i and the position embedding p_i. Finally, an input embedding matrix X is obtained.

3.4.2. Feature Extraction

After the input embedding layer, each Chinese character in the raw addresses is transformed into a 2D vector. We then apply the encoder module, i.e., the BERT model, to extract high-level semantic features from the input embedding matrices. The encoder module consists of N identical blocks (i.e., transformer blocks). Each block contains a multihead self-attention layer (MultiHead) and a feed-forward layer (FFN).

The self-attention mechanism [27] allows each character in the same address to build an attentive context by weighting them with different relevance to each other regardless of the address length. Formally, the computation steps in this layer are as follows:

$$f = \text{MultiHead}(X) = [\text{head}_1, \ldots, \text{head}_h]W^O, \tag{2}$$

$$\text{head}_i(X) = \text{softmax}(\frac{(XW_i^Q)(XW_i^K)^T}{\sqrt{d_k}})(XW_i^V). \tag{3}$$

where W_i^Q,W_i^K,W_i^V,W^O are trainable parameters and d_k is the dimension of W_i^K. Concatenating h heads together, we obtain one feature vector f after projection by W^O for each input character in the address. Following the MultiHead layer, the FFN layer is applied to generate the output of the block. Similar to [27], we employ the residual connection (brown dotted line in Figure 2) and layer normalization around two blocks.

3.5. Coarse-to-Fine Location Prediction

To conduct the coarse-to-fine location prediction, i.e., predicting the GeoSOT code sequence essentially, we leverage the LSTM architecture with attention mechanism as our decoder.

LSTM [6] is a recursive neural network which introduces a cell state and three elementwise multiplication gates, called forget gate, input gate and output gate, to control the cell state. These three gates control how information is stored, forgotten, and exploited inside the network.

As defined in Equation (1), the generated GeoSOT code c_t at time t is predicted based on all the previously generated parent codes $c_{<t}$ before c_t and the hidden states $H = \{h_t\}_{t=1}^{L}$ of the encoder. To be more specific:

$$\mathrm{P}\left(c_t \mid H, c_{<t}\right) = \mathrm{softmax}\left(W_s \odot g_t\right), \tag{4}$$

$$g_t = \tanh\left(W_t[s_t, a_t]\right), \tag{5}$$

$$s_t = \mathrm{LSTM}(c_{t-1}, s_{t-1}). \tag{6}$$

where s_t is the t-th hidden state of the decoder calculated by the LSTM cell. a_t is the attention vector which is widely used in many applications. The vanilla attention mechanism is proposed to focus on the semantic relevance between the encoder states $\{h_t\}_{t=1}^{L}$ and the decoder state s_t at time t. The attention vector is usually represented by the weighted sum of the encoder hidden states:

$$a_t = \sum_{i=1}^{L} \beta_{t_i} h_i,\ \beta_{t_i} = \mathrm{softmax}\left(u \tanh\left(\left[W_1^{Att} s_t, W_2^{Att} h_i\right]\right)\right). \tag{7}$$

where u, W_1^{Att}, and W_2^{Att} are learnable parameters in the attention mechanism.

4. Results and Discussion

4.1. Experiment Settings

Datasets. The address dataset (https://doi.org/10.18170/DVN/WSXCNM) is collected from Amap, which is a leading digital map content provider in China. The dataset covers the whole country and the total size is 400,000. It contains multiple attributes, including POI name, category, hierarchical address description, latitude and longitude, etc. We preprocess the raw dataset by deleting duplicate or nonconforming records (e.g., weird separators, non-Chinese characters). After preprocessing, 385,793 addresses are used. The distribution of the token length of all the addresses is shown in Figure 4. The average number of tokens is 12. In addition, according to the latitude and longitude information, we calculate the GeoSOT code based on the 17th level for each address. An example of the processed address data is shown in Table 2.

Table 2. An example address from the dataset.

Address (In Chinese)	Address (In English)	Lon.	Lat.	GeoSOT Code (L17)
北京市海淀区民族园路2号大润发超市	RT-MART, Minzuyuan Road No.2, Haidian District, Beijing	116.391121	39.982151	30232031113311211

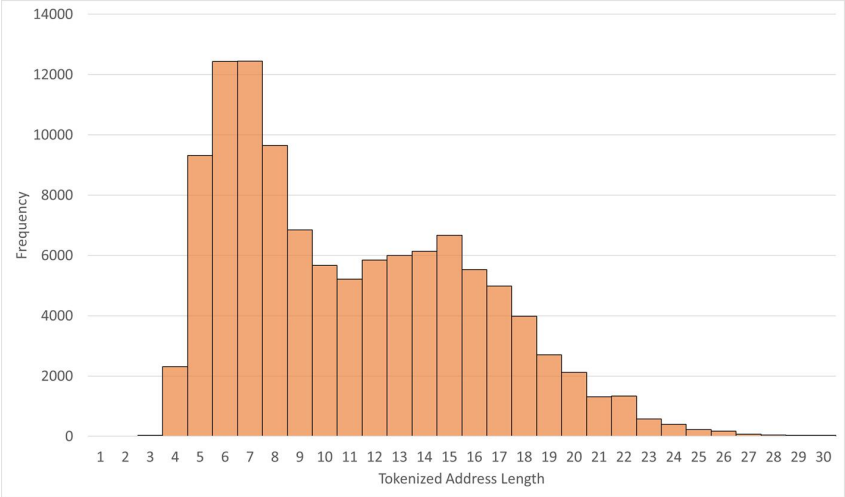


Figure 4. The distribution of the Chinese address lengths.

Implementation Details. The dataset is divided into training, validation and testing set in 8:1:1 proportions. We implement our approach by PyTorch (https://pytorch.org). In terms of hyper-parameter setting, the number of layers (i.e., Transformer blocks) and the number of self-attention heads in the encoder is 12. The dimensions of hidden vector are set as 768 in the encoder and decoder. We use the Adam optimizer [32] with the batch size 100 and the learning rate 0.001. The network was trained for 400 epochs and the best epoch was chosen by observing the performance on the validation set. In addition, all the training in this work was done on a single NVIDIA GeForce GTX 1080 Ti GPU with 11 GB RAM.

4.2. Visualizing the Performance in Polysemy Recognition

We claimed earlier that polysemy is a common phenomenon in Chinese addresses, which presents a challenge to correct geolocation. To demonstrate that our model is able to recognize the different semantic or geographical meanings of the same Chinese character, we visualize the t-SNE [33] plot of the learned character embeddings with tensorboard (https://tensorflow.google.cn/tensorboard).

First, we explore the Chinese character “市”, which is a common example of polysemy in Chinese addresses. Semantically, it can represent both a city-level region (e.g., Beijing, Shanghai, Guangzhou, etc.) and a market (e.g., supermarket, bazaar, country fair, etc.). Figure 5a shows two obvious clusters as the red cluster represents the “region” meaning and the blue cluster refers to the “market” meaning. The two clusters are separated because they have no semantic association at all.

(a) Embeddings of “市” (in Chinese) (b) Embeddings of “区” (in Chinese)

Figure 5. Performance in distinguishing polysemy.

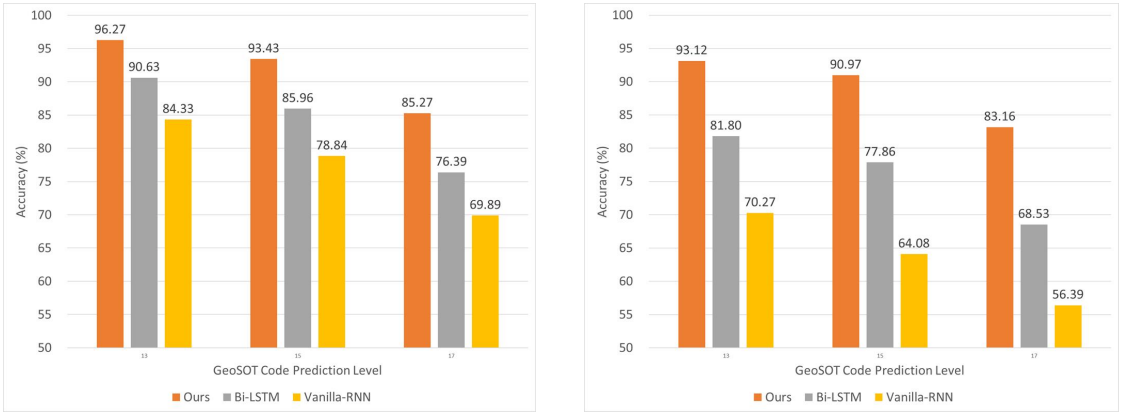
We provide further evidence of our model's distinguishability with another Chinese character "区". As we showed earlier by the example Figure 1, this character can represent both a district-level region and a residence-level region. Though they refer to similar semantic meanings, i.e., geographic regions, the geospatial meanings are different. An interesting finding is shown in Figure 5b. Character embeddings that refer to large geographic regions (i.e., districts) are clustered together (see the red dots). Similarly, those refer to small geographic regions (i.e., residential quarters) are also clustered together (see the blue dots).

The reasons why our model can distinguish polysemy are two-fold. First, the Bert-based encoder helps address this to a certain extent. It can capture the contextual information and obtain the precise semantic meaning of Chinese characters. Second, the coarse-to-fine predicting strategy assists. When decoding different levels of geographic regions, the model is forced to attend to the input characters that are truly useful.

4.3. Comparison in Geolocation Prediction

We evaluate our proposed method using three metrics. Accuracy is the percentage of correctly predicted GeoSOT codes. Taking the prediction of the L17 GeoSOT code as an example, only when all 17 digits are predicted correctly is it considered as a correct case. We take it as a hard metric because GeoSOT code can be directly used in various downstream applications. This is why correctly predicting the total GeoSOT code is important. Moreover, we use two distance-based metrics which are often used in textual geolocation related works: the mean and median error distances [34] between the centeroid of the assigned GeoSOT grid and the actual coordinate.

Accuracy. Two classic models often used in Seq2Seq learning are taken as baselines. One is the **Vanilla-RNN** model. It adopts a basic RNN to map the input sequence to a vector of a fixed dimension, and then uses another deep RNN to decode the target sequence from the vector. Similarly, we take the character embedding as the input and predict the corresponding GeoSOT grid code. The other one is the **Bi-LSTM** model. It is also provided as a strong baseline which uses the bidirectional LSTM units and character level attention mechanism. The performances of different models in terms of GeoSOT code prediction accuracy are presented in Figure 6. The difference between Figure 6a,b is the setting of input address sequence lengths, that is, the average character length of the former is 10 and that of the latter is 20. Moreover, under the same input length, we predict GeoSOT codes of different lengths: 13, 15 and 17, respectively. Please note that longer GeoSOT codes represent finer regions.



(a) Average address length: 10 (b) Average address length: 20

Figure 6. Comparison of the geolocation accuracy under different input/output lengths.

It is clearly shown in Figure 6 that our method outperforms the other baselines significantly. When the input length is fixed, the prediction accuracy of the three methods will decrease as the GeoSOT code length increases. This is in line with common sense since predicting fine-grained regions is more difficult than predicting coarse-grained regions. However, our model still outperforms the baselines significantly. Moreover, if we fix the output code length, the Vanilla-RNN model and Bi-LSTM model achieve similar performance when the input addresses are short. However, when the address lengths become longer, the Bi-LSTM model outperforms the Vanilla-RNN model. Regardless of the input address length, the geolocation performance of our model is stable and outperforms the baselines. This is because the self-attention mechanism used in our encoder is not sensitive to the sequence length. It exhibits superiority in capturing the correct context information. In addition, it is worth noting that, with the input and output length getting longer, the geolocating task becomes more difficult across all methods with increasing input and output length.

Distance-based metrics. We take two state-of-the-art machine learning algorithms based on decision trees as baselines. In detail, we experiment with the two following algorithms: (1) XGBoost-regression, (2) XGBoost-classification. XGBoost (extreme Gradient Boosting) is an advanced implementation of the gradient boosting algorithm. During the training phase, XGBoost grows a sequence of weak learners (i.e., shallow trees), in which each weak learner focuses on correcting the residual errors of the current model approximation. By aggregating the weak learner outputs, XGBoost generates a strong learner. Given the Chinese addresses, we use XGBoost-regression to predict the coordinates. We calculate the mean and median error distances between the predicted location and the actual location. As for XGBoost-classification, it is used to predict the GeoSOT code. Specifically, given a GeoSOT level, it predicts over a large set of grid cells. Similarly, we take the centeroid of the predicted GeoSOT grid to calculate the distance-based metrics. We implement the aforementioned methods with SciKit-learn (https://scikit-learn.org/). In terms of the hyper-parameter setting, the max depth and eta are set as 7 and 0.1, respectively. Moreover, we set the hidden size as 768, 1024, and 2048, respectively, in our model for comparison.

The performances of each model are shown in Table 3. The results show that the regression-based method that directly predicts coordinates performs poorly. As for XGBoost-classification, it is also not easy to predict correctly over a large number of classes. Taking the 13th level as an example, there are almost 14 million GeoSOT grids in the world. By contrast, our proposed model consistently outperforms the other two models under any hidden layer dimension. We attribute it to the fact that our method sequentially learns to assign multi-granularity geographic areas according to the hierarchical geographic information implied in the addresses.

Table 3. Comparison of the distance-based errors.

Methods	XGBoost-Regression	XGBoost-Classification	Ours-768	Ours-1024	Ours-2048
Mean (m)	1170.4	723.8	612.3	583.2	552.6
Mid (m)	1065.0	591.5	497.7	460.0	423.2

4.4. Ablation Study

Finally, we explore the impact of different parameter settings on the model performance. Taking the self-attention head as an example, Table 4 shows the comparative performance of our model under different numbers of heads. It can be seen that the increase in accuracy is small, which indicates that increasing the number of self-attention heads in the encoder module can improve the performance, but not significantly. In addition, we train the model with 12 heads for about 5.5 h more than that with six

heads. Although we choose 12 heads in order to achieve the highest performance in this work, we suggest that researchers balance the trade-off between the speed and performance.

Table 4. Comparison of the training performances under different parameter settings.

Number of Encoders	Training Time	Training Loss	Evaluation Loss	Accuracy
6	3 h, 21 m, 30 s	0.036	0.038	96.10%
8	5 h, 15 m, 12 s	0.032	0.030	96.23%
10	7 h, 11 m, 37 s	0.029	0.031	96.37%
12	8 h, 59 m, 42 s	0.028	0.029	96.41%

4.5. Discussion

Geolocating textual addresses is an important task in various LBS applications. Previous studies tried to predict the coordinates in a regression fashion or predict a discrete region by multi-classification. However, they all suffered from too large output space. By contrast, even for a person, the intuitive way of geolocating a textual address is to correlate a series of regions with different scales based on the hierarchical geographic information implied in the address. This motivates us to consider a coarse-to-fine geolocating approach. Specifically, this paper opens up a new paradigm for geolocation prediction, i.e., predicting a series of hierarchical regions in a Seq2Seq fashion. Its strength lies in taking full advantage of the inherent hierarchy information in Chinese addresses without relying on any additional information beyond the texts. Moreover, the discrete global grid system, GeoSOT, provides a globally unified benchmark for hierarchically discretizing the earth's surface. Without relying on any external gazetteers, we sequentially predict a GeoSOT code which exactly represents a set of regions from coarse to fine.

We think that there are at least three limitations and opportunities for new use. First, although our approach focuses on Chinese addresses, it is possible to be generalized to more types of geographical texts, e.g., Weibo or travel notes. Theoretically, these datasets can be directly trained with our proposed model. We take this as one of our future works. Second, the GeoSOT grid code can be replaced by any other type of geocodes such as GeoHash [35] and Google S2 [19]. Considering that different LBS applications use different geocoding methods, we plan to support user-defined coding methods in the future. Third, this approach is expected to be intelligent enough to predict the granularity. In this work, we take the explicit control of the granularity of the predicted region (e.g., L13, L15, L17). However, we believe that it is more intelligent to predict the granularity by the model itself according to the input data. This is because different applications and the amount of original information contained in the input data will affect the final prediction granularity. We plan to take these factors into account and adapt our model to predict the GeoSOT codes of variable lengths.

5. Conclusions

In this paper, we introduce a novel coarse-to-fine model for geolocating Chinese addresses. Our proposed approach first models the geolocation prediction as a Seq2Seq learning task, and then develops a deep learning-based neural network to solve it. Without any additional information beyond texts or external gazetteers, our model takes the textual address as input and outputs the GeoSOT grid code that exactly represents a series of hierarchical regions covered by the address. Compared with previous studies, our method effectively narrows the prediction space. The experimental results in terms of distinguishing polysemy and geolocation accuracy demonstrate the significant advantages of our model in the geolocation task.

Author Contributions: Chunyao Qian conceived, designed, and performed the experiments and wrote the manuscript; Chao Yi collected the dataset and reviewed the manuscript; Jiashu Liu polished the language; Chengqi Cheng supervised the study; and Guoliang Pu offered helpful suggestions and revised the manuscript critically. All authors have read and approved of the submitted manuscript, have agreed to be listed, and have accepted this version for publication.

Funding: This research was supported by the National Key Research and Development Program of China (Grant No. 2018YFB0505300 and Grant No. 2017YFB0503703).

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

References

1. Mahmud, J.; Nichols, J.; Drews, C. Where Is This Tweet From? Inferring Home Locations of Twitter Users. *ICWSM* **2012**, *12*, 511–514.
2. Wing, B.; Baldridge, J. Hierarchical discriminative classification for text-based geolocation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 336–348.
3. Huang, B.; Carley, K.M. A Hierarchical Location Prediction Neural Network for Twitter User Geolocation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 4734–4744.
4. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
5. Jin, A.; Cheng, C. Spatial Data Coding Method Based on Global Subdivision Grid. *J. Geomat. Sci. Technol.* **2013**, *30*, 284–287.
6. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM neural networks for language modeling. In Proceedings of the Thirteenth Annual Conference of the International Speech Communication Association, Portland, ON, USA, 9–13 September 2012.
7. Fink, C.; Piatko, C.; Mayfield, J.; Chou, D.; Finin, T.; Martineau, J. The geolocation of web logs from textual clues. In Proceedings of the 2009 International Conference on Computational Science and Engineering, Vancouver, BC, Canada, 29–31 August 2009; Volume 4, pp. 1088–1092.
8. Chi, L.; Lim, K.H.; Alam, N.; Butler, C.J. Geolocation prediction in twitter using location indicative words and textual features. In Proceedings of the 2nd Workshop on Noisy User-Generated Text (WNUT), Osaka, Japan, 11 December 2016; pp. 227–234.
9. Liu, B.; Yuan, Q.; Cong, G.; Xu, D. Where your photo is taken: Geolocation prediction for social images. *J. Assoc. Inf. Sci. Technol.* **2014**, *65*, 1232–1243. [CrossRef]
10. Rahimi, A.; Cohn, T.; Baldwin, T. Semi-supervised user geolocation via graph convolutional networks. *arXiv* **2018**, arXiv:1804.08049.
11. Miura, Y.; Taniguchi, M.; Taniguchi, T.; Ohkuma, T. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July– 4 August 2017; pp. 1260–1272.
12. Wing, B.; Baldridge, J. Simple supervised document geolocation with geodesic grids. In Proceedings of the 49th Annual Meeting of the Association For Computational Linguistics: Human Language Technologies, Portland, ON, USA, 19–24 June 2011; pp. 955–964.
13. Roller, S.; Speriosu, M.; Rallapalli, S.; Wing, B.; Baldridge, J. Supervised text-based geolocation using language models on an adaptive grid. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 13 July 2012; pp. 1500–1510.

14. Han, B.; Cook, P.; Baldwin, T. A stacking-based approach to twitter user geolocation prediction. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Sofia, Bulgaria, 4–9 August 2013; pp. 7–12.
15. Han, B.; Cook, P.; Baldwin, T. Text-based twitter user geolocation prediction. *J. Artif. Intell. Res.* **2014**, *49*, 451–500. [CrossRef]
16. Rout, D.; Bontcheva, K.; Preoţiuc-Pietro, D.; Cohn, T. Where's@ wally? A classification approach to geolocating users based on their social ties. In Proceedings of the 24th ACM Conference on Hypertext and Social Media, Paris, France, 1–3 May 2013; pp. 11–20.
17. Rahimi, A.; Vu, D.; Cohn, T.; Baldwin, T. Exploiting text and network context for geolocation of social media users. *arXiv* **2015**, arXiv:1506.04803.
18. Dredze, M.; Osborne, M.; Kambadur, P. Geolocation for twitter: Timing matters. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1064–1069.
19. Kulkarni, S.; Jain, S.; Hosseini, M.J.; Baldridge, J.; Ie, E.; Zhang, L. Spatial Language Representation with Multi-Level Geocoding. *arXiv* **2020**, arXiv:2008.09236.
20. Bengio, Y.; Ducharme, R.; Vincent, P.; Jauvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
21. Mikolov, T.; Karafiát, M.; Burget, L.; Cernocky, J.; Kombrink, S. Recurrent neural network based language model. In Proceedings of the Eleventh Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, 26–30 September 2010; pp. 1045–1048.
22. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; pp. 3111–3119.
23. Dai, A.M.; Le, Q.V. Semi-supervised sequence learning. In Proceedings of the Advances in Neural Information Processing Systems, Center, MO, Canada, 7–12 December 2015; pp. 3079–3087.
24. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
25. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (accessed on 17 October 2020).
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
28. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv* **2019**, arXiv:1909.11942.
29. Lawder, J.K. The Application of Space-Filling Curves to the Storage and Retrieval of Multi-Dimensional Data. Ph.D. Thesis, Citeseer, London, UK, 2000.
30. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.
31. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
32. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
33. Maaten, L.v.d.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
34. Cheng, Z.; Caverlee, J.; Lee, K. You are where you tweet: A content-based approach to geo-locating twitter users. In Proceedings of the 19th ACM international conference on Information and knowledge management, Toronto, ON, Canada, 26–30 November 2010; pp. 759–768.

35. Pao, Y.; Kodesh, S.; Chopra, J.; Fan, K.; Lin, C.; Shen, L.; Levi, D.; Haque, A.; Baz, Z.E. Geohash-Related Location Predictions. U.S. Patent 9,894,484, 13 February 2018.