applied sciences

MDPI

Article

Deep Transfer Learning Model for Semantic Address Matching

Liuchang Xu [1,2,3,4], Ruichen Mao [5], Chengkun Zhang [6], Yuanyuan Wang [7], Xinyu Zheng [1,3,4], Xingyu Xue [1,3,4] and Fang Xia [1,8,*]

1 College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China
2 College of Computer Science and Technology, Zhejiang University, Hangzhou 310063, China
3 Key Laboratory of Forestry Intelligent Monitoring and Information Technology of Zhejiang Province, Zhejiang A&F University, Hangzhou 311300, China
4 Key Laboratory of State Forestry and Grassland Administration on Forestry Sensing Technology and Intelligent Equipment, Zhejiang A&F University, Hangzhou 311300, China
5 Zhejiang Laboratory, Hangzhou 311121, China
6 School of Earth Sciences, Zhejiang University, Hangzhou 310027, China
7 Ocean Academy, Zhejiang University, Zhoushan 316021, China
8 Institute of Digital Village, Zhejiang A&F University, Hangzhou 311300, China
* Correspondence: xiaf@zafu.edu.cn

Abstract: Address matching, which aims to match an input descriptive address with a standard address in an address database, is a key technology for achieving data spatialization. The construction of today's smart cities depends heavily on the precise matching of Chinese addresses. Existing methods that rely on rules or text similarity struggle when dealing with nonstandard address data. Deep-learning-based methods often require extracting address semantics for embedded representation, which not only complicates the matching process, but also affects the understanding of address semantics. Inspired by deep transfer learning, we introduce an address matching approach based on a pretraining fine-tuning model to identify semantic similarities between various addresses. We first pretrain the address corpus to enable the address semantic model (abbreviated as ASM) to learn address contexts unsupervised. We then build a labelled address matching dataset using an address-specific geographical feature, allowing the matching problem to be converted into a binary classification prediction problem. Finally, we fine-tune the ASM using the address matching dataset and compare the output with several popular address matching methods. The results demonstrate that our model achieves the best performance, with precision, recall, and an F1 score above 0.98.

Keywords: semantic address matching; deep transfer learning; pretraining model; fine tuning

Citation: Xu, L.; Mao, R.; Zhang, C.; Wang, Y.; Zheng, X.; Xue, X.; Xia, F. Deep Transfer Learning Model for Semantic Address Matching. *Appl. Sci.* **2022**, *12*, 10110. https://doi.org/10.3390/app121910110

Academic Editors: Alexei Gvishiani and Boris Dzeboev

Received: 7 August 2022
Accepted: 5 October 2022
Published: 8 October 2022

1. Introduction

Addresses are used to describe a unique spatial location on Earth and are usually expressed in the form of an addressing system [1]. In recent years, with the rapid development of location services, massive amounts of industry data based on addresses as spatial information have started to emerge. Address matching is a crucial application in address services, which compares addresses with the same location in different address databases to obtain the best match with the search address and to determine position on a map [2]. Traditional address matching technology is challenged by the prevalence of high-precision address matching in urban industries, such as logistics and online taxi services. Therefore, an effective address matching method is required to facilitate the provision of accurate and efficient intelligent spatial location services and to promote the development of smart cities.

The pattern of arrangement of address elements varies from country to country. For instance, the US address pattern is "room number + street + state + country", and it performs well in creating a national geodatabase [3]. Japanese addresses, on the other hand, are coded based on location and geographic relativity, with the overall order being the opposite of the address pattern used in the US, and generally without "streets" [4]. In

general, the address patterns of the above countries are nested and relatively standardized. However, Chinese addresses are relatively more difficult to match due to their complex context and rules, mainly due to the following reasons: (1) Chinese addresses are written without separators; (2) Chinese addresses often contain landmarks or POI and topology (e.g., road intersections); (3) different government departments manage addresses, leading to confusion; and (4) address assignment and updating lags behind rapid urban renewal [5]. The data objects of this study are Chinese addresses.

Address matching is generally divided into matching based on rule-based or statistical methods and semantic similarity matching based on machine learning and deep learning. Character-based approaches match addresses by calculating their string similarity metrics and then manually establishing a threshold or a particular classifier to identify a match [6,7]. String similarity metrics include edit distance and its variants [8–10], Jaccard similarity metric [11], and Jaro distance and its variants [12,13]. Among them, Santos et al. compared 13 different string similarity metrics for place name matching and found that adjusting the similarity threshold was the key to achieving good performance [14]. In addition, the calculation of cosine similarity between embeddings based on N-grams is also a common method [15,16]. This method has better performance compared with traditional metrics. Recently, Yong et al. proposed a normalization method based on the Euclidean distance between the address to be processed and the address in the standard library, but it is only applicable to some specific datasets [17]. Another type of methods is address element based, which segments out address elements by rules or statistical methods and then compares the address elements and their hierarchy to determine whether they match [18–20]. Lin et al. point out that the degree of matching of address elements depends on whether they can be extracted correctly [21]. In general, dictionary queries [22], probability statistics such as CRF [23] and HMM [24], and creating matching rules [25,26] are the basic ways for retrieving address elements. Another common method is to construct a decision tree consisting of matching rules, each corresponding to a path in the tree. Kang et al. proposed an address matching tree model based on the analysis of the spatial constraint relationship between address elements; this requirement makes the address model more complex [27]. Focusing on the wrong word separation problem, Luo and Huang suggested a method based on a trie tree and finite-state machine [28]. The aforementioned techniques, however, frequently struggle when dealing with nonstandardized (missing address elements or represented by POI) and complexly structured addresses (such as the Chinese address feature aforementioned).

In recent years, the area of artificial intelligence has seen tremendous progress in natural language processing (NLP), most of which is attributable to deep learning's enhanced performance. Word2vec [29], ELMo [30], GPT [31], BERT [32], XLNet [33], ERNIE [34], and ELECTRA [35] are a few of these classical language models. Since addresses as special textual descriptions, more and more studies in address matching has also introduced natural language models based on deep learning [36]. Cruz et al. analyzed 41 papers on address matching published between 2002 and 2021 and discovered that most of the relevant studies have used deep learning methods. Among them, consistent with the above in this paper, due to the complexity of Chinese addresses, Chinese address matching accounted for half of the studies [37].

Comber et al. used CRF and word2vec for address matching to extract the semantics of addresses without designing complex rules [38]; Zhang et al. provides a convolutional neural network (W-TextCNN) for Chinese address pattern classification [39]. With the popularity of gating mechanism neural networks, address matching and normalizing based on LSTM and GRU have been carried out by an increasing number of researchers [40–43]. Santos et al. used a deep neural network based on bidirectional GRUs for place name matching [44]; Shan enriched the address context by collecting address data on the Internet and trained an address representation model with two LSTMs and attention mechanisms to extract address vectors [45]. While Li et al. incorporated the hierarchical relationship between address elements into a neural network and proposed a BiLSTM-based multitask

learning method [46], Chen et al. proposed a contrast learning address matching model based on attention-Bi-LSTM-CNN networks (ABLC) [47]. Subsequently, more and more researchers have used the attention mechanism in their address matching models [48–50]. With the popularity of pretrained language models, Lin et al. used the classical enhanced sequence inference model (ESIM) [51] for address record pair modelling [21], whereas Xu et al. and Qian et al. used the BERT model. Xu et al. proposed a BERT-based model for extracting address semantic representations to achieve the fusion of address semantics and geospatial information [36]; Qian et al. combined BERT and LSTM, and proposed a hierarchical region-based approach for geolocation of Chinese addresses [52]. However, all of the aforementioned methods require the extraction of address semantic features to embedding, and this can affect the effectiveness of address semantic understanding, as has been demonstrated in the field of NLP [32].

In summary, when dealing with nonstandardized addresses with complicated structures, the aforementioned approaches still lack a level of comprehension of address semantics, which negatively impacts the accuracy of address matching.

To address the above problems, we use a deep transfer learning approach. First, we pretrain an addresses corpus so that our address semantic model (abbreviated as ASM) can learn unsupervised address contexts to better understand address semantics. Then, we use the address-specific geospatial property to build a labelled address matching dataset, allowing the matching problem to be converted into a binary classification prediction problem. Finally, fine-tuning the ASM with the address matching dataset allows the model to improve its performance significantly.

The contributions of this paper are as follows: (1) A neural network based on a multihead self-attention mechanism and a permutation-based target task is used to train the ASM for a large-scale corpus in an unsupervised automated manner. The ASM can learn address semantics better. (2) A deep transfer learning approach is used to achieve semantic address matching by fine-tuning the ASM, which improves the matching accuracy. (3) A semantic address matching dataset construction method is proposed to convert address matching into a classification prediction task. The method constructs an address matching dataset with labels using location information as the inference condition. (4) Results demonstrate that with the transfer learning approach, a better-performing downstream task such as address matching can also be achieved with microsupervision.

The remainder of this paper is organized in four sections. Section 2 introduces the materials used in our study, as well as the data processing procedures. The methodology adopted is also demonstrated in Section 2, including the pretraining and fine-tuning based on XLNet. The results of our experiments are analyzed in Section 3. Section 4 presents our conclusions and the future work of this study.

2. Materials and Methods

In this section, we introduce a deep transfer learning approach in NLP and propose a semantic address matching framework. First, we tokenize all address data to be used as model input in the pretraining phase. Then, we use the XLNet model [33] to pretrain the address corpus and make the model understand the address semantics by learning contextual information. Finally, we construct a supervised dataset for semantic address matching, fine-tune the pre-trained ASM for address matching, and compare it with multiple models to evaluate the accuracy of the ASM.

2.1. Dataset

Address records for the raw data were manually collected in 2019 from various government departments. The geographical area to which this data refers is Shangcheng District, Hangzhou, Zhejiang Province, China. The address dataset contains a variety of location description types, including standard addresses, nonstandard addresses, POIs, road intersections, place name abbreviations, and so on. The preprocessed address dataset

amounted to 1,552,532, consisting of three fields of address records, longitude, and latitude, which served as the address corpus for the pretraining phase.

To use the address data for semantic address matching, we created a dataset of address pairs with labels based on the address corpus. Based on the set of addresses filtered with the same coordinates, we performed manual matching using a standard address database. In addition, to give the model better prediction performance and generalization capabilities, we augmented the dataset with easy data augmentation [51] methods for text classification tasks, mainly using synonym replacement, address element deletion, and address element insertion. To improve the robustness of the model, we constructed mismatched address pairs in the set of address pairs with Jaccard similarity coefficients [11] greater than zero. We finally obtained a dataset of 64,358 address pairs and corresponding labels, a sample of which is shown in Table 1. The statistical features of the dataset used for semantic address matching are shown in the Table 2, where we used the difference in the number of characters, Levenshtein distance [8], and Jaccard similarity coefficient [11] to show the similarity of address pairs in the dataset. Unmatched address pairs will perform worse in terms of text similarity, in line with our common sense.

Table 1. Examples of some data in the labelled address dataset.

Address S_a	Address S_b	Label
Block 3, No. 15 Haiyue Road, Hangzhou City (杭州市海月路15号3幢)	Block 3, Haiyue Garden Residential Unit, Shangcheng District, Hangzhou (杭州上城区海月花园3幢)	1
Block 3, No. 15 Haiyue Road, Hangzhou City (杭州市海月路15号3幢)	Hangzhou Haiyue Bathing Centre (杭州海月洗浴中心)	0

Table 2. Statistical characteristics of the labelled address dataset.

Statistical Characteristics	Value
Total number of address pairs	64,358
Number of matching address pairs	32,179
Number of unmatched address pairs	32,179
Average length difference for all the address pairs	4.76
Average length difference for matching address pairs	3.73
Average length difference for unmatched address pairs	5.79
Average Levenshtein distance for all the address pairs	12.16
Average Levenshtein distance for matching address pairs	7.28
Average Levenshtein distance for unmatched address pairs	17.05
Average Jaccard similarity coefficient for all the address pairs	0.46
Average Jaccard similarity coefficient for matching address pairs	0.68
Average Jaccard similarity coefficient for unmatched address pairs	0.24

2.2. Semantic Address Matching Definition

In this paper, we study the address matching in the absence of a standard address database, referred to as the semantic address matching task. The following description defines semantic address matching:

Given the address dataset: $D = \{add_1, add_2, \ldots, add_n\}$, the goal of semantic address matching is to find each address pair: (add_i, add_j), satisfying $add_i \doteq add_j$, where $add_i \in D$, $add_j \in D$, $i \neq j$, and $\doteq$ represent the comparison operator. The operation objects on either side of the comparison operator refer to the same real-world object with the same coordinates. It is important to note that no information other than the string itself and its corresponding geospatial information is utilized in this study to calculate the similarity of two addresses. Therefore, the task addressed in this study focuses on the problem of matching addresses with the same location instead of address disambiguation. In addition, due to the many different representations of the same location, we believe that it is not possible to achieve a correct match without processing from a natural language understanding

perspective. Therefore, in our study, "address semantic understanding" refers to the textual understanding of the address corpus, while "address semantic reasoning" used for address matching is based on the spatial relationship reasoning of addresses.

2.3. Pretraining Phase Using the Address Corpus Based on XLNet

This section presents a transfer learning-based pretraining model for address semantics: address semantic model (ASM). The ASM is based on the characteristics of Chinese addresses, combined with the advantages of semantic understanding in deep learning natural language models. The model takes as input a single character of a Chinese address that has been tokenized, and uses a multihead self-attention-based semantic extraction module to help the model understand the semantics of the address with the objective of permutation unknown character prediction. For the practical training problem resulting from the prediction objective, a two-stream self-attention structure for target position representations is used. The overall structure of the ASM is shown in Figure 1.

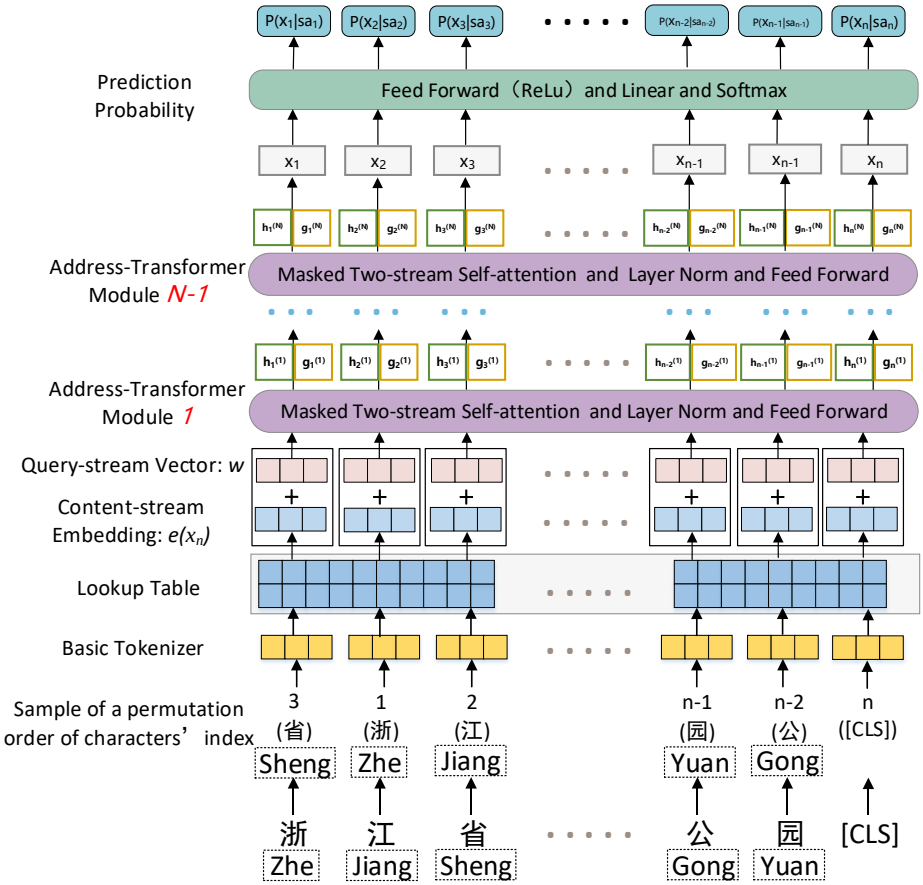

Figure 1. The overall structure of the ASM.

2.3.1. Tokenization of Address Characters

The conversion of Chinese addresses into input that can be received by the ASM is the basis for training. Since Chinese addresses are not like alphabetic forms of languages, such as English, they do not have delimiters. Therefore, most Chinese address studies start with the segmentation of address elements. Due to the unique hierarchy of addresses, partitioning addresses into various address elements is already a problem worth studying. Our study, however, aims to convert the complex address matching into a classification problem that can be automated for computer computation. Although the commonly used SentencePiece method [53] in NLP can automate the segmentation of Chinese addresses by counting high-frequency co-occurring characters combined into subword units and constructing dictionaries, the subwords obtained by its segmentation are too long, and

some of the segmented words do not conform to the common sense of Chinese addresses, which will affect the semantic understanding during pretraining.

We therefore use the Basic Tokenizer, which tokenizes a character as a unit. It separates words and symbols according to spaces. We first add blank characters before and after each character of the address. Then the characters are matrix-transformed according to the lookup table to become the input of the one-hot encoding, and the activated dimensions in the one-hot encoding are the index number corresponding to the character in the dictionary key–value pair. In this study, two dictionaries—one with non-Chinese characters and the other with solely Chinese characters—are created once the individual characters from each address have been obtained. These dictionaries have 9425 and 3491 characters, respectively.

2.3.2. Objective of Permutation Unknown Character Prediction and Two-Stream Self-Attention Structure

The objective of permutation language modeling is derived from the XLNet model [30]. Without altering the character order of the original text, the target employs rearrangement to sabotage the index order of text descriptions. This training target not only preserves the high-order and long-range dependencies present in the text context, but also improves on the disadvantages of past autoregressive language modeling's targets that could only exploit unidirectional contexts (forward or backward), enabling a pretrained model to utilize deep, bidirectional contextual information more effectively. Addresses, as special natural languages incorporating geospatial information and hierarchy, need to fully utilize the bidirectional contextual information, so we use a permutation language model objective for pretraining the address corpus.

Specifically, we assume that given an address record X of length T, there are a total of T! sequences of permutations. If all permutations are traversed and the parameters of the model are shared, then the model must be able to learn the context of all positions. We take a simplified address record, for example, "Hangzhou Underwater World" ("Hang Zhou Hai Di Shi Jie" in Chinese pinyin), and predict the third character "Hai" in a different order, as shown in Figure 2. In Figure 2b, for instance, the address permutation is disordered as 3→2→4→1→6→5, so when predicting "Hai", there is no address context character, and the prediction can only be made based on the previous hidden state. For Figure 2f, the "Hai" (3) character learns all five context characters except itself.

The objective function of XLNet is to maximize the log-likelihood function of the target subsequence conditional on the nontarget subsequence:

$$\max_{\theta} \mathrm{E}_{z\sim Z_T}[\log p_\theta(X_{z>c}|X_{z\leq c})] = \mathrm{E}_{z\sim Z_T}\left[\sum_{t=c+1}^{|z|} \log p_\theta(x_{z_t}|X_{Z<t})\right] \quad (1)$$

where Z_T denotes the set of all permutations of the index of an address record of length T; $z \in Z_T$ is one of the sequences of indexed permutations, where z_t denotes the t-th element of the sequence of indexed permutations, and $z_{<t}$ denotes the first t-1 elements of z; $\mathrm{E}_{z\sim Z_T}$ denotes the maximum Expectation; and p_θ denotes the predicted probability. In addition, XLNet used the partial prediction optimization. It slices a permutation z into two subsequences, $z_{\leq c}$ and $z_{>c}$, where c is the slice point that slices the two subsequences into a nontarget sequence and a target sequence, respectively.

While the above objective of permutation unknown character prediction works well for understanding address semantic by removing ambiguity from the target prediction, it creates the problem that the model does not know the position of the character to be predicted in the original address record. Therefore XLNet [33] introduces a two-stream self-attention structure to let the model know where the character to be predicted is located explicitly, which consists of two sets of hidden representations instead of one. The two streams of representations are updated with a shared set of parameters as follows:

$$g_{z_t}^{(m)} \leftarrow Attention(Q = g_{z_t}^{(m-1)}, KV = h_{z<t}^{(m-1)}; \theta), (query\ stream : use\ z\ but\ cannot\ see\ x_{z_t}) \quad (2)$$

$$h_{z_t}^{(m)} \leftarrow Attention(Q = h_{z_t}^{(m-1)}, KV = h_{z\leq t}^{(m-1)}; \theta), (content\ stream : use\ both\ z_t\ and\ x_{z_t}) \quad (3)$$

where Q, K, V denote the query, key, and value in an attention operation [54]; h_{z_t} denotes the content representation, which serves a similar role to the standard hidden states in Transformer, and g_{z_t} denotes the query representation, which only has access to the contextual information $X_{z<t}$ and the position X_{z_t}, but not the content X_{z_t}.

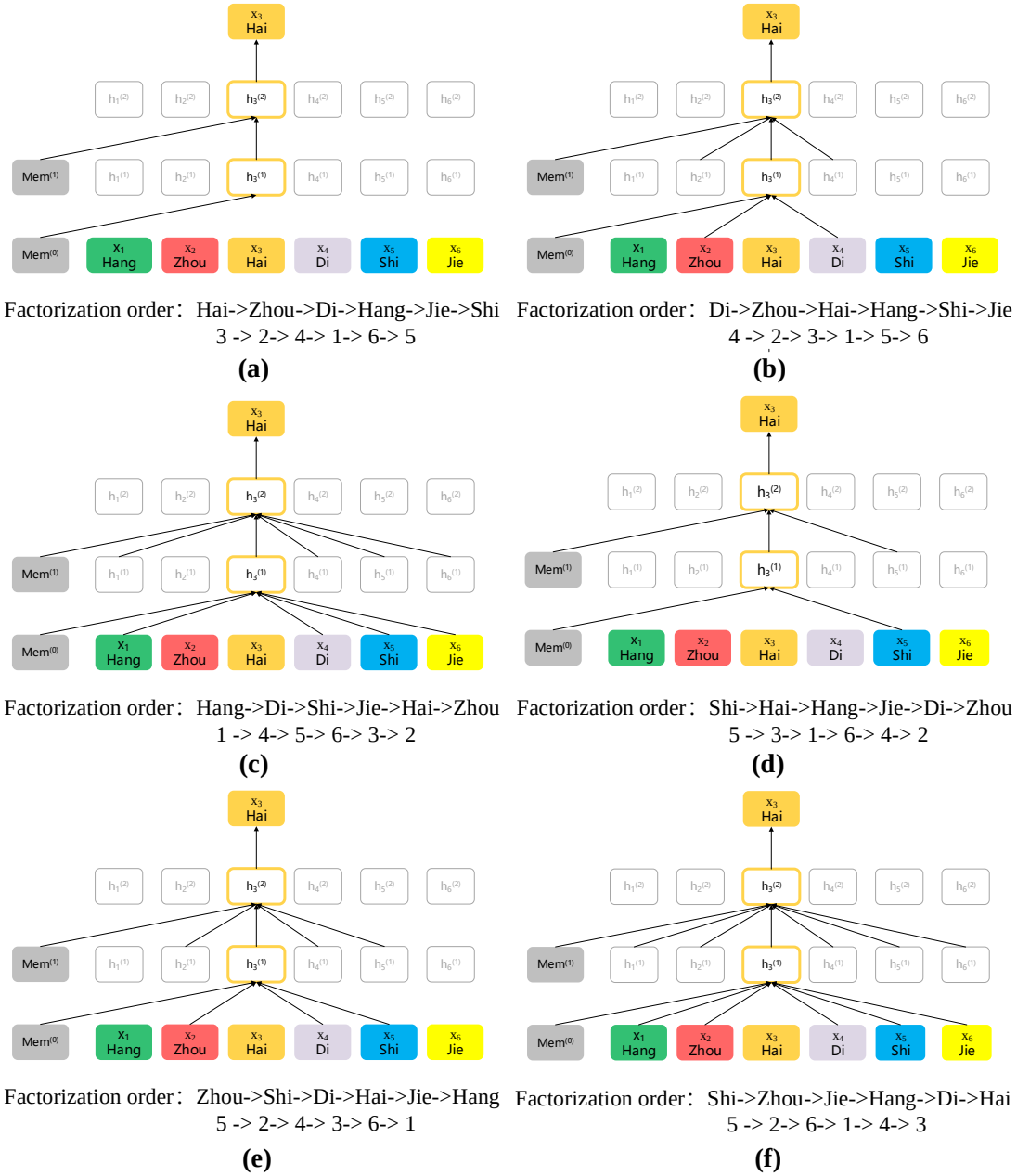

Figure 2. Illustration of the permutation language modeling objective for predicting "Hai" given the same simplified address record but with different factorization orders.

In addition, we employ Transformer-XL with a multihead self-attention mechanism as an address semantic feature extractor [55]. Transformer-XL integrates two important techniques, namely, the relative positional encoding scheme and the segment recurrence mechanism. This allows for better adaptation to the two-stream attention permutation

language model. As the number of semantic feature extraction structures affects the performance of the model in subsequent experiments, each layer of the Transformer-XL module is tentatively defined in this section as the address-transformer module.

2.4. Fine-Tuning for Semantic Address Matching

Fine-tuning is an implementation of deep transfer learning, which refers to adding task-relevant structures and parameters to an already-trained model, and then retraining on a task-relevant corpus [56]. We therefore used a newly constructed labelled address matching corpus for the semantic address matching, adding a new neural network structure for a fine-tuned learning model and training framework based on the classification task. The network structure is first superimposed with a layer of fully connected feedforward neural networks for nonlinear transformation, with an activation function of *tanh*, which is mathematically formulated as follows:

$$\mathrm{t}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4}$$

After obtaining the probability distribution features using the fully connected neural network, we then connected the fully connected neural network without the activation function for linear transformation. Since semantic address matching is a binary classification task of whether to match, the output of this layer is two-dimensional. Finally, we passed the output probability distribution score of this layer into the SoftMax normalization function to predict the probability of matching or not matching the address pair, respectively. We designed the deep semantic address matching model (abbreviated as DSAMM) with the following objective function. Here, given that the size of the number of address string pairs per batch iteration is *batch_size*, the predicted probability output is *prob(batch_size,2)*, and the true label sequence is *label(batch_size)*, the true label probability for each address pair is as follows:

$$true\_prob(batch\_size) = gather(prob, Label) \tag{5}$$

The final objective function is obtained by taking logarithmic values of the probabilities and then summing them (i.e., log transformation) and averaging them. The objective function is specified below:

$$\frac{\sum_{i=1}^{batch\_size} \log(true\_prob[i])}{batch\_size} \to 0 \tag{6}$$

The accuracy metrics used in this study include precision, recall, and F1 score [57]. Precision calculates the proportion of true positive samples out of those predicted to be positive; recall reflects the rate at which positive examples in this are predicted to be accurate and, in semantic address matching, refers to the percentage of correctly matched pairs out of all address pairs that should be correctly matched; and the F1 score is the harmonic mean of precision and recall.

3. Results and Discussion

3.1. Address Semantic Model Pretraining

We examine the semantic understanding effectiveness of the ASM by examining the prediction accuracy of address characters for permutation language objective. In the experimental design, we refer to the influencing factors of pretraining in a study by Xu et al. and use the number of address-transformer modules and whether the numbers in the address records are replaced with uniform identifiers as the independent variables for the analysis of the pretraining hyperparameters [36]. The purpose of the experiments in this section is to validate the effectiveness of the ASM without testing or predicting it, so only the training set and the verification set are required for pretraining the model. Due to the large address corpus, we set the proportions of training set and verification set to be approximately 99% (1,537,532) and 1% (15,000), respectively.

The optimal values for the relevant hyperparameters were determined by drawing on previous studies of pretrained language models and previous experiments. It is shown

in Table 3. In terms of the number of address-transformer modules, we set the number of modules to 6, 8, 10, and 12 to observe the performance of the target task in different situations. As shown in Figure 3, the training loss gradient for different numbers of modules decreases rapidly until about 40k steps, and then keeps decreasing slowly and gently in the following iterations, and basically levels off in the last 40k steps, indicating that the ASM instances have been adequately trained. Additionally, the comparison of each ASM instance reveals that the positions of the four curves overlap more, which indicates that the number of address-transformer modules has little influence on prediction performance. As shown in Table 4, the accuracy of the model validation under the above four comparisons ranged from 90.5% to 91.5%, and the target accuracy increased slightly with the number of modules, indicating that the more the number of address-transformer modules, the better understanding of address semantic. However, due to the tiny increase in accuracy and the large increase in training time, building six layers of modules was the most cost-effective option. The findings of this study are consistent with those of Xu et al. [36].

Table 3. Setting values for each hyperparameter of the ASM.

Hyperparameters	Are Numbers Replaced with "CODE|"	
	Replaced	**Not Replaced**
num_address_transformer_module	12	6, 8, 10, 12
batch_size	32	32
hidden_size	768	768
num_head	4	4
dim_head	8	8
FFN_hidden_size	2048	2048
Dropout	0.1	0.1
K	3	3
sequence_len	32	32
mem_len	24	24
η	0.00005	0.00005
epoch	4	4
voc_size	3491	9425

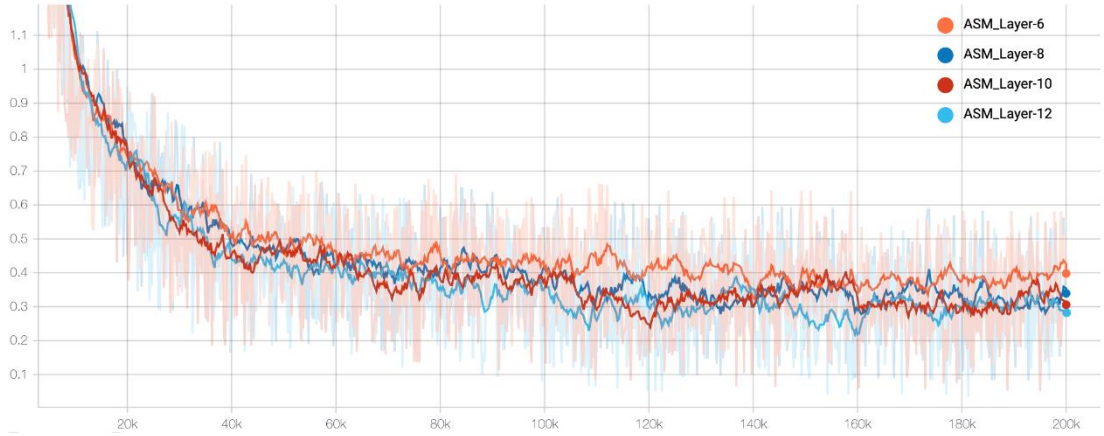

Figure 3. Comparison of training loss gradient curves for the ASM with a different number of address-transformer modules, where the curve smoothness is set to 0.9, and the opaque curve is after the smoothing setting and the translucent curve is the original loss gradient curve.

Table 4. Values for the ASM indicator for a different number of address-transformer modules.

Number of Address-Transformer Modules	Training Time	Training Loss	Mean Evaluation Loss	Accuracy
6	6 h 19 m 36 s	0.3864	0.3615	90.62%
8	8 h 21 m 54 s	0.3395	0.3258	90.98%
10	10 h 23 m 5 s	0.3016	0.3003	91.21%
12	12 h 32 m 45 s	0.2831	0.2793	91.47%

Considering that most of the numbers in the address records have only geospatial property and no semantic information (e.g., "No. 116 Tianmushan Road" and "No. 226 Tianmushan Road", there is no difference in their contexts other than numbers, so the model cannot make accurate predictions at all. Therefore, we replaced all Arabic numerals in the address corpus with a uniform identifier: "CODE". Since Xu et al. [36] used BERT [32] to construct the ALM for pretraining the address corpus, we used their method for comparison. Figure 4 shows a comparison of the training loss gradient curves of the replaced and unreplaced "CODE" corpus under the ASM. The training loss curve of the model with the "CODE" replacement is always below the original address corpus, demonstrating that the numbers confound the predictive target of the model and reduce the predictive power of the model. As can be seen from Table 5, the prediction accuracy of the ASM increased by approximately 7 percentage points after the replacement with "CODE". In addition, when compared with the ALM, our model's prediction accuracies all improved, indicating that the ASM performs better in understanding address semantics.

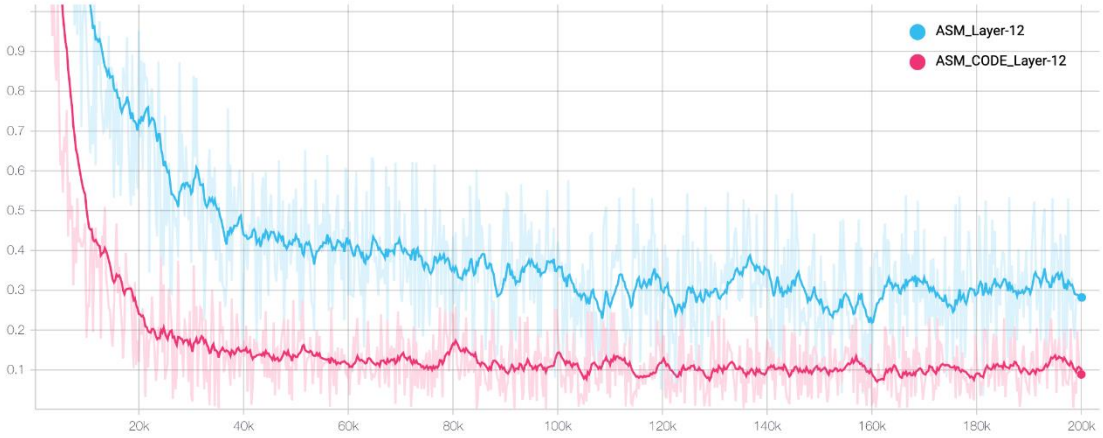

Figure 4. Comparison of training loss gradient curves for the ASM with Arabic numeric code replacement and without replacement, where the curve smoothness is set to 0.7, and the opaque curve is after the smoothing setting and the translucent curve is the original loss gradient curve.

Table 5. The prediction accuracy of the ASM and the ALM in the validation datasets.

Prediction Accuracy	Without "CODE" Replacement	"CODE" Replacement
ALM [40]	90.31%	97.22%
ASM	91.47%	98.53%

3.2. Fine-Tuning for Semantic Address Matching

To explore whether there is overfitting, we tested the validation dataset by fine-tuning it while also spacing the number of iterations by 500, and determined the set of hyperparameters that worked best. In this study, 80% of the data were randomly taken as the training dataset and the remaining 20% as the validation dataset. The loss gradient curves for training and validation are shown in Figure 5.

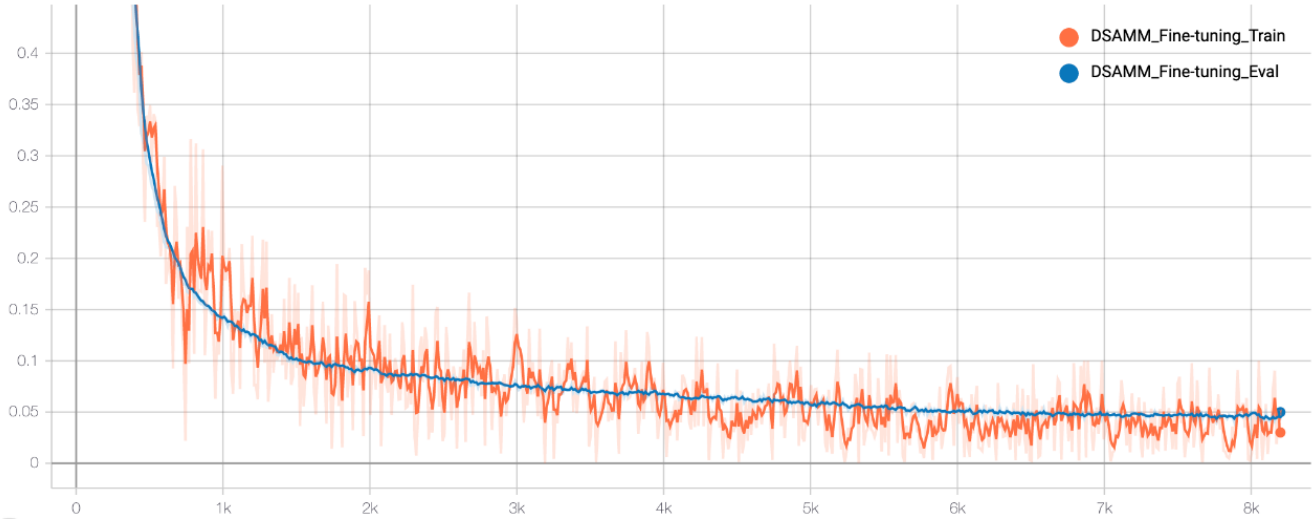

Figure 5. Training loss-gradient curve overlaid with validation loss-gradient curve of the DSAMM, where the curve smoothness is set to 0.7, and the opaque curve is after the smoothing setting and the translucent curve is the original loss gradient curve.

Figure 5 shows that the training loss value of the DSAMM decreases rapidly before the steps of training iterations are 1k, indicating the model's excellent learning ability and ability to make a good assessment of whether address pairs match even before the dataset is entirely learned. This has inspired us to investigate whether better matching can be achieved even with microsupervision, that is, with a small amount of supervised training data.

Between 2k and 6k training iteration steps, the training loss value decreases more gently and steadily, indicating that the model is still learning the task objectives for the supervised data. Between the final 6k and 8k, the training loss values have largely flattened out, demonstrating that the DSAMM instance has been sufficiently learned by that number of iterations to warrant further training. The final training loss values ranged from 0.01 to 0.05, indicating that the fine-tuning training was effective, and the exact metric values will be discussed further in the subsequent comparative experimental analysis. In addition, the trend of the loss values is like that of the training loss values, with a "high rate of decline—slow decline—gradual levelling off". The final loss values for the validation datasets range from 0.04 to 0.05, indicating that there is no overfitting. We selected the optimal iteration step of the validation datasets as the DSAMM instance after training 8 epochs.

We set up comparative experiments with various gradients of the proportion of training sets to examine whether the DSAMM only needs a limited number of labelled training datasets to attain high matching precision. Figure 6 displays the matching prediction precision with 1%, 5%, 10%, 15%, and 20% of the original training datasets. When employing only 15% of the initial training set of labelled address pairings, the DSAMM achieves a matching precision of above 0.80, and 0.84 when using 20%. It demonstrates that the DSAMM can perform well in weakly supervised learning, most likely due to the transfer learning employed in our study. Our model has fully understood the address semantics using self-supervised learning in the pretraining phase, and when then fine-tuned for task-based learning using partially supervised data, it is able to combine the advantages of the two phases mentioned above, allowing the model to perform at a high level on the task with less supervised training data. This also coincides with the research that first proposed the idea of pre-training [58].

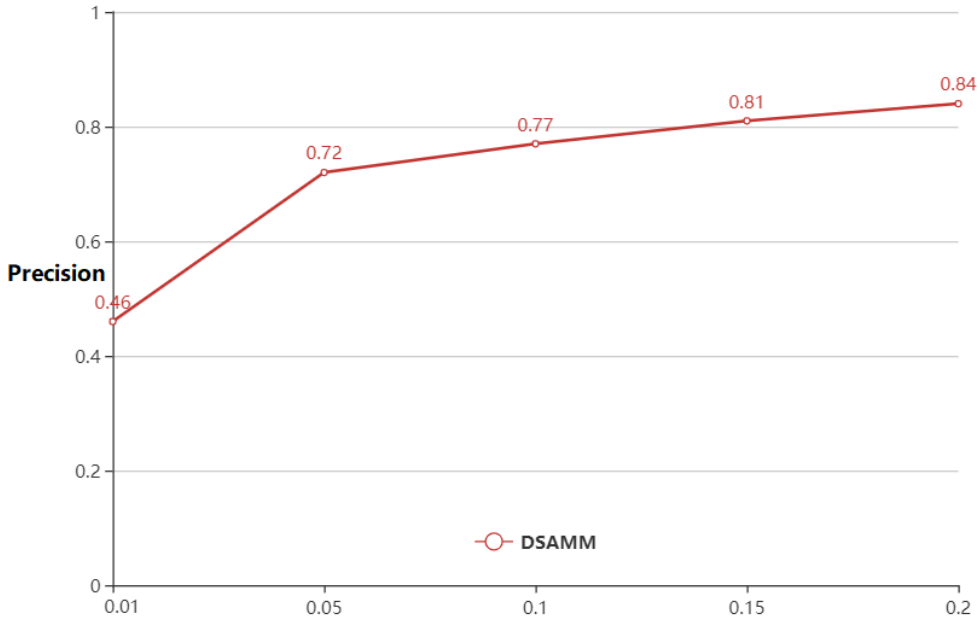

Figure 6. Comparison of the precision of the validation set of the DSAMM instances with different scaled training sets with labels.

3.3. Comparative Experiment Analysis of the Address Matching

To evaluate the semantic address matching performance of the DSAMM we proposed, several baseline models were selected for comparison. The methods compared include character-based matching methods, machine-learning-based matching methods, and deep-learning-based matching methods, where the character-based matching methods in this paper use Levenshtein distance [8], Jaccard similarity coefficient [11], and Jaro similarity [12] to measure string correlation, followed by a random forest (RF) classifier [59] and a support vector machine (SVM) classifier [60] to determine whether the address pairs match. In terms of comparing machine-learning-based matching methods, we compared with the method proposed by Comber et al. [38]. It uses CRF to label the address elements, then uses word2vec for embedding, and applies RF and SVM for classification prediction. In terms of deep-learning-based matching methods, we compared with the method proposed by Lin et al. [21]. It uses a two-stage model, with the first stage using word2vec to obtain embeddings of address pairs for use as input to the next stage of the deep neural network, and the second stage using a typical deep learning model for interaction-based text matching, ESIM [47], which is directly for address matching.

The results of the comparison of the three metrics for each method are shown in Table 6. As the F1 score is the most representative and important evaluation metric, we present the F1 scores for each method in the form of a bar chart for a more visual comparison. As shown in Figure 7, the Jaccard similarity coefficient stands out when using a string-similarity-based approach for the semantic address matching. The classifier employing RF as the task regularly outperforms SVM, and this is also true for the other two groups using machine learning techniques, according to performance comparisons.

When comparing the machine-learning-based matching methods, the CRF method performs worse, probably because the pretraining corpus is more susceptible to influence. In addition, the performance of the machine-learning-based matching methods did not outperform the string-based matching methods, which may be since Comber et al. used relatively generic English addresses, whereas Chinese addresses have their own uniqueness and, therefore, lead to different conclusions [38].

Table 6. Comparative evaluations of different address matching methods in precision, recall, and F1 score.

Comparison Methods	Predictive Results		
	Precision	Recall	F1 Score
Levenshtein distance + SVM	0.849	0.795	0.822
Levenshtein distance + RF	0.875	0.819	0.847
Jaccard similarity coefficient + SVM	0.890	0.881	0.885
Jaccard similarity coefficient + RF	0.914	0.911	0.912
Jaro similarity + SVM	0.902	0.807	0.854
Jaro similarity + RF	0.891	0.858	0.874
word2vec + SVM	0.832	0.762	0.797
word2vec + RF	0.910	0.879	0.895
CRF + word2vec + SVM	0.826	0.751	0.788
CRF + word2vec + RF	0.895	0.846	0.871
word2vec + ESIM	0.969	0.961	0.965
DSAMM	0.987	0.982	0.984

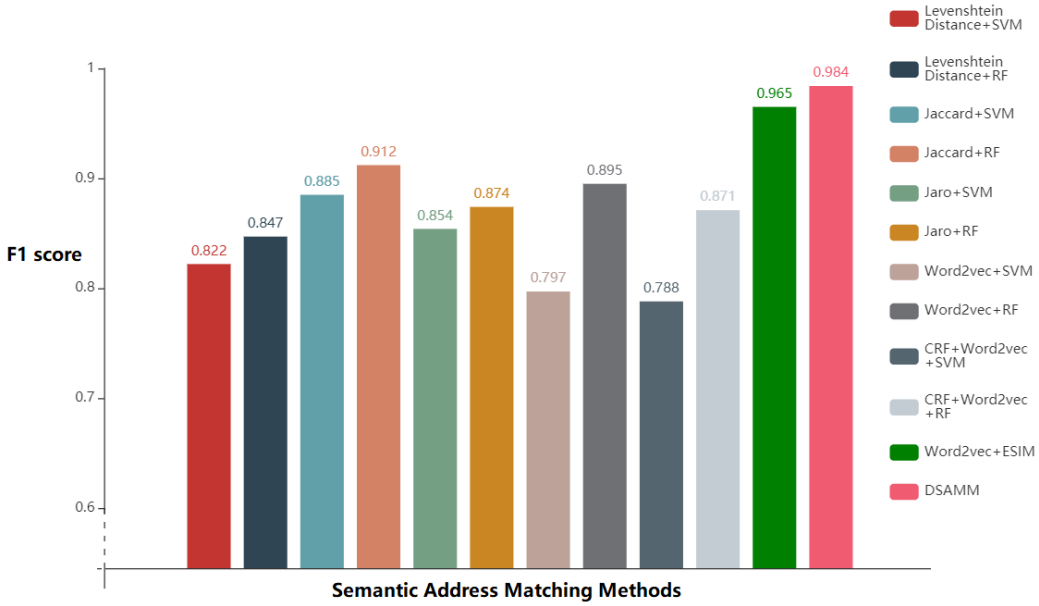

Figure 7. Comparative evaluations of different address matching methods in F1 score.

The deep learning method of "word2vec + ESIM" outperformed the string-similarity-based and machine-learning-based methods in a comparison of results, with all the three evaluation metrics having values above 0.96. This shows how the deep learning framework can significantly increase the accuracy of the semantic address matching. All the final metric values attained by training the DSAMM instances in our study were above 0.98, with the F1 value coming in at 0.984, which represents a notable improvement in prediction evaluation values and the best metrics. This suggests that the migration learning model used in this work can effectively increase the accuracy of semantic address matching. Additionally, we constructed the model without segmenting the address text for elements, which significantly increased the effectiveness of address matching.

4. Conclusions

In this study, we built the ASM with strong address semantic understanding using a pretraining approach to semantic modelling of a vast and complicated address corpus. We introduced a fine-tuning approach in deep transfer learning to achieve a high accuracy of semantic address matching. The main conclusions of this study are as follows:

1. The ASM was constructed using a self-supervised pretrained language model. The experimental results demonstrate that the ASM can achieve a high level of accuracy with the objective of predicting unknown characters.
2. The DSAMM was constructed using the fine-tuning approach in deep transfer learning. The results of the comparison experiments showed that the DSAMM performed the best, with all the metrics above 0.98.
3. It is shown that utilizing deep transfer learning, high address matching accuracy can be attained with only a few labelled training datasets.

However, this study has some limitations. (1) We treated each address record as a sentence, which resulted in the hierarchy of addresses being ignored; (2) our experimental area was limited to a city, which ignored the ambiguity of place names. Therefore, in a future work, we will consider, first, studying how to ensure the accuracy of address matching when there are semantic similarities or ambiguities in addresses at a larger regional scale, and, second, trying to introduce geographical information associated with addresses and learning external knowledge to assist in achieving better address matching.

Author Contributions: Conceptualization, L.X. and F.X.; methodology, L.X., C.Z. and R.M.; validation, X.X. and X.Z.; formal analysis, L.X., F.X. and R.M.; data curation, Y.W.; writing—original draft preparation, L.X.; writing—review and editing, L.X., F.X. and X.Z.; visualization, X.X.; supervision, F.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China Grant Nos. 42001354, 42050103; Natural Science Foundation of Zhejiang Province Grant Nos.LGG22D010001, LQ19D010011, LGN22D010003; Scientific Research Fund of Zhejiang Provincial Education Department Grant No. Y202147221; and Scientific Research and Development Foundation of Zhejiang A&F University Grant No. 2020FR060, 2020FR064.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank the editor and the reviewers for their contributions.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Drummond, W.J. Address matching: GIS technology for mapping human activity patterns. *J. Am. Plan. Assoc.* **1995**, *61*, 240–251. [CrossRef]
2. Longley, P.A.; Goodchild, M.F.; Maguire, D.J.; Rhind, D.W. *Geographic Information Systems and Science*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
3. Holtzheimer, P.E. Introduction to the GBF/DIME: A primer. *Comp. Environ. Urban Syst.* **1983**, *8*, 133–173. [CrossRef]
4. Harada, Y.; Shimada, T. Examining the impact of the precision of address geocoding on estimated density of crime locations. *Comput. GeoSci.* **2006**, *32*, 1096–1107. [CrossRef]
5. Chen, J.; Chen, J.; She, X.; Mao, J.; Chen, G. Deep Contrast Learning Approach for Address Semantic Matching. *Appl. Sci.* **2021**, *11*, 7608. [CrossRef]
6. Lee, B.H.; Waddell, P.; Wang, L.; Pendyala, R.M. Reexamining the influence of work and nonwork accessibility on residential location choices with a microanalytic framework. *Environ. Plan. A* **2010**, *42*, 913–930. [CrossRef]
7. Sun, Z.; Qiu, A.G.; Zhao, J.; Zhang, F.; Zhao, Y.; Wang, L. Technology of fuzzy Chinese-geocoding method. In Proceedings of the 2013 International Conference on Information Science and Cloud Computing, Guangzhou, China, 7–8 December 2013; 2013; pp. 7–12.
8. Levenshtein, V.I. Binary codes capable of correcting deletions, insertions, and reversals. *Sov. Phys. Dokl.* **1966**, *10*, 707–710.
9. Navarro, G. A guided tour to approximate string matching. *ACM Comput. Surv. CSUR* **2001**, *33*, 31–88. [CrossRef]
10. Zhang, Z.; Hadjieleftheriou, M.; Ooi, B.C.; Srivastava, D. Bed-tree: An all-purpose index structure for string similarity search based on edit distance. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, Indianapolis, IN, USA, 6–10 June 2010; pp. 915–926.
11. Jaccard, P. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* **1908**, *44*, 223–270.

12. Jaro, M.A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *J. Am. Stat. Assoc.* **1989**, *84*, 414–420. [CrossRef]
13. Winkler, W.E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In Proceedings of the Section on Survey Research Methods, American Statistical Association, Anaheim, CA, USA, 6–9 August 1990; pp. 354–359.
14. Santos, R.; Murrieta-Flores, P.; Martins, B. Learning to combine multiple string similarity metrics for effective toponym matching. *Int. J. Digit. Earth* **2018**, *11*, 913–938. [CrossRef]
15. Banerjee, S.; Pedersen, T. The design, implementation, and use of the ngram statistics package. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, Mexico, 16–22 February 2003*; Springer: Berlin/Heidelberg, Germany; pp. 370–381.
16. Pedersen, T.; Banerjee, S.; McInnes, B.; Kohli, S.; Joshi, M.; Liu, Y. The Ngram statistics package (text:: nsp): A flexible tool for identifying ngrams, collocations, and word associations. In Proceedings of the workshop on multiword expressions: From parsing and generation to the real world, Oregon, Portland, 23 June 2011; pp. 131–133.
17. Wang, Y.; Liu, J.; Guo, Q.; Luo, A. The standardization method of address information for pois from internet based on positional relation. *Acta Geod. Cartogr. Sin.* **2016**, *45*, 623.
18. Xueying, Z.; Guonian, L.; Boqiu, L.; Wenjun, C. Rule-based approach to semantic resolution of Chinese addresses. *J. Geo-Inf. Sci.* **2010**, *12*, 9–16.
19. Cheng, C.; Yu, B. A rule-based segmenting and matching method for fuzzy Chinese addresses. *Geogr. Geo-Inf. Sci.* **2011**, *3*.
20. 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]
21. Lin, Y.; Kang, M.; Wu, Y.; Du, Q.; Liu, T. A deep learning architecture for semantic address matching. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 559–576. [CrossRef]
22. Wu, D.; Fung, P. Improving Chinese tokenization with linguistic filters on statistical lexical acquisition. In Proceedings of the Fourth Conference on Applied Natural Language Processing, Stuttgart, Germany, 13–15 October 1994; pp. 180–181.
23. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 2001. Available online: https://repository.upenn.edu/cgi/viewcontent.cgi?article=1162&context=cis_papers (accessed on 6 August 2022).
24. Bokaei, M.H.; Sameti, H.; Bahrani, M.; Babaali, B. Segmental HMM-based part-of-speech tagger. In Proceedings of the 2010 International Conference on Audio, Language and Image Processing, Shanghai, China, 23–25 November 2010; pp. 52–56. [CrossRef]
25. Tian, Q.; Ren, F.; Hu, T.; Liu, J.; Li, R.; Du, Q. Using an optimized Chinese address matching method to develop a geocoding service: A case study of Shenzhen, China. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 65. [CrossRef]
26. Koumarelas, I.; Kroschk, A.; Mosley, C.; Naumann, F. Experience: Enhancing address matching with geocoding and similarity measure selection. *J. Data Inf. Qual. JDIQ* **2018**, *10*, 1–16. [CrossRef]
27. 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.
28. Luo, M.; Huang, H. New method of Chinese address standardization based on finite state machine theory. *Comput. Appl. Res.* **2016**, *33*, 3691–3695. (In Chinese)
29. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
30. Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
31. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 6 August 2022).
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
33. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–11.
34. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced language representation with informative entities. *arXiv* **2019**, arXiv:1905.07129.
35. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
36. Xu, L.; Du, Z.; Mao, R.; Zhang, F.; Liu, R. GSAM: A deep neural network model for extracting computational representations of Chinese addresses fused with geospatial feature. *Comput. Environ. Urban Syst.* **2020**, *81*, 101473. [CrossRef]
37. Cruz, P.; Vanneschi, L.; Painho, M.; Rita, P. Automatic Identification of Addresses: A Systematic Literature Review. *ISPRS Int. J. Geo-Inf.* **2021**, *11*, 11. [CrossRef]
38. Comber, S.; Arribas-Bel, D. Machine learning innovations in address matching: A practical comparison of word2vec and CRFs. *Trans. GIS* **2019**, *23*, 334–348. [CrossRef]

39. Zhang, C.; Guo, R.; Ma, X.; Kuai, X.; He, B. W-TextCNN: A TextCNN model with weighted word embeddings for Chinese address pattern classification. *Comput. Environ. Urban Syst.* **2022**, *95*, 101819. [CrossRef]
40. Luo, A.; Liu, J.; Li, P.; Wang, Y.; Xu, S. Chinese address standardisation of POIs based on GRU and spatial correlation and applied in multi-source emergency events fusion. *Int. J. Image Data Fusion* **2021**, *12*, 319–334. [CrossRef]
41. Liu, J.; Wang, J.; Zhang, C.; Yang, X.; Deng, J.; Zhu, R.; Nan, X.; Chen, Q. Chinese Address Similarity Calculation Based on Auto Geological Level Tagging. In Proceedings of the International Symposium on Neural Networks, Moscow, Russia, 10–12 July 2019; Springer: Cham, Switzerland; pp. 431–438.
42. Li, H.; Lu, W.; Xie, P.; Li, L. Neural Chinese address parsing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, MN, USA, 2–7 June 2019; Volume 1, pp. 3421–3431.
43. Li, P.; Luo, A.; Liu, J.; Wang, Y.; Zhu, J.; Deng, Y.; Zhang, J. Bidirectional gated recurrent unit neural network for Chinese address element segmentation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 635. [CrossRef]
44. Santos, R.; Murrieta-Flores, P.; Calado, P.; Martins, B. Toponym matching through deep neural networks. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 324–348. [CrossRef]
45. 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*, 2005–2022. [CrossRef]
46. Li, F.; Lu, Y.; Mao, X.; Duan, J.; Liu, X. Multi-task deep learning model based on hierarchical relations of address elements for semantic address matching. *Neural. Comput. Appl.* **2022**, *34*, 8919–8931. [CrossRef]
47. Chen, Q.; Zhu, X.; Ling, Z.; Wei, S.; Jiang, H.; Inkpen, D. Enhanced LSTM for natural language inference. *arXiv* **2016**, arXiv:1609.06038.
48. Shan, S.; Li, Z.; Qiang, Y.; Liu, A.; Xu, J.; Chen, Z. DeepAM: Deep Semantic Address Representation for Address Matching. In *Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Chengdu, China, 1 August 2019*; Springer: Cham, Switzerland; pp. 45–60.
49. Zhang, H.; Ren, F.; Li, H.; Yang, R.; Zhang, S.; Du, Q. Recognition method of new address elements in Chinese address matching based on deep learning. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 745. [CrossRef]
50. Gupta, V.; Gupta, M.; Garg, J.; Garg, N. Improvement in Semantic Address Matching using Natural Language Processing. In Proceedings of the 2021 2nd International Conference for Emerging Technology (INCET), Belagavi, India, 21–23 May 2021; pp. 1–5.
51. Wei, J.; Zou, K. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv* **2019**, arXiv:1901.11196.
52. Qian, C.; Yi, C.; Cheng, C.; Pu, G.; Liu, J. A coarse-to-fine model for geolocating Chinese addresses. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 698. [CrossRef]
53. Kudo, T.; Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv* **2018**, arXiv:1808.06226.
54. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems 30 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
55. Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q.V.; Salakhutdinov, R. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv* **2019**, arXiv:1901.02860.
56. Dai, A.M.; Le, Q.V. Semi-supervised sequence learning. In Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS 2015), Montreal, QC, Canada, 7–12 December 2015; pp. 3079–3087.
57. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2020**, arXiv:2010.16061.
58. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef]
59. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
60. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol. TIST* **2011**, *2*, 1–27. [CrossRef]