GeoAgent: To Empower LLMs using Geospatial Tools for Address Standardization

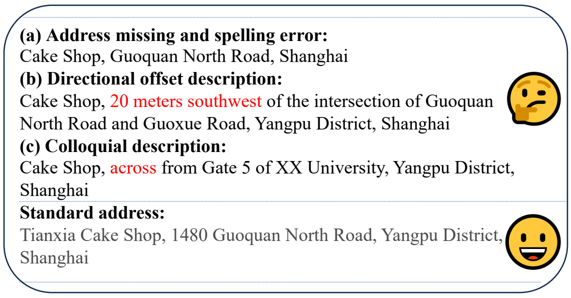
Chenghua Huang[♦], Shisong Chen[♥], Zhixu Li[♦*], Jianfeng Qu[♣*]
Yanghua Xiao[♦], Jiaxin Liu[◇], Zhigang Chen[†]
[♦]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♥]Lab of Artificial Intelligence for Education, East China Normal University
[♣]School of Computer Science and Technology, Soochow University.
[◇]IFLYTEK Research, Suzhou, China. [†]Jilin Kexun Information Technology Co., Ltd.
huangch22@m.fudan.edu.cn, sschen@stu.ecnu.edu.cn, jfqu@suda.edu.cn
{zhixuli,shawyh}@fudan.edu.cn, {jxliu6, zgchen}@iflytek.com

Abstract

This paper presents a novel solution to tackle the challenges that posed by the abundance of non-standard addresses, which input by users in modern applications such as navigation maps, ride-hailing apps, food delivery platforms, and logistics services. These manually entered addresses often contain irregularities, such as missing information, spelling errors, colloquial descriptions, and directional offsets, which hinder address-related tasks like address matching and linking. To tackle these challenges, we propose GeoAgent, a new framework comprising two main components: a large language model (LLM) and a suite of geographical tools. By harnessing the semantic understanding capabilities of the LLM and integrating specific geospatial tools, GeoAgent incorporates spatial knowledge into address texts and achieves efficient address standardization. Further, to verify the effectiveness and practicality of our approach, we construct a comprehensive dataset of complex non-standard addresses, which fills the gaps in existing datasets and proves invaluable for training and evaluating the performance of address standardization models in this community. Experimental results demonstrate the efficacy of GeoAgent, showcasing substantial improvements in the performance of address-related models across various downstream tasks.[1]

1 Introduction

With the widespread using of navigation maps (e.g. Google Maps), ride-hailing apps (e.g. Uber), food delivery platforms (e.g. Uber Eats), and logistics services (e.g. Amazon Logistics) in our daily life, a significant amount of user-entered addresses have been collected. However, these addresses often suffer from irregularities, such as missing addresses and spelling errors (Figure 1(a)), directional offset descriptions (Figure 1(b), colloquial descriptions (Figure 1(c)) etc. Handling these non-standard addresses presents considerable challenges for subsequent downstream tasks such as address matching (Lin et al., 2019) and address linking (Huang et al., 2021).



(a) Address missing and spelling error:
Cake Shop, Guoquan North Road, Shanghai
(b) Directional offset description:
Cake Shop, 20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai 🤔
(c) Colloquial description:
Cake Shop, across from Gate 5 of XX University, Yangpu District, Shanghai
Standard address:
Tianxia Cake Shop, 1480 Guoquan North Road, Yangpu District, Shanghai 😀

Figure 1: Examples of three non-standard address types and their corresponding standard addresses. (a) Many address elements are missing. Description information in (b) and (c) is highlighted.

To address this issue, the task of **Address standardization** (Lu et al., 2019) is proposed to correct, complete, and normalize input address data, converting non-standard addresses into standard ones. Actually, there are various possible ways to express an address. For example, in Figure 1, phrase a, b and c represent the same geographical location but are totally different in surface expression. This requires the model to possess relevant spatial knowledge and strong semantic understanding ability, so as to normalize these addresses to the standard one.

Conventional methods for address standardization often rely on character matching and rules, such as splitting address hierarchies based on address tree (Mengjun et al., 2015) and then completing missing administrative region elements based on a hierarchical address database (Tian et al., 2016). Unfortunately, these methods face limitations in handling fine-grained non-standard addresses, and shallow rules may not cover all sce-

[*] Corresponding authors.
[1] Resources of this paper can be found at https://github.com/chenghuahuang/GeoAgent

Findings of the Association for Computational Linguistics: ACL 2024, pages 6048–6063
August 11-16, 2024 ©2024 Association for Computational Linguistics

narios. In recent years, with the advancements in deep learning technology, researchers have explored the use of neural network models to tackle non-standard address issues (Ye et al., 2022). Recently, many methods of pre-training in geographic text corpus have been proposed (Huang et al., 2022; Ding et al., 2023; Gao et al., 2022), or design models for specific tasks (Wu et al., 2023; Halterman, 2023; Brunila et al., 2023; Tao et al., 2022). However, such methods require a large amount of annotated data and are unable to handle common descriptive information (e.g., 20 meters southwest) in addresses.

With the emergence of large language models (LLMs) like ChatGPT, they have demonstrated powerful semantic understanding capabilities, leading to remarkable performance across multiple tasks. Existing work shows that LLMs have some coarse-grained geographic knowledge, but it still suffer from the following problems according to our observations: 1) **Spatial Similarity Issues**: LLMs excel in processing natural language texts and understanding their semantics. However, address information processing involves spatial similarity issues. Geographically close addresses may have significant textual differences, such as "No. 323 Songhan Road, Baoshan District" and "No. 861 Guofan Road, Yangpu District". Although the two addresses are literally completely different, they are actually very close, located on opposite sides of an intersection. To tackle such cases, the model needs spatial knowledge to discern spatial correlations. Unfortunately, existing LLMs often lack geographic and address-related fine-grained knowledge, resulting in unsatisfactory results when dealing with addressing-related problems. 2) **Changes in Geographic Spatial Knowledge**: Geographic information is subject to frequent changes, with businesses, companies, and establishments experiencing relocations, closures, and other transformations over time. Large language models store knowledge in parameters, making it difficult for them to adapt to such knowledge changes. It requires substantial training costs to update these parameters, which makes it challengeable to keep the model up-to-date with the real-world changes in address information 3) **Deficiencies in Precise Numerical Calculations**: Existing work indicates that LLMs have deficiencies in performing precise numerical calculations (Schick et al., 2023). This limitation makes it challengeable for them to handle non-standard addresses that involve directional shifts (e.g. figure 1(b)).

To tackle the above problems, inspired by the latest research on LLM Agent (Yang et al., 2023; Wang et al., 2023b), we propose GeoAgent, an innovative framework that combines a LLM with geospatial tools. By leveraging the language understanding and decision-making capabilities of LLMs to interact with geospatial tools, GeoAgent effectively processes non-standard addresses. By introducing an address knowledge base, fine-grained geospatial knowledge can be obtained. To maintain geospatial knowledge efficiently, we store it in an external vector database. This approach significantly reduces costs compared to re-training the model every time when the knowledge needs updating. Furthermore, our method uses spatial computing tools to achieve accurate spatial offset calculations.

Recognizing the lack of a dataset that contains various non-standard addresses, especially those with descriptive information, we construct a comprehensive dataset for providing non-standard addresses for this task. Our dataset covers complex non-standard address linking, address standardization, and geocoding, which effectively fills the gaps in existing datasets. We have conducted extensive experiments on both our dataset and existing datasets, demonstrating that GeoAgent's standardized processing significantly enhances the performance of address-related models in downstream tasks. To account for the importance of text position in address text evaluation, we propose a new standardized metric called GeoRouge, which provides a more comprehensive and relevant evaluation for address standardized.

Our major contributions are highlighted as follows:

- We propose the GeoAgent framework, a novel approach that combines the capabilities of LLM with geospatial tools to address issues related to address text.
- Our work includes the construction of a comprehensive dataset comprising various non-standard addresses, aiming at closely resembling real-life scenarios of address text problems. Additionally, we introduce a new metric, GeoRouge, specifically tailored for measuring address standardization performance.
- Extensive validation of our framework's effectiveness has been conducted through a series

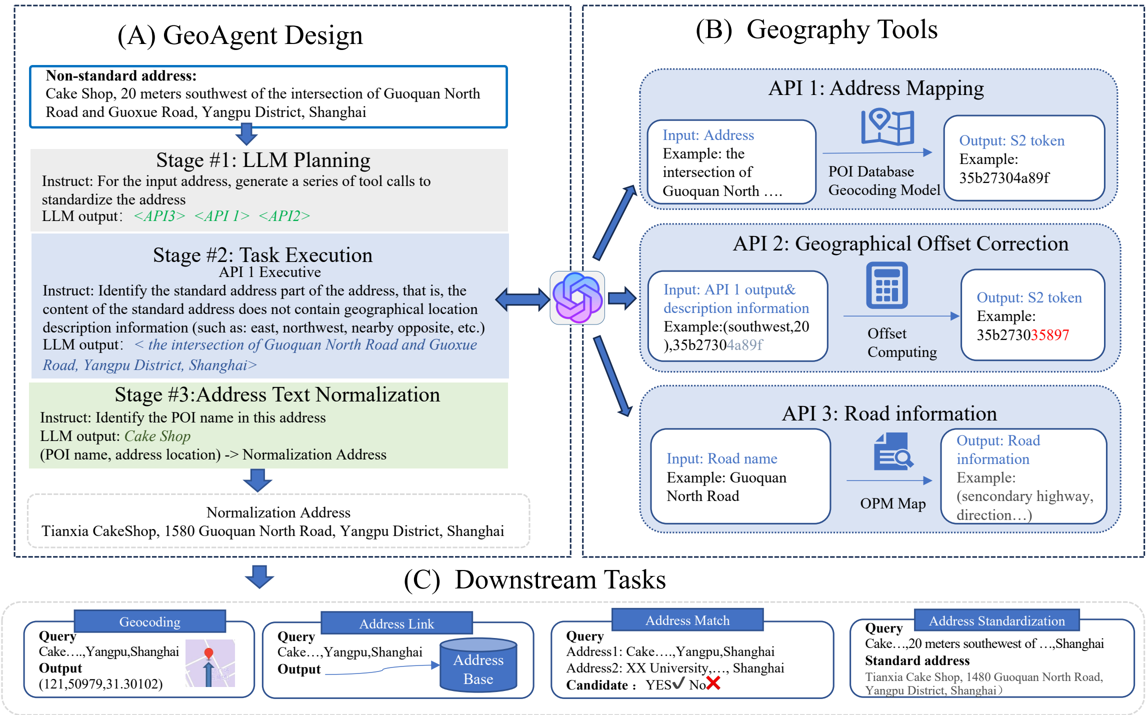

(A) GeoAgent Design
Non-standard address:
Cake Shop, 20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai
Stage #1: LLM Planning
Instruct: For the input address, generate a series of tool calls to standardize the address
LLM output: <API3> <API 1> <API2>
Stage #2: Task Execution
API 1 Executive
Instruct: Identify the standard address part of the address, that is, the content of the standard address does not contain geographical location description information (such as: east, northwest, nearby opposite, etc.)
LLM output: < the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai>
Stage #3:Address Text Normalization
Instruct: Identify the POI name in this address
LLM output: Cake Shop
(POI name, address location) -> Normalization Address
Normalization Address
Tianxia CakeShop, 1580 Guoquan North Road, Yangpu District, Shanghai
(B) Geography Tools
API 1: Address Mapping
Input: Address
Example: the intersection of Guoquan North ….
POI Database
Geocoding Model
Output: S2 token
Example: 35b27304a89f
API 2: Geographical Offset Correction
Input: API 1 output& description information
Example:(southwest,20),35b27304a89f
Offset Computing
Output: S2 token
Example: 35b273035897
API 3: Road information
Input: Road name
Example: Guoquan North Road
OPM Map
Output: Road information
Example: (sencondary highway, direction…)
(C) Downstream Tasks
Geocoding
Query
Cake….,Yangpu,Shanghai
Output
(121,50979,31.30102)
Address Link
Query
Cake…,Yangpu,Shanghai
Output
Address Base
Address Match
Query
Address1: Cake….,Yangpu,Shanghai
Address2: XX University,…, Shanghai
Candidate : YES✓ No✗
Address Standardization
Query
Cake…,20 meters southewest of …,Shanghai
Standard address
Tianxia Cake Shop, 1480 Guoquan North Road, Yangpu District, Shanghai）

Figure 2: An illustration of GeoAgent. (A) is a demonstration of the GeoAgent workflow. (B) depicts the series of geography tools and the corresponding inputs and outputs. (C) are some of the downstream tasks related to address text.

of experiments. The results demonstrate significant improvements in various tasks when using the normalized data.

2 Related Work

Address Standardization. Mainstream address standardization methods can be divided into two categories: standard address matching based, address element split based. Standard address based method requires to establish a reference standard address database, followed by fuzzy matching according to some rules (Buckles et al., 1994). The disadvantage is that it requires a high-quality and large-scale standard address database, and the algorithm bottleneck is quite obvious. Address element split based method split address by administrative region element and complete missing elements (Tian et al., 2016). The address split methods include rule split,statistical model split methods (Mengjun et al., 2015) and deep learning methods (Li et al., 2018; Matcı and Avdan, 2018; Zhao et al., 2019; Yassine et al., 2020). Due to the diversity and arbitrariness of address colloquial expression, these methods were limited generalization ability for address combinations that are not present in the training data and unable to handle common descriptive information in address. To address such problems, in this paper, we propose GeoAgent which combines LLM and geospatial tools. LLM's powerful semantic understanding ability can handle various address expression, and convert the description information in the address into location movement problem is calculated by geospatial tools.

Geography Pre-trained Model. Pre-trained language models have achieved excellent results in many task. Some scholars improve the address related tasks performance by injecting geographic knowledge into pre-trained models (Gao et al., 2022; Huang et al., 2022; Ding et al., 2023; Deng et al., 2023; Li et al., 2023b; Chen et al., 2022). However, these works focused on specific address tasks rather than improving the quality of address data, which is the focus of this work. Recently, some works attempt to stimulate geographic knowledge within the LLM (Roberts et al., 2023; Manvi et al., 2023; Gurnee and Tegmark, 2023), but these work show that there is only contain coarse-grained geographic knowledge in the model. In this paper, we introduce fine-grained knowledge into the LLM by interact with knowledge base and geographic tools.

Specific Task-Soving with LLM Agent. Specific task-soving LLM agents rely on LLMs for proficiency in task decomposition (Wei et al., 2022b), generalization in decision-making, language understanding capabilites (Wei et al., 2022a), interacts with the environment solves tasks that the LLM itself cannot solve well (Wang et al., 2023a; Shen et al., 2023). These agents have diverse applications in robotics, law and complex reasoning (Dalvi et al., 2022; Cui et al., 2023; Pan et al., 2023). In this paper, we interact with the environment through the LLM, introduce fine-grained geographical knowledge, accurate spatial computation, complete the task of address standardization.

3 GeoAgent

Before diving into technical details, we formally define the Address Standardization task below:

Address Standardization. We define the address standardization task as the problem of going from a non-standard address S to a standard address S^*. The standard address refers to the address expression that conforms to the address writing rules. Taking the Chinese scenario as an example, a standard city address expression should be "province - city - district - street - street number - POI name" (Although our proposed approach is language-independent, we provide an analysis of different language implementations details in the appendix E).

The core architecture of our framework is shown in Figure 2, which consists of three main steps: (1) Task planning. Given an address S, the model generates a series of task execution sequences. Splitting the address standardization task into smaller tasks. (2) Task execution. Execute and return results based on the sequence of tasks are generated by the LLM. (3) Address text standardization. According to the results of the previous task, the adress text processing process is executed to obtain the standardized address S^*. The resulting standardized address is the output of the GeoAgent.

3.1 Task Planning

The aim of GeoAgent is to enable the LLM to process non-standard addresses in a way that aligns with human thinking and intuition. For a non-standard address that contains descriptive information, human intuition dictates that we first find the location of the standard part address section, which is called "Address Mapping", and then use the description to determine the final location, which is called "Offset Correction".

In this step, as shown in Figure 2, due to the variety of description styles of non-standard addresses, the LLM should generate different tool calls based on the input address. For example, for the address "Cake Shop, 20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District". LLM plan first calls the API 3 to query the relevant road information, then calls the API 1 to find the real location, and finally calls the API 2 directional offset tool based on the type of description information.

3.2 Task Execution

The task execution step executes all the tasks generated by the Task Planning step. LLM extracts parameters from the input address to pass the tool, we provide prompt API 1 in Figure 2, other API parameters and prompt can be seen in table 6. Here we need to build a toolset to facilitate LLM in executing these tasks, including address mapping tool, offset calculation tool and road information tool.

3.2.1 Address Mapping Tool

The function of this tool is to map the address text to the corresponding real geographical location. For example, for an address "the intersection of Guoquan North Road and Guoxue Road, Yangpu District", the output of the tool is "35b27304a89f", we use Google S2[2] tokens to represent real geographic locations rather than latitude and longitude. This tool mainly consists of two components:

Standard POI Database. We construct a standardized POI (Point of Interest) database containing Shanghai address, which comprises approximately 1.4 million POI address data. We organize the database in the format of "standard address -S2token" pairs.

Geocoding Model. To improve the generalization of the address mapping tool, we train a Geocoding model to deal with the address inputs that do not exist in the standard POI database. We model the mapping between address and location in the real world as a seq-to-seq problem in this paper, aiming to capture potential correlations between tokens.

Specifically, we choose transformers (Vaswani et al., 2017) as the model architecture. By encoding the input address text and decoding it into S2 tokens, the model can learn the semantics and correlations between addresses and spatial locations.

[2] https://s2geometry.io

The model is trained on the Geocoding dataset mentioned in the experiment. The implementation details for the model can be seen in the Appendix A.2.

Address text differs from the normal text in that the tokens at the beginning of the sequence are more important and represent a larger range (*Chinese address conventions, in English may be reversed*). When prediction errors occur at these tokens, the resulting deviation from the true position is greater. Taking into consideration the characteristics of address text, we use GeoEntropy as loss function, it is defined as follows:

$$L_{Geo} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{i=1}^{l} w_s \cdot L_i \tag{1}$$

In equation(1), N represents the number of samples, and l represents the sequence length. w_s is a one-dimensional weight matrix with dimensions equal to the sequence length. We assign higher weights to tokens located closer to the beginning, indicating their higher importance. L_i represents the loss function for the i-th token, computed according to the formula shown below:

$$L_i = \sum_{c=1}^{T} w_t \cdot \left(y_{\text{true},i,c} \cdot \log(y_{\text{pred},i,c})\right) \tag{2}$$

In equation(2), T represents the number of categories for S2 tokens. Each layer of S2 encoding has 16 possible values, including 6 English letters (a to f) and 10 Arabic numerals (0 to 9). Within the same layer, the distances between different tokens vary. To account for these differences in token distances, we design a weight matrix W with dimensions of $T * T$. The weight assigned to tokens increases as the distance between them becomes larger. We have verified the effectiveness of our loss function design, see detail in Appendix B.

3.2.2 Offset Calculation Tool

The function of this tool is to perform spatial calculations based on the description information. The input of this tool is the descriptive information element extracted by the LLM from the non-standard address, and the results of the address mapping tools. The output of the tool is the position after displacement according to the description information. This tool mainly consists of two components:

Directional Descriptions Offset. Directional offset description is a common way to express addresses, with the format of a direction followed by a distance, such as “200 meters southeast”. We utilize spatial calculation tools[3] to perform spatial calculations based on the initial location obtained in address mapping tool, along with the displacement direction and distance.

Colloquial descriptions offset. The colloquial description is a vague description of the location, such as “nearby”, “next to”, “opposite”, and so on. We need to perform corresponding actions based on the specific type of colloquial description provided through LLM. Take “opposite” as an example. When the parameter “opposite” is received, the tool queries the result of the road information tool. The tool determines the displacement direction according to the direction of the road and displacement distance according to the width of the road, then performs spatial calculation to obtain the final geographic location of the input address. When the argument is “next to” or “near”, we will take the results of the address mapping tool as the output of this tool, because these verbal expressions are very close and the error is within our acceptable range.

3.2.3 Road Information Tool

The purpose of this tool is to obtain more accurate displacement direction and displacement distance. we observe that directional displacement descriptions are often based on roads, such as “50 meters northwest of the intersection of XX Road and XX Road”, or they involve two addresses that are directly connected by a road. Therefore, if there are roads within 50 meters of the starting point that have an angle deviation from the displacement direction within ±22.5 degrees, the direction of those roads is considered as the displacement direction. we introduce road network information from OPM (OpenStreetMap [4]) to improve the accuracy of the displacement direction.

The input is the name of the road, and the output is the information of the road, including the direction of the road and the level information of the road (such as the main road, the rural slip road, etc., the road width can be inferred by the road level). This tool provides a more accurate displacement direction and enhances the precision of location displacement.

3.3 Address Text Standardization

In this step, we use geographic location information and the original non-standard address text to

[3] https://github.com/shapely/shapely

[4] https://www.openstreetmap.org

standardize the input address. The LLM recognizes the POI name from the non-standard address and then performs a query in POI database based on the geospatial location and POI name. If the query is successful, the corresponding address in the standardized POI database is taken as the output of this step. If the query fails, we integrate the address information from the original address text.

POI Database Linking Based on the original input address's POI name identified by LLM, along with the geographic location information, we make another attempt to match it with the POI database based on a comprehensive evaluation of both POI name similarity and S2 token similarity. If a successful match is found, the standardized address from the standard POI database is returned as the standardization result.

Address information integration Based on the geographic location information, we utilize the open-source OPM library for reverse geocoding to perform error correction and completion on the original input address. This process involves the following steps:

Administrative area standardization: The address location information in the open-source OPM library is used to correct and complete the administrative area at all levels in the original address text.

Road name standardization: For original input texts that lack road information or contain errors with multiple road names, we standardize the road by selecting the closest road based on the road network data. We also clean up additional descriptive information, such as directions. By incorporating the recognized original POI name, we obtain the final standardized result.

3.4 Instruction Tuning

In order to enable LLM to call tools in the way we want (such as calling APIs with <> symbols and passing parameters with [] symbols), we manually built some dialog instructions containing tool calls based on the non-standard addresses mentioned above, and then extended them with ChatGPT. For the model finetuning and tool call dialogues, see the Appendix A.3.

4 Experiments

In this section, we present the results of the GeoAgent standardized address on four address-related tasks (address matching, geocoding, address standardization, and address linking) compared to the original non-standard address and the construction process of our dataset.

Task	Train	Dev	Test
Geocoding	7540K	76K	76k
Address Linking	175k	4K	4k
Address standardization	175k	4k	4k

Table 1: Statistics of our dataset

4.1 Dataset Construction

Considering that there is no relevant dataset for non-standard addresses, especially the problem of containing descriptive information, we construct a non-standard dataset containing descriptive information based on some heuristic rules and a standard POI database. The dataset contains 3 address-related downstream tasks (geocoding, address linking, address standardization), and the dataset size is shown in the table 1. We construct non-standard addresses that conform to human writing habits by the following method:

Address Missing: For a standard address, we adopt the following strategies with 15% probability: delete administrative area entities, delete characters, and replace characters. This is in line with the handwritten address in the administrative area entity and characters missing, spelling errors, etc.

Address Descriptive Information: We design algorithm 1 to add description information to an address based on the direction and distance between addresses in the POI database. The details of the algorithm can be viewed in the Appendix D.1.

ChatGPT expanding: Our approach to data construction is based on heuristic rules, but rules obviously can't cover everyone's language habits. Since ChatGPT has demonstrated excellent in-context learning capabilities, we use ChatGPT to enrich the expression of descriptive information in addresses, our prompt can be viewed in the Appendix D.2.

The statistics of our dataset as shown in Table 1, through the above methods, we construct about 7700K non-standard addresses. 183K of this data is allocated to address standardization and address linking tasks, and the rest is used for geocoding tasks. An example of the dataset is shown in Appendix D.3.

4.2 Evaluation Metrics and Baseline

Address Standardization. we choose the paid service provided by the logistics service provider with the largest number of domestic users as the baseline, we use AS-1[5] to refer to it. The evaluation metric we choose is the edit distance, which measures the average edit distance between the non-standard address and the corresponding standard address in the POI database.

In address standardization tasks, the importance of n-grams located at the beginning of the text is obviously greater than those located at the end (in Chinese address text, the administrative regions are arranged from large to small, from the front to the back, which is opposite to English addresses). For example, for the address “No. 2005 Guoquan North Road, Yangpu District, Shanghai”, administrative areas “Yangpu” and “Shanghai”, the traditional gram-based metrics (Rouge (Lin, 2004), BLEU (Papineni et al., 2002)) are of the same importance in the calculation. However, from the point of view of address accuracy, the importance of “Shanghai” is greater than “Yangpu”, and if the prediction is wrong, the actual location error is greater.

Therefore, we propose a new metric for the address standardization task, called “GeoROUGE-N”, inspired by ROUGE metrics used in text generation tasks. The GeoROUGE-N metric assigns higher weights to n-grams with higher importance. The GeoROUGE-N formula is as follows:

$$GeoROUGE-N = LP \cdot \frac{\sum\limits_{gram_n \in S} w_i \cdot v_{match}[i]}{\sum\limits_{gram_n \in S} w_i} \tag{3}$$

v_{match} is a one-dimensional vector with a dimension equal to the number of n-grams. For the i-th gram, if it is successfully matched in the candidate, $v_{match}[i]$ is 1, otherwise it is 0. w_i is the weight based on the position of the gram in the text, with higher weights assigned to grams located closer to the beginning of the text. The specific formula is shown below:

$$w_i = (1 - \frac{i}{n}) * (1 - \lambda) + \lambda \tag{4}$$

$$\text{LP} = \begin{cases} 1, & \text{if } c < r \\ \exp^{(1-c/r)} & \text{if } c \geq r \end{cases} \tag{5}$$

Here, i represents the index of the gram in the sequence, n represents the length of the gram sequence, and λ represents the minimum threshold for the weight. LP is the length penalty coefficient, which is used to penalize excessively long texts. c is the length of the candidate text and r is the length of the reference text.

Geocoding. The Geocoding task takes as input an address text and outputs the real geographic location coordinates corresponding to the address. The Geocoding task is a basic service for many mapping applications. Therefore, we choose the paid geocoding service provided by the two map applications with the largest number of users in China to illustrate the effectiveness of our method. we use GC-1[6] and GC-2[7] to refer to them, without specifying their names to avoid commercial competition. The evaluation metrics follow the ERNIE-Geo setting, using “Accuracy@N km” as the evaluation metric. It represents the percentage of samples in which the predicted distance from the true distance is less than N km. In our experiments, we set N to 0.1 and 1. In addition, we also provide a more granular evaluation metric, “Average distance”, measured in meters. It represents the average distance between the predicted distance and the true distance.

Address Matching. The goal of the address matching task is to determine whether two different address texts refer to the same geographic entity. The task takes two address texts as input and outputs a label indicating whether they match, with the type “exact match”, “partial match”, or “not match”. We conducted the experiment at the Shanghai address in the GeoGlue (Li et al., 2023a).

The baseline models we choose for comparison are three strong generic PTMs, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019),StructBert (Wang et al., 2019), MGeo (Ding et al., 2023), Among them, MGeo performs pre-training on geographic corpus and sets a unique pre-training task to learn location information in the map. The models are all of the base model size and are trained for 3 epochs on the training set. For the PTM+GeoAgent experiments, the model is also trained and tested on the training and validation data after being normalized by GeoAgent. We use Precision, Recall, and Macro F1 as evaluation metrics, and the results are shown in Table 3.

Address Linking. The task of Address Linking is to link an input address to a standard address in a standard POI database. Our approach to this

[5] https://www.aliyun.com/product/addresspurification/addrp

[6] https://lbs.amap.com/api/webservice/guide/api/georegeo

[7] https://api.map.baidu.com/geocoding

Method	AED	GR-1	GR-2	GR-3	GR-4
Non-standard	12.12	0.739	0.677	0.645	0.618
AS-1	15.27	0.738	0.608	0.519	0.438
GeoAgent	**3.742**	**0.926**	**0.892**	**0.871**	**0.852**

Table 2: Address standardization results. AED indicates the average edit distance. GR-N is GeoRouge-N grams.

Method	Precision	Recall	F1
Bert	**0.814**	0.699	0.738
+*GeoAgent*	0.769	**0.733**	**0.749**
RoBERTa	0.739	0.669	0.691
+*GeoAgent*	**0.809**	**0.760**	**0.774**
StructBert	0.727	0.656	0.679
+*GeoAgent*	**0.834**	**0.769**	**0.792**
Mgeo	**0.809**	0.741	0.762
+*GeoAgent*	0.761	**0.767**	**0.763**

Table 3: The results of different models on the address matching. +*GeoAgent* indicates the results of the model on address that have been standardized by GeoAgent.

Method	MRR@5
Mgeo	0.698
Mgeo+GeoAgent	**0.744**

Table 4: Address linking results

Method	Ac@100M	Ac@1KM	AD(m)
GC-1	0.579	0.920	535.65
+*GeoAgent*	**0.590**	**0.938**	**469.10**
GC-2	**0.626**	0.904	396.46
+*GeoAgent*	0.617	**0.942**	**354.05**
GeoAgent	**0.866**	**0.977**	**53.05**

Table 5: The performance of the different geocoding apis, +*GeoAgent* indicates the performance of the api on addresses that have been standardized by GeoAgent. AD indicates the average error distance.

task is a combination of pre-ranking and ranking. We first filter the candidate addresses based on the input address's district, and then the model scores the candidate addresses, recalling the top 5 candidates with the highest scores. Our evaluation metric is MRR@5, which is a measure commonly used for evaluating linking algorithms. we choose the MGeo as the baseline.

4.3 Overall results

Address standardization results. Our results are shown in Table 2. Non-standard indicates the difference between the original non-standard address and the corresponding standard address. It can be observed that GeoAgent significantly reduces the edit distance (-8.378). GeoAgent also achieved very high scores in the GeoRouge-N metric, with no significant decay in grams 1, 2, 3, and 4. This shows that GeoAgent is able to take into account the importance of the order of address elements. The results show that it is necessary to process the description information in non-standard address in the process of address standardization. We provide a case study of the task in the Appendix C.1.

Geocoding results. The Table 5 shows the results of geocoding for non-standard address and address standardized by GeoAgent, the average distance error is reduced by 66.55M and 42.41M, respectively. This proves the necessity and validity of standardizing non-standardized addresses. In addition, the average error of GeoAgent is 53.06m, and the Ac@100M is 86.6%. This shows that GeoAgent's method of splitting non-standardized addresses into standard address mapping and description information offset steps is effective. We provide a case study of the task in the Appendix C.2.

Address matching results. GeoAgent's standardization of non-standard addresses improves the performance of all models, with the largest improvements seen in RoBERTa (+8.3 %) and StructBert (+11.3 %). The results of combining GeoAgent with StructBert surpass the SOTA model MGeo, demonstrating the effectiveness of GeoAgent. To further analyze the reasons, we analyze the prediction changes for each class and found that GeoAgent increases the similarity between part match addresses and reduces the similarity between not match addresses through the standardization process, greatly increasing the model's classification accuracy for these two classes. Details will be analyzed in the Appendix C.3.

Address linking results. The results of address linking are shown in Table 4, which demonstrates that GeoAgent helps improve the performance of SOTA models in their domain of expertise (from 69.8% to 74.4%).

5 Conclusion

In conclusion, this paper introduces GeoAgent, a novel solution designed to address the challenges posed by non-standard addresses frequently input by users in modern applications. We utilize LLM task planning ability and tool utilization ability to make up for LLM lack of fine-grained spatial knowledge and accurate spatial calculation problems, leading to efficient address standardization. Additionally, we create an extensive dataset of complex non-standard addresses, which bridges the gaps in existing datasets. The experimental results demonstrate the effectiveness of GeoAgent, as it showcases significant enhancements in the performance of address-related models across various downstream tasks.

Limitations

First of all, our experiment was conducted on Chinese address. Although the framework of our proposed method is language-independent, it may require some minor adjustments to generalize our method to other languages due to the differing conventions of addressing (GeoLoss, GeoRouge).

Secondly, limited by our resources, we chose ChatGLM-6B as the LLM backbone, which has limited ability to follow and understand instructions. In order to make the model output in the format we want, we conducted fine-tuning. As the size of the LLM increases and its ability to understand and follow instructions improves, the model should be able to output in the desired format via in-context learning, thereby skipping the fine-tuning step.

Finally, because our method consists of multiple steps, cascading errors can occur. For example, if the geocoding has an average error of 50 meters, and the address happens to be at the junction of two administrative regions, this could lead to an administrative region prediction error, although such instances are relatively rare.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work is supported by the National Natural Science Foundation of China (No.62072323, U21A20488, No.62102276), Shanghai Science and Technology Innovation Action Plan (No.22511104700), China Postdoctoral Science Foundation (Grant No. 2023M732563), Zhejiang Lab Open ResearchProject (No.K2022NB0AB04).

References

Mikael Brunila, Jack LaViolette, Sky CH-Wang, Priyanka Verma, Clara F'er'e, and Grant McKenzie. 2023. Toward a critical toponymy framework for named entity recognition: A case study of airbnb in new york city. In *Conference on Empirical Methods in Natural Language Processing*.

Bill P. Buckles, James J. Buckley, and Fernando Petry. 1994. Architecture of fame: fuzzy address matching environment. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*, pages 308–312 vol.1.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *North American Chapter of the Association for Computational Linguistics*.

Jiaxi Cui, Zongjia Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *ArXiv*, abs/2306.16092.

Bhavana Dalvi, Oyvind Tafjord, and Peter Clark. 2022. Towards teachable reasoning systems: Using a dynamic memory of user feedback for continual system improvement. *ArXiv*, abs/2204.13074.

Cheng Deng, Tianhang Zhang, Zhongmou He, Qiyuan Chen, Yuanyuan Shi, Le Zhou, Luoyi Fu, Weinan Zhang, Xinbing Wang, Cheng Zhou, Zhouhan Lin, and Junxian He. 2023. K2: A foundation language model for geoscience knowledge understanding and utilization.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

Ruixue Ding, Boli Chen, Pengjun Xie, Fei Huang, Xin Li, Qiang-Wei Zhang, and Yao Xu. 2023. Mgeo: Multi-modal geographic language model pre-training. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2021. Glm: General language model pretraining with autoregressive blank infilling. In *Annual Meeting of the Association for Computational Linguistics*.

Yunfan Gao, Yun Xiong, Siqi Wang, and Haofen Wang. 2022. Geobert: Pre-training geospatial representation learning on point-of-interest. *Applied Sciences*.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time. *ArXiv*, abs/2310.02207.

Andrew Halterman. 2023. Mordecai 3: A neural geoparser and event geocoder. *ArXiv*, abs/2303.13675.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Jizhou Huang, Haifeng Wang, Yibo Sun, Miao Fan, Zhengjie Huang, Chunyuan Yuan, and Yawen Li. 2021. Hgamn: Heterogeneous graph attention matching network for multilingual poi retrieval at baidu maps. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.

Jizhou Huang, Haifeng Wang, Yibo Sun, Yunsheng Shi, Zhengjie Huang, An Zhuo, and Shikun Feng. 2022. Ernie-geol: A geography-and-language pre-trained model and its applications in baidu maps. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Dongyang Li, Ruixue Ding, Qiang-Wei Zhang, Zheng Li, Boli Chen, Pengjun Xie, Yao Xu, Xin Li, Ning Guo, Fei Huang, and Xiaofeng He. 2023a. Geoglue: A geographic language understanding evaluation benchmark. *ArXiv*, abs/2305.06545.

Lin Li, Wei Wang, Biao He, and Yu Zhang. 2018. A hybrid method for chinese address segmentation. *International Journal of Geographical Information Science*, 32:30 – 48.

Zekun Li, Wenxuan Zhou, Yao-Yi Chiang, and Muhao Chen. 2023b. Geolm: Empowering language models for geospatially grounded language understanding. *ArXiv*, abs/2310.14478.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the Association for Computational Linguistics*.

Yue Lin, Mengjun Kang, Yuyang Wu, Qingyun Du, and Tao Liu. 2019. A deep learning architecture for semantic address matching. *International Journal of Geographical Information Science*, 34:559 – 576.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yan Lu, Hongli Liu, and Yanquan Zhou. 2019. Chinese address standardization based on seq2seq model. *Proceedings of the 2019 2nd International Conference on Computational Intelligence and Intelligent Systems*.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David B. Lobell, and Stefano Ermon. 2023. Geollm: Extracting geospatial knowledge from large language models. *ArXiv*, abs/2310.06213.

Dilek Küçük Matcı and Uğur Avdan. 2018. Address standardization using the natural language process for improving geocoding results. *Comput. Environ. Urban Syst.*, 70:1–8.

Kang Mengjun, Du Qingyun, and Wang Mingjun. 2015. A new method of chinese address extraction based on address tree model.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *ArXiv*, abs/2305.12295.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*.

Jonathan Roberts, Timo Luddecke, Sowmen Das, K. Han, and Samuel Albanie. 2023. Gpt4geo: How a language model sees the world's geography. *ArXiv*, abs/2306.00020.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *ArXiv*, abs/2302.04761.

Yongliang Shen, Kaitao Song, Xu Tan, Dong Sheng Li, Weiming Lu, and Yue Ting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *ArXiv*, abs/2303.17580.

Liufeng Tao, Zhong Xie, Dexing Xu, Kai Ma, Qinjun Qiu, Shengyong Pan, and Bo Huang. 2022. Geographic named entity recognition by employing natural language processing and an improved bert model. *ISPRS Int. J. Geo Inf.*, 11:598.

Qin Tian, Fu Ren, Tao Hu, Jiangtao Liu, Ruichang Li, and Qingyun Du. 2016. Using an optimized chinese address matching method to develop a geocoding service: A case study of shenzhen, china. *ISPRS Int. J. Geo Inf.*, 5:65.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Neural Information Processing Systems*.

Wei Wang, Bin Bi, Ming Yan, Chen Wu, Zuyi Bao, Liwei Peng, and Luo Si. 2019. Structbert: Incorporating language structures into pre-training for deep language understanding. *ArXiv*, abs/1908.04577.

Zekun Wang, Ge Zhang, Kexin Yang, Ning Shi, Wangchunshu Zhou, Shaochun Hao, Guangzheng Xiong, Yizhi Li, Mong Yuan Sim, Xiuying Chen, Qingqing Zhu, Zhenzhu Yang, Adam Nik, Qi Liu, Chenghua Lin, Shizhuo Wang, Ruibo Liu, Wenhu Chen, Ke Xu, Dayiheng Liu, Yi-Ting Guo, and Jie Fu. 2023a. Interactive natural language processing. *ArXiv*, abs/2305.13246.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2023b. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. *ArXiv*, abs/2307.05300.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed Huai hsin Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Lixia Wu, Jianlin Liu, Junhong Lou, Haoyuan Hu, Jianbin Zheng, Haomin Wen, Chao Song, and Shu He. 2023. G2ptl: A pre-trained model for delivery address and its applications in logistics system. *ArXiv*, abs/2304.01559.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Anima Anandkumar. 2023. Leandojo: Theorem proving with retrieval-augmented language models. *ArXiv*, abs/2306.15626.

Marouane Yassine, David Beauchemin, François Laviolette, and Luc Lamontagne. 2020. Leveraging subword embeddings for multinational address parsing. *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 353–360.

Zi Ye, Xuefeng Piao, Fanchao Meng, Bo Cao, and Dianhui Chu. 2022. A non-standardized chinese express delivery address identification model based on enhanced representation. *2022 International Conference on Service Science (ICSS)*, pages 5–11.

Ji Zhao, Dan Peng, Chuhan Wu, H. Chen, Meiyu Yu, Wanji Zheng, Li Ma, Hua Chai, Jieping Ye, and Xiaohu Qie. 2019. Incorporating semantic similarity with geographic correlation for query-poi relevance learning. In *AAAI Conference on Artificial Intelligence*.

A Implementation Details

A.1 LLM backbone

Considering the need for good Chinese understanding ability, we choose ChatGLM-6B (Du et al., 2021) as the base LLM and fine-tuned it on Lora (Hu et al., 2021). We use the AdamW (Loshchilov and Hutter, 2017) optimizer with a learning rate of 3e-5 and a weight decay of 0.001. The data is shuffled, and the training set and test set are divided. The model is trained for 3 epochs with a batch size of 8 on 4 GTX3090s. The max input/output token length of the model is set to 200. For the language model's generation function, the following hyper-parameters are used: max length: 200, temperature: 0, do sample: False. All other parameters follow the default settings. The parameters of Lora are as follows: rank: 8, Lora alpha: 32, Lora dropout: 0.1, Lora layer: 0-27.

A.2 Geocoding model

We choose the transformer as the network structure for the address mapping model. The number of attention heads is 8, and the dimension of the hidden state is 512. The training batch size is 1024, and we use AdamW as the optimizer with a learning rate of 1e-4 and weight decay of 0.02. We train the model for 50 epochs on 4 GTX3090 GPUs and select the epoch with the best performance on the validation set as the final checkpoint.

We have design a loss function for the address mapping model, Which includes two loss weights: sequence loss weight W_s and token loss weight W_t. The sequence loss weight is an 12-dimensional vector, we set it to [1,1,1,1,1.35,1.30,1.25,1.20,1.15,1.10,1.05,1]. The token loss weight W_t is a 16*16 matrix, please visit our GitHub repository to learn more about the specific parameters.

A.3 LLM Instruction Sample

We followed the example in the Table 6, built 2K data, and trained 3 epoch of Chatglm using Lora. The fine-tuning parameters are shown in Appendix A.1. The aim of finetuning is to make the output format of the model conform the predefined format (such as wrapping street names with <[]> symbols, shown in Table 6), rather than to inject fine-grained geographical knowledge. In larger models with greater language understanding(such as ChatGPT), the fine-tuning process might be replaced by in-context learning prompt.

B Loss Function Ablation Study

We investigate the impact of a loss function designed for the address mapping task on the performance of address mapping, as shown in Table 7. We choose the baseline model trained with the original CrossEntropy Loss as the loss function, while keeping the remaining parameters unchanged. It can be observed that the GeoEntropy, designed specifically for the characteristics of address mapping, can effectively improve the model's prediction performance compared to the original CrossEntropy Loss. The average distance between the predicted S2token and the true geographic location was reduced from 236.34 meters to 180.89 meters

Table 6: A summary of API call examples

APIs	API Call Templates	Prompt Examples		
		Instructions	Inputs	Outputs
API 1	<at[]>	"Use the <at[]>tag to mark the standard address portion in this address:"	Cake Shop, across from Gate 5 of XX University , Yangpu District, Shanghai	Cake Shop, across from <at[]>Gate 5 of XX University , Yangpu District, Shanghai
API 2	<go[direction, distance] >	"Use the <go[]>tag to mark the geographic location descriptive information in this address:"	Cake Shop, 20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai	Cake Shop, <go[southwest,20]>20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai
	<go[colloquial]>		Cake Shop, across from Gate 5 of XX University , Yangpu District, Shanghai	Cake Shop, <go[across from]>across from Gate 5 of XX University , Yangpu District, Shanghai
API 3	<rb[road name]>	"Use the <rb[]>tag to mark the road in this address:"	Cake Shop, 20 meters southwest of the intersection of Guoquan North Road and Guoxue Road, Yangpu District, Shanghai	Cake Shop, 20 meters southwest of the intersection of Guoquan North Road <rb[Guoquan North Road]>and Guoxue Road <rb[Guoxue Road]>, Yangpu District, Shanghai

Loss	Ac@100M	Ac@1KM	AD(m)
CrossEntropy	0.655	0.958	236.34
GeoEntropy	**0.743**	**0.968**	**180.89**

Table 7: Ablation results in loss function

(-55.45 meters), and the proportion of predicted distances within 100 meters of the true distance increased from 65.5% to 74.3% (+8.8%). These results demonstrate that weighting the output sequence positions and types helps the model capture the correlation between address text and real locations.

C Case study

C.1 Address standardization case

As shown in Table 2 of our paper. It can be observed that GeoAgent significantly reduces the edit distance (8.378), while the API service even shows an increase in edit distance relative to the original non-standard address. As the specific algorithm of the API-provided address standardization service is not transparent, we can only analyze it based on the API data. It may be due to the fact that the service relies on POI address database characters for error correction and completion, which cannot handle addresses containing descriptive information well.

The no-standard address "No. 356, Aomen Road, Sanxiang Building, 246 meters south of Shibachuan (Jiangning Road Branch), Putuo District, Shanghai" was standardized by AS-1 as: "Shibachuan (Jiangning Road Branch), No. 356 Aomen Road, Changshou Road Subdistrict, Putuo District, Shanghai". However, the correct address is "Shibachuan (Jiangning Road Branch), No. 1228 Jiangning Road, Putuo District, Shanghai (160 meters walk from Exit 4 of Jiangning Road Subway Station)". For addresses that do not contain descriptive information, such as address element missing :"B2 D-065, Paris Spring (Pu Jian Branch), No. 118 Jian, Pudong District", AS-1 output: "D-065, B2, Paris Spring (Pu Jian Branch), No. 118 Pu Jian

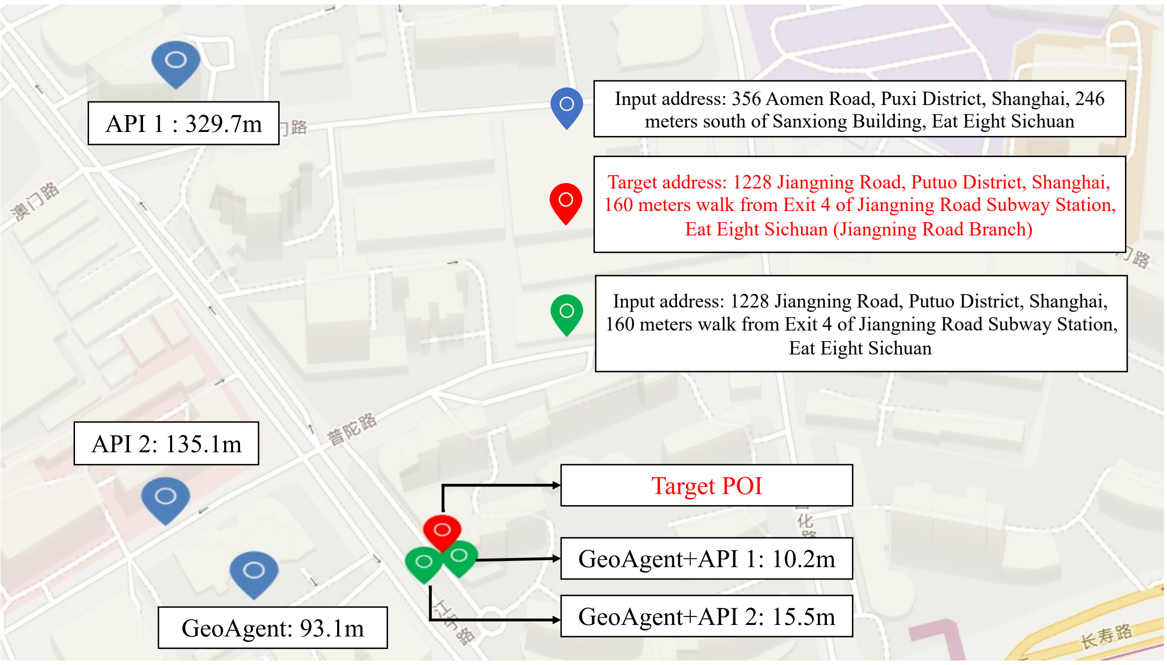

API 1 : 329.7m
Input address: 356 Aomen Road, Puxi District, Shanghai, 246 meters south of Sanxiong Building, Eat Eight Sichuan
Target address: 1228 Jiangning Road, Putuo District, Shanghai, 160 meters walk from Exit 4 of Jiangning Road Subway Station, Eat Eight Sichuan (Jiangning Road Branch)
Input address: 1228 Jiangning Road, Putuo District, Shanghai, 160 meters walk from Exit 4 of Jiangning Road Subway Station, Eat Eight Sichuan
澳门路
API 2: 135.1m
普陀路
Target POI
GeoAgent+API 1: 10.2m
GeoAgent+API 2: 15.5m
GeoAgent: 93.1m
长寿路

Figure 3: Geocoding case

Road, Tangqiao Street, Pudong New Area, Shanghai", is very close to the standard address.It can be seen that AS-1 only removes the descriptive information by extracting features, without properly understanding and processing the descriptive information. This also shows that in the task of address standardization, it is very necessary to process the description information in the address

C.2 Geocoding case

As described in section 5.2, the Geocoding task takes address text as input and aims to parse the input address into its corresponding real-world geographic location. Figure 3 shows a typical example from the experiment. The blue markers represent the Geocoding results for non-standardized addresses, the green markers represent the Geocoding results standardized by GeoAgent, and the red markers represent the true geographic location of the address.

For non-standardized address input text (text marked in blue), both API 1 and API 2 have significant errors (329.7 meters and 135.1 meters, respectively). GeoAgent achieves the best performance(93.1 meters) by splitting non-standardized addresses into two stages: standard address mapping and location offset.

After the input address is standardized by GeoAgent, it was successfully linked to the standard address library, and the results are very impressive when the standardized address is used as the input for API calls. This indicates that map service providers, based on their massive data resources, can map standard addresses to very accurate positions. However, when an address contains descriptive information, it can greatly increase the error, which also demonstrates the importance of standardizing non-standardized addresses.

C.3 Address matching case

To investigate the reasons why standardized address text can improve the performance of address matching tasks, we take StructBert as an example and analyze the model's prediction performance before and after address standardization on each class in the GeoGLUE dataset, as shown in Table 10.

Our analysis reveals that GeoAgent increases the text similarity between addresses with exact matches and reduces the text similarity between addresses with non-matches through the standardization process. This greatly improves the model's classification performance for these two classes, with F1 scores increasing by 8% and 17%, respectively. By standardizing missing and incorrect address elements, addresses with partial matches have the same prefix, making it easier to discern the labels for partial matches, resulting in a 12% improvement in performance.

For example, for the original data labeled as non-matching "address1: No. 206, Lane 999, Pinglu Road, Shanghai; address2: Building 206, Wenxiang Mingyuan, No. 3088, Wenxiang Road, Songjiang District, Shanghai", after standardization, it becomes "address1: No. 206, Lane 999, Pinglu Road, Jing'an District, Shanghai; address2: Building 206, Wenxiang Mingyuan, No. 3088, Wenxiang Road, Songjiang District, Shanghai". This increases the difference between non-matching addresses, which helps to improve the performance of this type.

For two models (Bert, MGeo) that have made only modest progress, we analyze the bad cases to find out why. We found an overall trend, the accuracy and recall rate of the model are improved for the two categories of partial matching and non-matching, but the precision of the model is decreased for the category of full matching.The reason is that after GeoAgent address standardization, the text similarity between the two addresses is improved, so the model may mistakenly predict that the partially matched data is a full match. But StructBert and Roberta have stronger semantic understanding ability and can effectively distinguish it. The detailed results are shown in the table 8 and table 9.

For example, address 1: "Qianming Cun She, Qingping Road, Qingpu District, Shanghai", address 2: "Qianming Cun Bridge G50, Qingpu District", after GeoAgent standardization, address 2 becomes: "Qianming Cun Bridge, G50, Xujing Town, Qingpu District, Shanghai". The similarity between the two addresses becomes higher at the text level. For Bert, by encoding the text as a vector and then calculating the Cos similarity of the two vectors, may not be able to distinguish this small difference, so the two addresses are incorrectly predicted to full match, The actual results is not match. For MGeo, its text-only training method is very similar to Bert, which trains the model through MLM pre-training task. Although the MGM (Masked Geographic Modeling) pre-training task is added to learn location information, this module is more relevant to situations with geo-location information (such as latitude and longitude) input.So it makes the same mistake as the Bert.

D Dataset Construction Details

D.1 Descriptive Information Construction

As shown in Algorithm 1, the algorithm takes as input the standard address u that has undergone the address entity missing process. The algorithm searches for a POI v within a distance of less than 500 meters from u. If the distance between u and v satisfies the condition in line 3 of the algorithm, we use spatial calculation tools to obtain the relative direction and distance between u and v as the descriptive information. Similarly, if u and v satisfy the condition in line 7 of the algorithm, "nearby" is returned as the descriptive information. If the distance between u and v satisfies the condition in line 10 of the algorithm, "next to" is returned as the descriptive information. Following the formula in line 13 of the algorithm, we combine the address u after the address entity missing process with the descriptive information and the POI name of v to obtain the final non-standard address with descriptive information.

For simplicity, we use "u.ad" to denote the missing address of u, "desc" to denote the descriptive information, and "v.name" to denote the name of the POI v. The final non-standard address with descriptive information is obtained by combining the address u after the address entity missing process with the descriptive information and the POI name of v.

Class	Precision	Recall	F1
Full Match	1(-0.33)	0.583(+0.083)	0.736(+0.07)
Part Match	0.615(+0.079)	0.66(+0.034)	0.64(+0.054)
Not Match	0.82(+0.066)	0.848(+0.038)	0.837(+0.049)

Table 8: The performance of Bert in the three types (full match, partial match, and no match), with the result of GeoAgent processing in parentheses

Class	Precision	Recall	F1
Full Match	0.818(-0.126)	0.75(+0)	0.782(-0.06)
Part Match	0.782(-0.055)	0.5(+0.16)	0.61(+0.08)
Not Match	0.827(+0.037)	0.974(-0.088)	0.895(-0.02)

Table 9: The performance of MGeo in the three types (full match, partial match, and no match), with the result of GeoAgent processing in parentheses

D.2 ChatGPT Expanding Prompt

we use ChatGPT (gpt-3.5-turbo) in this paper to enrich the expression of descriptive information in addresses. To do so, we adopt the In-Context Learning (ICL) approach and use prompts to provide input instructions and three examples, as shown in Table 11.

	StructBert			StructBert+GeoAgent		
Class	Precision	Recall	F1	Precision	Recall	F1
Exact match	0.875	0.583	0.70	0.818	0.750	0.782
Paritial match	0.50	0.638	0.560	0.840	0.583	0.688
Not match	0.808	0.746	0.776	0.846	0.974	0.905

Table 10: The performance of StructBert in the three types

Algorithm 1: Non-standard Address Construction

```
Input: Standard addresses in the address database
Output: Non-standard addresses containing descriptive information
for each address u in the address database do
    for each address v in the address database and u != v do
        if 20 <distance(u, v) <500 then
            calculate the direction m and distance n of u relative to v;
            desc = m + n;
        end
        else if 10 <distance(u, v) <20 then
            desc = “nearby”;
        end
        else if 5 <distance(u, v) <10 then
            desc = “next to”;
        end
        Non-standard address = u.ad + desc + v.name;
        add the non-standard address to the non-standard address list;
    end
end
```

D.3 Dataset sample

An example of the dataset we built is shown in Table 12, where we provide examples of inputs and outputs for each task. Note that in the table, the input address is in English, but it is actually in Chinese in the dataset.

For an geocoding task, the input is a non-standard address and the output is the actual spatial location of that address (S2token). For an address Standardization task, the output is the correct address for that address. For the address linking task, the output is the index of the corresponding standard address in the standard address library. In the address linking task, the model will sort all the addresses in the standard address library based on their degree of association with the input address, and we will determine the performance of the task based on whether the top 5 addresses with the highest degree of association contain the correct answer.

E Generalized Discussion

Due to the complexity of Chinese address (there is no symbol to separate the address, and the expression forms are diverse), compared with other languages, Chinese address is very challenging, so we choose Chinese address as the research object.

However, our framework(GeoAgent) are language independent.Specifically, we model the address standardization task in two steps: first, get the real world location of the address part in the address text, and second, offset the location according to the description information. This framework is not tied to the language itself and can be generalized to use in different languages. To implement the above steps, it is up to the LLM to determine the use and order of the geographic tools based on the different input addresses.

Next, we discuss how our work can be applied to languages with different styles of expression - using English as an example. The expression of address in English is different from that in Chinese. It is customary to arrange the address elements according to the size of administrative area from small to large, which is the opposite of Chinese. So to use our method in English addresses, you only need to make the following changes. LLM: Choose a LLM with English ability. Modify the GeoLoss: If you want to train a model that maps addresses to specific locations yourself, you just need to modify the weights of the GeoLoss to give higher scores to the parts that represent larger administrative areas. Modify GeoRouge: If you need to use GeoRouge for addresses with administrative areas arranged from smallest to largest, simply change the formula $1 - i/n$ in W_i in formula (4) to i/n.

Prompt	
Given an address, we aim to rephrase the descriptive information related to the location. Descriptive information refers to additional details about the location, such as “234 meters to the south”, “nearby”, “opposite”, “next to”, etc.	
Demonstrations	
Input: Red Swallow Food Business located 409 meters northwest of Cheng Ye Bath, No. 266 Xinhua West Road, Zhangyan Town, Jinshan District, Shanghai.	Output: Red Swallow Food Business is located 409 meters northwest of Cheng Ye Shower Room, No. 266 Xinhua West Road, Zhangyan Town, Jinshan District, Shanghai.
Input: Beside Industrial and Commercial Bank of China, 20 Fengbin Road, Chongming District, is Yixuan Cultural Communication.	Yixuan Cultural Communication is located near Industrial and Commercial Bank of China, 20 Fengbin Road, Chongming District.
No. 8-2, Zhongyi Residence, Gaoxi Village, is located about 150 meters southwest of the intersection of Ting’an Road and Pudong North Road, Pudong New District, Shanghai.	8-2, Zhongyi Residence, Gaoxi Village, is located about 150 meters southwest of the intersection of Ting’an Road and Pudong North Road, Pudong New District, Shanghai.
Demonstrations End	

Table 11: The details of the prompt design for ChatGPT expanding

Task	**Input**	**Output**
Geocoding	Opposite Huicai Restaurant, No. 652, Libao Road, Ma Town, Jiading District, Shanghai	35b26bc225f5
Address Standardization	Opposite Huicai Restaurant, No. 652, Libao Road, Ma Town, Jiading District, Shanghai	Huiwei Huicai Restaurant ,No. 2350 Baoan Road, Jiading District, Shanghai
Address Linking	Opposite Huicai Restaurant, No. 652, Libao Road, Ma Town, Jiading District, Shanghai	index in POI database

Table 12: Dataset sample