Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 2, 2023, 291-302.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Persian Address Geocoding: an LALR Parsing and Dynamic Programming Approach

Alireza Mazochi[1], Sara Bourbour[2], Mohammadreza Ghofrani[1] and Saeedeh Momtazi[1*]

1. Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran.
2. Tehran Center for Urban Statistics and Observatory, Tehran, Iran.

Article Info

Article History:
Received 11 December 2022
Revised 11 January 2023
Accepted 18 March 2023

DOI:10.22044/jadm.2023.12493.2399

Keywords:
Geocoding, Persian, Address parsing, Postal address, Natural language processing, Viterbi.

**Corresponding author: momtazi@aut.ac.ir (S. Momtazi)*

Abstract

Converting a postal address to a coordinate, geocoding, is a helpful tool in many applications. Developing a geocoder tool is a difficult task if this tool relates to a developing country that does not follow a standard addressing format. The lack of complete reference data and non-persistency of names are the main challenges besides the common natural language process challenges. In this paper, we propose a geocoder for Persian addresses. To the best of our knowledge, our system, TehranGeocode, is the first geocoder for this language. Considering the non-standard structure of Persian addresses, we need to split the address into small segments, find each segment in the reference dataset, and connect them to find the target of the address. We develop our system based on address parsing and dynamic programming for this aim. We specify the contribution of our work compared to similar studies. We discuss the main components of the program, its data, and its results, and show that the proposed framework achieves promising results in the field by finding 83% of addresses with less than 300 meters error.

1. Introduction

Addressing has a vital application in city services and many businesses. Delivery services use postal addresses to specify the source where a packet was sent and the destination where it should be delivered; routing services use addresses to find their path from source to destination.

Converting a postal address to a coordinate is called geocoding. As a formal definition, given a textual address as input, the geocoding task aims to find the corresponding geographic coordinates (latitude and longitude) as output, representing the location described by the address. Geocoding is challenging in countries that do not follow standard addressing. These challenges include the lack of standard naming conventions for roads, non-persistency of names of roads, absence of a complete and uniform system for data of cities, and different formats to specify an address from one person to another [1]. In some cases, finding the intention coordinate of an address is a complex task, even for a human. Khan, Pinault, Tjepkema, and Wilkins [2] showed the impact of postal codes in address geocoding. This information, however, is not included in addressing structure in many countries.

Our target language is Persian. Persian is one of the Indo-European languages that has borrowed its script from Arabic, a member of Semitic language family [3]. Persian, also known as Farsi, is spoken primarily in Iran, Afghanistan, and Tajikistan. Persian addresses follow a different style from standard addressing formats in other languages, such as English. A Persian address usually begins with a wide area. Each segment makes the remaining area more minor and specific than the previous one. Finally, the last segment specifies the target point. Not only is the overall style different, but there is also no standard way to present an address. In other words, to present a Persian address, we need to imagine a route from a famous location or a large city area to reach the target location. Therefore, observation of different addresses for a location is entirely possible. As an example of a Persian address, the Amirkabir University of Technology addresses can be stated as "Tehran, Valiasr Square, Karimkhan Str, Hafez Avenue, No. 350"[1] or "College Crossroads, under Hafez Bridge, No. 350".

Moreover, Tehran, the capital of Iran, and other cities of Iran do not follow a standard pattern for the name of the roads. Thus we cannot find locational information from the semantics of words in an address. Finally, Persian is a non-Latin language with Arabic characters that require its pre-processing tools, which are less accurate than the English pre-processing tools.

This research proposes a comprehensive architecture for the Persian addresses geocoding. The proposed system receives a postal address in Tehran and returns the coordinate of the given postal address. This program uses the API of the Tehran map from the Tehran Municipality ICT Organization[2]. Although this API is limited to the map of Tehran, our processing units are general and can be used for any Persian address for Iranian cities.

The main issue about the Map API is that it only accepts a short single segment of an address; e.g., the name of a road, like "Azadi Street" or the name of an important place like "Milad Tower". Therefore, sending a long address with different segments including the name of a square, a street, an alley, and the apartment number results in no answer. Moreover, when sending a short address segment to the API, the output of the Map API is a list of points rather than a single point, which requires human action to select a point among this list; i.e., even if we have a minimal address that consists of only one segment that is related to a famous location, we cannot automatically geocode this address since the system does not know which of these points is the coordinate of the input address. In fact, the Map API is not a geocoder at all. It works like a database that matches a single segment input address to a list of coordinates. This shortcoming motivated us to propose a pipeline for parsing addresses and segmenting the text. The extracted segments can then be sent to the API. We also propose a post-processing component that receives different geocoding and combines them to achieve the final result for each address using a dynamic programming approach. The proposed framework allows finding a coordinate for addresses that include various parts and even contain over ten words.

In the literature, several methods are used for geocoding, which can be categorized into two main branches: rule-based [1, 4-8] and learning-based approaches [9-14]. In the developing countries with limited sources, rule-based methods are more applicable. In contrast, in the developed countries with well-structured addresses, learning-based methods are preferable.

Our contribution is developing a Persian geocoder system. To the best of our knowledge, our system, TehranGeocode, is the first geocoder for the Persian language. For processing a non-standard Persian address, we develop our system with a pipeline structure. We design an LALR parser with handcrafted diverse and relatively robust rules. The rules are extracted from comprehensive investigations of the existing Persian non-standard addresses. This dedicated parser splits an address into segments. Then another module collects data for each segment with an existing Map API. In addition, we propose a novel dynamic programming module to relate the collected data and find the final location related to an address. Our results reveal the effectiveness of TehranGeocode for real applications.

The structure of the paper is as follows: Section 2 reviews related works in address geocoding for other languages and other countries. Section 3 introduces our proposed framework and its main components. Section 4 describes the datasets used for developing the system, the reference data, and different Persian addresses. Section 5 reports the quantitative and qualitative results, and finally, Section 6 summarizes the paper.

2. Related Works

Geoparsing is converting a text to a coordinate of location that the text references. This process breaks into two sub-processes: "geotagging" and "geocoding". Geotagging aims to find the locations that exist in a text but geocoding focuses on converting a non-structural and erroneous location to a coordinate. Some research studies presented a tool for the whole process of geoparsing [9, 13] but some other studies only focused on one part of geocoding such as pre-processing [8]. Note that our program is a complete end-to-end geocoding tool but it does not have components for geotagging; i.e., the system assumes that the input text is an address only. As mentioned in Section 1, geocoding has various applications. Wheeler, Gerell, and Yoo [15] evaluated the accuracy of geocoding for estimating the location of shooting incidents.

[1] To make the text readable for all readers, we translated the Persian texts to English.

[2] https://tmicto.tehran.ir/

Wilson and Wilson [16] proposed an iterative procedure for health data to compare the housing condition and the health situation of child occupants

Considering different applications in different countries and different languages, the researchers tried to create a geocoder for their countries or cities. Kebe *et al*. [1] implemented a geocoder for Senegal as a developing country with English and French languages. Alcántara *et al*. [9] developed a tool for Spanish address disambiguation. Matci and Avdan [8] and kilic and Gülgen [17] studied geocoding for addresses in Turkey. Cetl, Kliment, and Jogun [18] compared different geocoding methods for the city of Zagreb, Croatia. Charif *et al*. [4] studied the task for Luxembourg. L. Li, Wang, He, and Zhang [19] proposed a model for segmenting Chinese addresses. Dumedah *et al*. [20] selected Ghana as the research goal location. Cortes *et al*. [5] studied enhancing geocoding for the Portuguese language in Rio de Janeiro, the capital of Brazil. Malaainine and Lechgar [7] developed a geocoder for the city of Casablanca in Morocco. Nguyen, Tsolak, Karmann, Knauff, and Kühne [21] chose Germany as the target location.

Various methodologies have been used for geoparsing in related research in the field. For research studies related to developing countries, such as Turkey and Senegal, rule-based techniques are more favorites [1, 4-8]. LOGEMAS is a geocoder that has several components. Each component has specific rules and processes without the machine learning concept [6]. In developing countries, sometimes, even the reference data does not become available. For developing a geocoder for Gana, Dumedah *et al*. [20] used polling location data as the reference data. The lack of a good reference dataset is the main bottleneck of the mentioned studies. We can see good and novel heuristic ideas for overcoming this challenge but reaching the best answer is still an open challenge.

In contrast to rule-based technique, learning-based techniques are used in other studies [9-14]. Nizzoli *et al*. [13] proposed a geoparser based on Geo Semantic Parsing (GSP). This approach uses a knowledge graph and traverses it to collect more data. The last phase in GSP is selecting a coordinate. The idea of the coordinate selection phase of GSP is to define it as a regression task. This work also has a geotagging part, making GSP a complete geoparsing system. Although GSP can enrich the text with knowledge graphs, it is worth mentioning that a knowledge graph commonly includes main places and cities rather than all streets of a city. In another research study, Alcántara *et al*. [9] proposed a model that first converts the text to a vector via word2vec [22] and doc2vec [23], and then trains a model with these vectors and finds the similarity between texts for geoparsing disambiguation. This method focuses on the disambiguation of geoparsing, and does not find coordinates as its target task. But applying semantic features can boost the accuracy of any geoparsing system. Yan *et al*. [14] developed the model LGGeoCoder. This model uses global context embedding beside local context embedding. The topic embedding for linguistic features and the location embedding for geospatial embedding are two new global features in their model. By adding new diverse features, LGGeoCoder can overcome its baselines. P. Li *et al*. [12] proposed a model based on a gated recurrent unit network for segmenting input Chinese address text and finding the main elements. Segmentation of Chinese addresses is a challenging and vital part of Chinese geocoding. In this article, the relationship between address segments and any semantic information is ignored. Kilic and Gülgen [10] used logistic regression and analyzed the quality of reverse geocoding. They tested fifteen similarity metrics and found the best for four existing geocoding systems. Rashidian, Dong, Avadhani, Poddar, and Wang [24], Rashidian, Dong, Jain, and Wang [25] proposed a tool called EaserGeocoder, which provides scalable address geocoding with distributed computing and parallel searching. Presenting an available and open-source geocoding system is the strength of this work.

Some research in the field is dedicated to enhancing geocoding with techniques like additional text processing tasks. Matci and Avdan [6] showed that text pre-processing could improve geocoding accuracy. Clemens [26] enhanced the text processing module by using spelling variations captured from user queries. Using query logs as another source of data is a proper idea. Lee *et al*. [11] enhanced geocoding by adopting matching learning techniques for address matching and showed that extreme gradient boosting achieves the best result. Applying machine learning techniques can remove human errors and improve the overall output, but this method only works for particular addresses (street-based addresses). Nguyen *et al*. [21] designed a geocoder for addresses on Twitter in Germany. This geocoder is based on the open-source geocoder Nominatim, and they customized it for their location and domain.

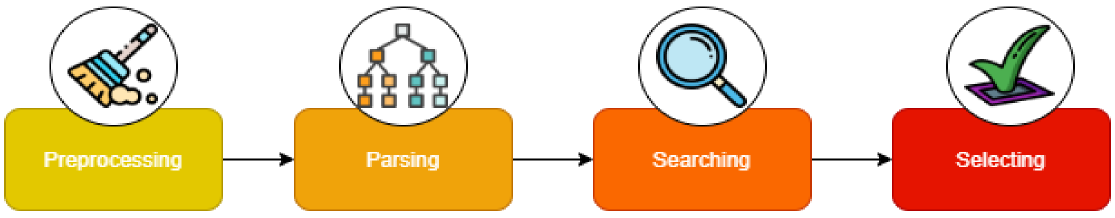

Figure 1. Main components of the program in a pipeline structure.

To the best of our knowledge, our proposed system is the first Persian geocoder. Some research studies [27-30] acknowledged the lack of a good Persian geocoding tool.
As mentioned, since the addressing format in Persian is non-standard and mainly follows a routing scenario from important locations to specific ones; the available methods for other languages do not work on this addressing model. Therefore, we propose our framework, which aims to find the location of each address by finding the location of segments and connecting them using dynamic programming. Dynamic programming is an efficient and speedy programming technique used in many applications; Still, some works are present for improving it [31].

3. Proposed Framework

The proposed framework includes four components, each comprising different modules dedicated to a specific task. Figure 1 presents our model's architecture, a pipeline architecture, and the components serially process input to generate the final output. In the following sections, we describe each component in detail.

3.1. Pre-processing

Pre-processing is the first component of the pipeline aiming to normalize the given address to ease other components' tasks. The component is responsible for removing unnecessary characters and unifying characters with different writing styles.
For removing unnecessary characters, the following tasks are considered:

- Short Arabic vowels
- Arabic sukun character
- Unnecessary characters, such as $ and !
- Some abbreviations
- Unbalanced parentheses by using stack

For unifying characters with different writing styles, the normalization of the following characters is taken into consideration:

- Characters that have different notations in different languages such as comma and the letters Kaf and Ya, which have different notations in Arabic.
- Numbers that have different representations in Arabic and English characters.

3.2. Address segmentation

This component's input is an address that has been pre-processed but is still in unstructured text format. This component aims to specify the address's elements such as names of streets, squares, and alleys. We perform the task by converting the address format from text to a tree structure.
The main challenge is that in the Persian language, there is no standard and constant method for naming roads and even writing an address; e.g., an address may have no street name but another address may include five streets. The main reason for this problem is that we do not have any standard way of expressing an address in Persian, and we have different streets with the same name, which should be distinguished by expressing the neighboring parts.
To overcome this problem, we introduce a context-free grammar to analyze Persian addresses. The grammar can also be adapted to other languages with some changes based on the differences between addressing formats in the languages.
Every address has several main parts. This article uses the term "Segment" for each part. For example, the address "Shadmehr Street, Teimoori Boulevard, Gole Sorkh Alley" has these segments: "Gole Sorkh Alley", "Teimoori Boulevard", "Shadmehr Street". Note that each segment can have sub-segments, which our context-free grammar should recognize to convert raw text address to tree format address.

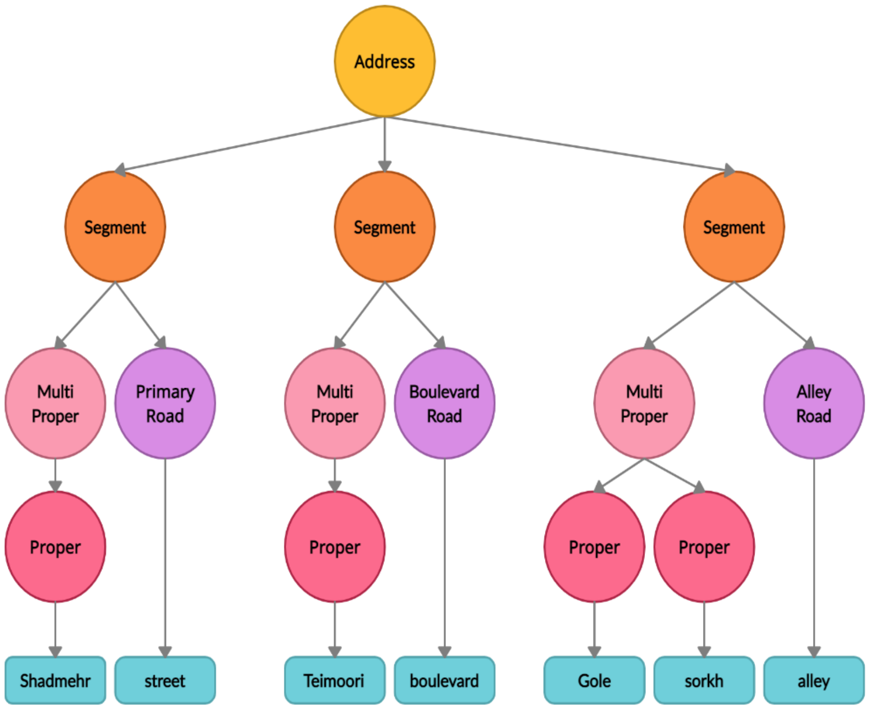

Figure 2. The parse tree for address "Shadmehr Street, Teimoori Boulevard, Gole Sorkh Alley".

To this aim, we develop a compiler based on a LookAhead Left to Right (LALR) parser for parsing a text using this grammar. This component has two modules: Lexer and Parser. The Lexer module tokenizes the input address, and the Parser module parses tokens and converts them into a tree structure. Figure 2 shows the parse tree of the above example.

3.3. Collecting data from map

The previous pipeline component's output is a tree containing different segments of an address. In this step, we send our requests to the Tehran Map API and collect data from this source. This component collects many coordinates that relate to roads and places of the address. Having complex addresses, sending the original raw address to the Map API has no results but sending single segments of that address can achieve relatively relevant results for that part. This component uses the following modules:

- The query generator module: this module, which is the main part of this component, receives addresses in the tree structure and uses two other modules to generate and send appropriate queries to the Map API server. Also this module checks collected results, and can change and resend some queries. Considering the example in Section 3.2, after processing this address in the previous component and creating the tree structure, it becomes clear that this address has three segments: "Shadmehr Street", "Teimoori Boulevard", and "Gole Sorkh Alley". Therefore, this module generates one or several queries for each segment.[3] Also this module can ignore parts of the address like floor number and other parts that the API server cannot provide data for. Finally, this module can attach metadata to collected data from each segment.
- The Geographer module: this module has local information about the Tehran city; e.g., this module knows the geographical area bounds of Tehran's regions. The region of the answer coordinate is an optional field in the program; in case of the availability of this field, it will play an essential role in filtering collected coordinates and limiting them to the specified region.

3.4. Selecting answer coordinate

At this point, several coordinates for each segment of an address are collected. This component proposes an algorithm to select exactly one coordinate among the collected coordinates.

[3] We may have more than one query for a segment in some cases; e.g., if we face more than 20 points in the database that match a segment (because each query can provide up to 20 points from the database) or if we have some geographical direction in the segment (at first, we have a query with geographical direction and after that, we have another query without geographical direction) or something else.

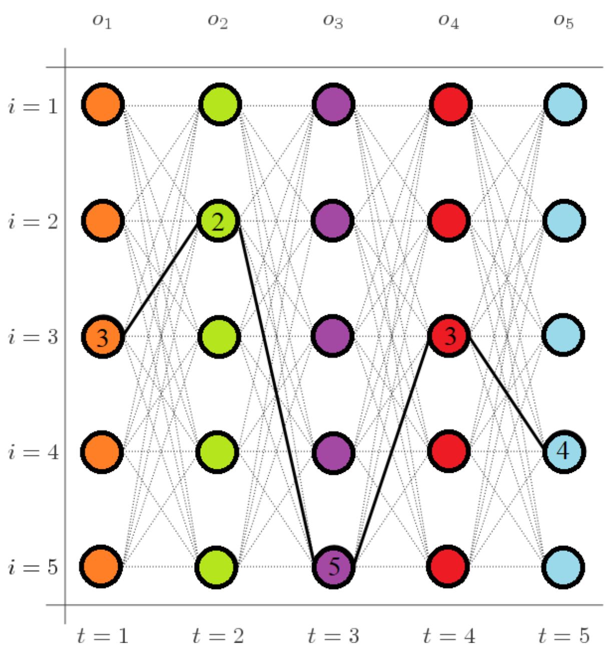

Figure 3. Selecting the most likely sequence of hidden states using Viterbi algorithm [32].

In Persian, when an address is expressed, in fact, a route is expressed. In the Persian addressing structure, the first segment usually expresses a wide area, and the subsequent segments make this area more specific step by step. For example, the first segment of a Persian address after country and city is the main avenue, square or highway. The address then continues with another avenue, street, alley name, and house number; the last segment is usually the floor number. It might include the postal code as well, but very rarely. This structure is entirely different from the English addressing style.

In other words, a Persian address is usually a route from the first segment to the last segment. For each segment, we achieve a list of coordinates from the map. Then we must select a coordinate for each segment to connect these coordinates in their order of appearance in the address ends to the shortest path. For this part, we have developed a distance calculator module based on UTM coordinates (x and y) and geographical coordinates (latitude and longitude). In the next step, the Viterbi algorithm, which works based on dynamic programming, aims to find the shortest path. The algorithm is developed for decoding the hidden Markov model and aims to find the most likely sequence of hidden states. Figure 3 shows a schematic view of selecting the most probable sequence of states by Viterbi [32].

In our geocoding task, we imagine that each segment is a time step, and the set of coordinates collected from the map for that segment forms hidden states related to that time step. By using the Viterbi algorithm, our system selects the best coordinate for each segment, minimizing the overall distance between coordinates.

After finding the shortest path, the selected coordinate in the last segment, which is the most specific segment according to the Persian addressing format, is returned as the answer and the program's output.

Note that it is possible that the program cannot find any coordinate for some segments or finds many coordinates for some segments. In the case of finding several coordinates, based on the available metadata from the query generator, our system filters out some coordinates. If a query is generated for an alley and we have more than 300 coordinates for that query, our system automatically filters out this segment.

In case of finding no coordinate for all segments, the system fails to return an output. Consider the example from Section 3.2 to make the process more precise. The program collects several coordinates for "Shadmehr Street" and "Teimoori Boulevard". It collects another number of coordinates for "Gole Sorkh Alley" too. Several alleys with the name "Gole Sorkh" are located in different locations in Tehran city, but one is close to "Teimoori Boulevard". This alley is probably the answer, and the program returns coordinates corresponding to this alley.

As mentioned, the overall structure of the proposed model is a pipeline of modules. In summary, the preprocessing component makes the given address more clear. The address segmentation component converts the address format from the text structure to the tree structure. Then the next component uses this tree structure to collect coordinates related to each address segment. In the last component, the program selects exactly one coordinate as output.

4. Data

This section describes the datasets, source, structure, and variety and how we have used them. In our system, two types of data are used:

The Map API data (reference data): the Map API is an API that receives a small segment of an address and returns a list of data points. Our system uses this data to find the answer. Each data consists of the following features:

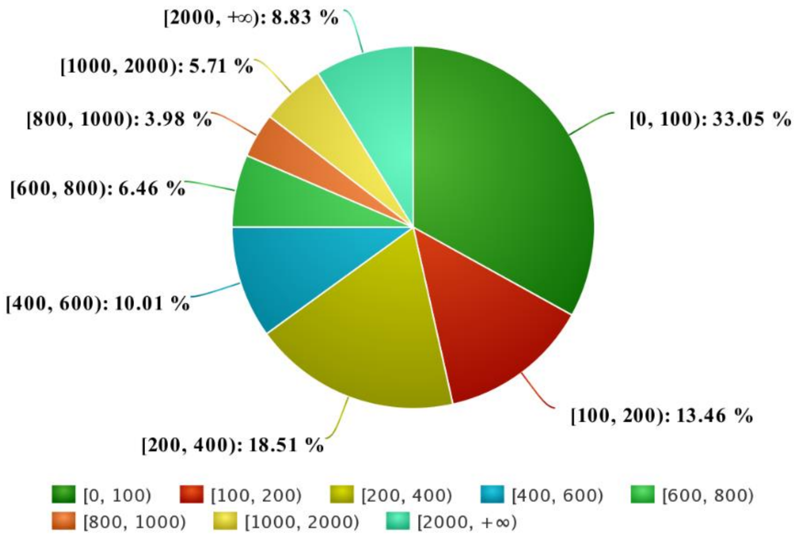

Figure 4. Accuracy of the program according to the distance error of the program.

- Title
- Region ID
- UTM coordinate
- Type (This parameter specifies the type of location that the data point is related to; e.g., a square or an avenue.)

The Map API is a simple API that receives a short text and returns data points whose title is part of the input text; e.g., if we pass “Amir” to the Map API, data with titles like “Amirkabir University” or “Samira Avenue” can be returned. Some features such as filtering the data type are implemented in the Map API.

The Addresses dataset: to develop a parser for Persian addresses, we need to define a context-free grammar. Defining such grammar to cover different formats of Persian addresses requires a dataset of addresses. With such a dataset, we can find important and frequent keywords, proper rules, and essential preprocessing steps. Moreover, we need a dataset to test our system with unseen data labeled with their correct locations on the map. We used the collected data from the 137’s city management system from the Tehran municipality for these two datasets. Tehran’s citizens use this system to report different problems in the city. The reason for using this dataset is that the result of the current project can be used in this system to point an address to the map once the operator has entered the address. We split the dataset into two parts with no overlap. One part defines the context-free grammar, and the other is used as test data. Each data entry in the address dataset consists of the following parts:

- Postal address
- Region ID
- Expected geographical coordinate (expected latitude and longitude)

The program expects postal address and region as input. The program can work without providing the region of the address but specifying the region helps decrease the program’s error. Expected coordinates were used to evaluate the output of the program. In this part, we describe different structures used in some of the addresses given to the program to present the project’s challenges better. The variety of address formats and using old and new names for roads are examples of challenges in this research.

Some addresses consist of only one segment such as “Aftabe Vanak” or “Ghanat Alley”; some belong to a vast area such as “Azadi”, some others consist of many segments, such as “Resalat Highway, Dardasht Crossroad, beginning of the Heidarkhani Street, Shahid Ahadzadeh Street, corner of Azizi Alley” or “intersection of Shahid Sani Street (forty-six Street) and Abdelmajid Goldar Street, Khodabandeh Alley, in front of Number 4”. Some addresses are well-formed such as “north Nabard Street, Aemeh Athar Crossroad, Mohammad Kazemizadeh Alley” but others are not well-structured, such as “Hemmat Highway from east to west, in parallel to the Azadegan Line, entrance of the Azadegan Highway”. Some addresses are incredibly detailed and cannot be found using the map; e.g., “Valiasr Street, in front of the Mellat Park, pink apartment, Number 2665”.

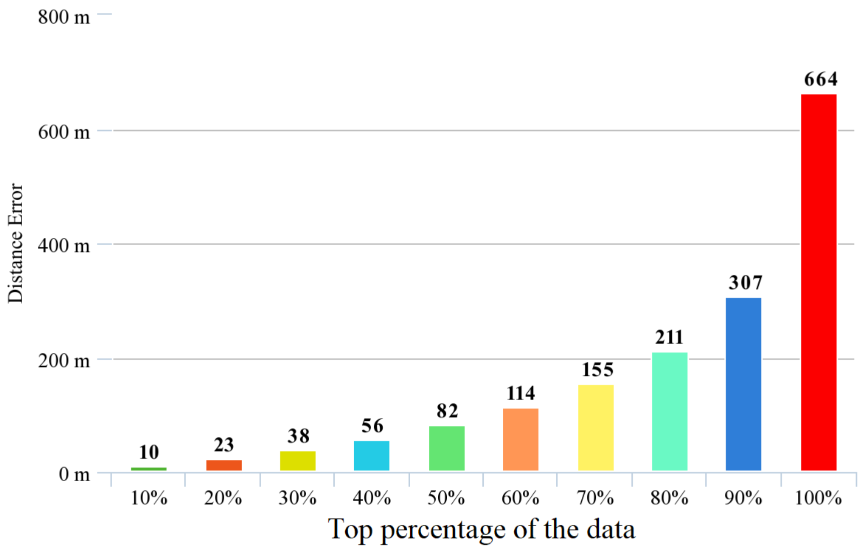

Figure 5. Accuracy of the program according to the percentage of the data.

5. Evaluation

5.1. Setup of experiments

As mentioned in the previous section, one part of the addresses dataset was selected as test data for the final evaluation. As mentioned, we have developed a Python compiler based on an LALR parser. We implemented this parser with the *ply* library. This library has two important sections that cover Lexer and Parser. We import the library *lex* for Lexer and the library *yacc* for the parser.
The API_Interface module (as the name implies) is an interface between the Map service and other program modules. This module aims to ease using the API server and to better maintenance in the future. For connection to the Map API server the *Request* library has been helped.

5.2. Results

To evaluate our system, we used a dataset including 1000 addresses. The addresses are actual samples provided by Tehran Municipality ICT Organization and tagged by their correct coordinate. Each instance of this data has a region id field. Our system found an answer coordinate for 92.9% of the input test, and the system found no answer for the remaining. Clearly, we could increase this number by returning a careless coordinate for any input data. However, we prefer a high precision system that covers a smaller number of data with high accuracy answers rather than a large number of data with low accuracy answers.
For data the program found an answer to, we calculated the error: the distance between the returned coordinate and the expected coordinate for a given data. For calculating this distance, we used the distance calculator module, as explained in Section 3. The results of this evaluation are presented in Table 1. We split the data into different groups in this table according to distance ranges. For better visualization, we presented Figure 4.

Table 1. Accuracy of the program according to the distance error of the program.

Distance error of the program	Percentage frequency of the data
Lower than 100 m	33.05%
Between 100 and 200 m	13.46%
Between 200 and 400 m	18.51%
Between 400 and 600 m	10.01%
Between 600 and 800 m	6.46%
Between 800 and 1000 m	3.98%
Between 1000 and 2000 m	5.71%
Upper than 2000 m	8.83%

About 7% of total data has no answer, and 8.83% of 93% of total data (data with an answer) has a significant distance. These two groups of data (addresses with no answer and addresses with a long distance to the target answer) are the most challenging instances. The main reasons for poor results on such challenging data are as follows:

- Very ambiguous addresses: Some addresses are very ambiguous, and there is no valuable data in Map API for them.

- Very complex segments: Some addresses have very complex segments. We defined a context-free grammar for the designed parser, and it may not be able to parse very complex patterns of a natural language.
- New rare segments: We investigated a considerable amount of real addresses and proposed many rules. Nevertheless, an address may have a segment with a new, rare pattern that confuses our grammar.
- Very dirty address: In case of a very noisy structure in the input data, the preprocessor cannot clean data very well and needs more advanced techniques.

Table 2. Accuracy of the program according to the percentage of the data.

A top percentage of the data	Distance error of the program
10%	10 m
20%	23 m
30%	38 m
40%	56 m
50%	82 m
60%	114 m
70%	155 m
80%	211 m
90%	307 m
100%	664 m

As mentioned before, our system is the first geocoder for the Persian language. Similar methods implemented in other languages cannot be used for Persian because of fundamental differences in Persian addressing format to other languages. Therefore, we have no baseline or competitor method. The Tehran Map API also does not work on the input addresses since they include more than one single segment. As a result, in this section, only the result of our system is presented.

In another part of the evaluation, the distances are sorted in ascending order, and the average of the distances is calculated. The results of this evaluation is presented in Table 2 and Figure 5.

In this section, we must point to the influence of outlier data. In Table 2, the distance error of 90% of data is about three hundred meters on average but the distance error of all data is dramatically more than six hundred meters. Also in Table 2, about 9% of data has more than 2000 meters distance with the correct coordinate. Therefore, the influence of a small number of outlier data is high in this research study.

For a better understanding of the results, we provide two visual examples. In Figure 6, we present our system's expected location and output for "Shahid Rajaei, Beheshti Square, Abrisham Street, Hamedani Street, Baqeri Alley, in front of No. 12". The green pin represents the expected location, and the red pin represents our prediction. The distance between these two points is 52 meters. In this example, the program finds the entire route except for the last segment ("in front of No. 12"). The locations of "Abrisham Street", "Hamedani Street", and "Baqeri Alley" can be seen in this figure. Figure 7 presents another example: "Padegan-e-Valiasr Street, corner of Booshehr Street". In this example, our system's error is near zero meters.

We had several challenges in developing this tool. The most critical challenges are as follows:

- Data Correctness: Some test data have the wrong expected location. Furthermore, some test data addresses are ambiguous, like a small part of a complete address. In these cases, the accuracy of the program decreases unfairly.
- Reference Data: As we said before, our reference data is not a complete and correct tool; i.e., some roads are not defined in the reference data.
- Variety of Writing: Variety and irregularity are a part of any natural language, which is more pronounced in Persian [3].

In conclusion, the average distance error of the program is about three hundred meters for 83% of all data. This accuracy is good enough to be used as an assistant for speeding up finding the location by a human agent. However, it still needs improvement to be used within a fully automated system.

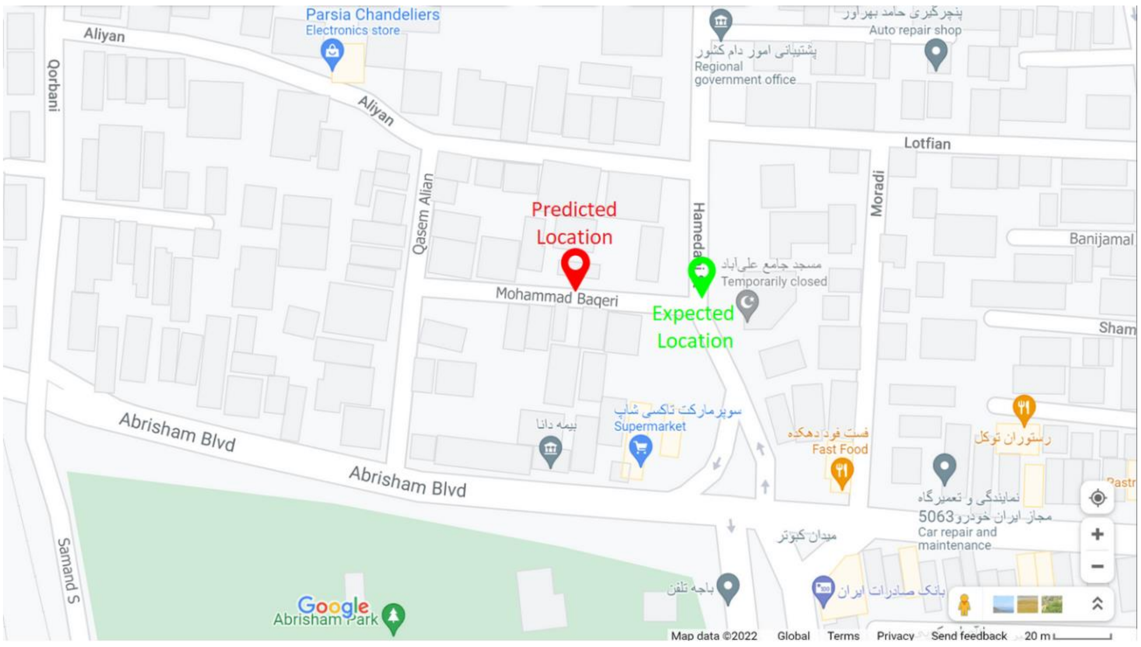

Figure 6. The predicted location (red pin) and the expected location (green pin) for "Shahid Rajaei, Beheshti Square, Abrisham Street, Hamedani Street, Baqeri Alley, in front of No. 12"

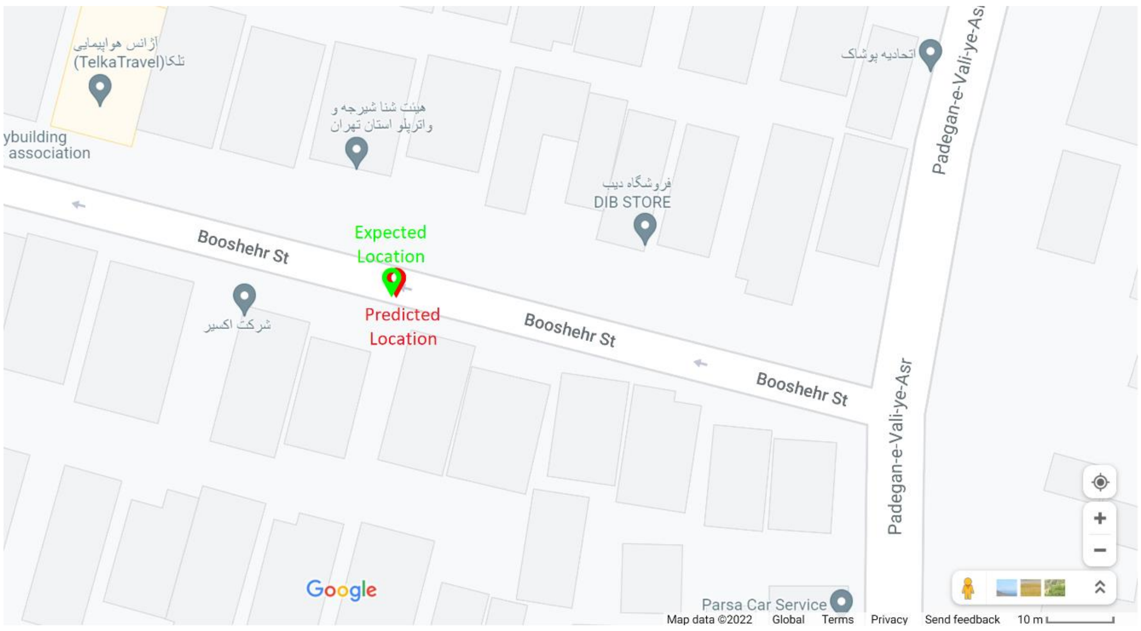

Figure 7. The predicted location (red pin) and the expected location (green pin) for "Padegan-e-Valiasr Street, corner of Booshehr Street".

6. Conclusion

In this paper, we proposed a rule-based geocoder tool for Persian addresses. The existing Map API is too simple and just accepts small phrases, which is the main reason for developing a new framework for Persian geocoding. We discussed the main challenges including different formats to specify an address from one person to another, the lack of standard naming convention for roads, the non-persistency of names of roads, and the absence of a complete and uniform system for data of cities. We specified the components of the system. "Pre-processing", "Parsing", "Searching", and "Selecting" work based on the LALR parsing algorithm for the address segmentation and the Viterbi algorithm for path selection. We evaluated our system on test data of 1000 samples. We showed that our proposed framework had reached an appropriate error rate with the test data, about 300 m for 83% of test data. To the best of our knowledge, the proposed system is the first Persian geocoder. Therefore, we do not have any baseline systems for comparison.

Our results showed that an advanced rule-based system with an immature reference dataset that benefits from different algorithms to process the data could be suitable for some applications. Although the accuracy of our system is not a perfect fit for sensitive applications like post services, it can be used as an assistant to speed up and improve user experience. Moreover, TehranGeocode is good enough for big data analysis since a very accurate output is not expected in such applications.

For future works, we suggest improving the Parser module to better parse complex and rare phrases. Since our parser has context-free grammar with defined rules, it cannot segment some complex or rare phrases and requires a more powerful grammar or parser. Moreover, we suggest adding a confidence score in output to consider the tradeoff between response rate and accuracy with attention to any application. Our system, with minimal changes, can be used for the other cities of Iran; the main requirement to generalize the geographical area of using our method is the existence of reference data for the other cities.

Acknowledgment

The project funding and the required data were provided by Tehran Municipality ICT Organization.

References

[1] A. Kebe, R. Faye, and C. Lishou, “Multi agent-based addresses geocoding for more efficient home delivery service in developing countries,” in *e-infrastructure and e-services for developing countries*, Springer International Publishing, 2019, pp. 294-304.

[2] S. Khan, L. Pinault, M. Tjepkema, and R. Wilkins, “Positional accuracy of geocoding from residential postal codes versus full street addresses,” *Health reports*, vol. 29, no. 2, pp. 3-9, 2018.

[3] M. Ghayoomi, S. Momtazi, and M. Bijankhan, “A study of corpus development for Persian,” *International Journal on Asian Language Processing*, vol. 20, no. 1, pp. 17-33, 2010.

[4] O. Charif, H. Omrani, O. Klein, M. Schneider, and P. Trigano, “A method and a tool for geocoding and record linkage,” in *2010 second iita international conference on geoscience and remote sensing*, 2010, vol. 1, pp. 356-359.

[5] T. R. Cortes, I. H. D. Silveira, and W. L. Junger, “Improving geocoding matching rates of structured addresses in Rio de Janeiro, Brazil,” *Cadernos de Saúde Pública*, vol. 37, 2021.

[6] A. Kebe, R. Faye, and C. Lishou, “A multi-agent-based approach for address geocoding in poorly mapped areas through public company data,” *International Journal of Information Technology and Applied Sciences (IJITAS)*, vol. 3, no. 1, pp. 1-9, 2021.

[7] M. E. I. Malaainine, and H. Lechgar, “Conception of geocoding matching algorithm for Casablanca City-Morocco,” in *2020 ieee international conference of moroccan geomatics (morgeo)*, 2020, pp. 1-4.

[8] D. K. Matci, and U. Avdan, “Address standardization using the natural language process for improving geocoding results,” *Computers, Environment and Urban Systems*, vol. 70, pp. 1-8. 2018.

[9] F. Alcántara, A. Molina-Villegas, and V. Muñiz, “A vector semantics approach to the geoparsing disambiguation task for texts in Spanish,” in *Proceedings of the 1st international conference on geospatial information sciences*, 2019, vol. 13, pp. 47-55.

[10] B. Kilic, and F. Gülgen, “Investigating the quality of reverse geocoding services using text similarity techniques and logistic regression analysis,” *Cartography and Geographic Information Science*, vol. 47, no. 4, pp. 336-349, 2020.

[11] K. Lee, A. R. C. Claridades, and J. Lee, “Improving a street-based geocoding algorithm using machine learning techniques,” *Applied Sciences*, vol. 10, no. 16, p. 5628, 2020.

[12] P. Li, A. Luo, J. Liu, Y. Wang, J. Zhu, Y. Deng, and J. Zhang, “Bidirectional gated recurrent unit neural network for Chinese address element segmentation,” *ISPRS International Journal of Geo-Information*, vol. 9, p. 635, 2020.

[13] L. Nizzoli, M. Avvenuti, M. Tesconi, and S. Cresci, “Geo-semantic-parsing: Ai-powered geoparsing by traversing semantic knowledge graphs,” *Decision Support Systems*, vol. 136, p. 113346. 2020.

[14] Z. Yan, C. Yang, L. Hu, J. Zhao, L. Jiang, and J. Gong, “The integration of linguistic and geospatial features using global context embedding for automated text geocoding”, *ISPRS International Journal of Geo-Information*, vol. 10, no. 9, p. 572, 2021.

[15] A. P. Wheeler, M. Gerell, and Y. Yoo, “Testing the spatial accuracy of address-based geocoding for gunshot locations”, *The Professional Geographer*, vol. 72, no. 3, pp. 398-410, 2020.

[16] B. Wilson, and N. Wilson, “An iterative approach to the parcel level address geocoding of a large health dataset to a shifting household geography,” Working paper, Center for Economic Information, University of Missouri-Kansas City, 2017.

[17] B. kilic, and F. Gülgen, “Accuracy and similarity aspects in online geocoding services: A comparative evaluation for Google and Bing maps,” *International Journal of Engineering and Geosciences*, vol. 5, pp. 109-119, 2020.

[18] V. Cetl, T. Kliment, and T. Jogun, “A comparison of address geocoding techniques case study of the city

of Zagreb, Croatia," *Survey Review*, vol. 50, no. 359, pp. 97-106, 2018.

[19] L. Li, W. Wang, B. He, and Y. Zhang, "A hybrid method for Chinese address segmentation," *International Journal of Geographical Information Science*, vol. 32, no. 1, pp. 30-48. 2018.

[20] G. Dumedah, N. N. Y. Binche, G. M. Bob-Milliar, S. Iddrisu, E. K. Twumasi, and J. A. Boateng, "The case of electoral polling station data for geocoding in facilitating accessibility to social, economic and cultural opportunities in Ghana," *African Geographical Review*, pp. 1-14, 2022.

[21] H. L. Nguyen, D. Tsolak, A. Karmann, S. Knauff, and S. Kühne, "Efficient and reliable geocoding of German twitter data to enable spatial data linkage to official statistics and other data sources," *Frontiers in sociology*, vol. 7, 2022.

[22] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th international conference on neural information processing systems*, Curran Associates Inc. 2013, vol. 2, pp. 3111-3119.

[23] Q. Le, and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st international conference on international conference on machine learning* JMLR.org, 2014, vol. 32, pp. II-1188-II-1196.

[24] S. Rashidian, X. Dong, A. Avadhani, P. Poddar, and F. Wang, "Effective scalable and integrative geocoding for massive address datasets," in *Proceedings of the 25th acm sigspatial international conference on advances in geographic information systems*, Association for Computing Machinery, 2017.

[25] S. Rashidian, X. Dong, S. K. Jain, and F. Wang, "Easergeocoder: integrative geocoding with machine learning (demo paper)," in *Proceedings of the 26th acm sigspatial international conference on advances in geographic information systems*, Association for Computing Machinery, 2018, pp. 572-575.

[26] K. Clemens, "Deriving spelling variants from user queries to improve geocoding accuracy," in *Proceedings of the 5th international conference on geographical information systems theory, applications and management, GISTAM 2019, Heraklion, Crete, Greece, May 3-5, 2019*, pp. 53-59.

[27] N. Firouraghi, N. Bagheri, F. Kiani, L. Goshayeshi, M. GhayourMobarhan, K. Kimiafar, S. Eslami, and B. Kiani, "A spatial database of colorectal cancer patients and potential nutritional risk factors in an urban area in the middle east," *BMC Research Notes*, vol. 13, no. 1, pp. 1-3, 2020.

[28] L. Goshayeshi, A. Pourahmadi, M. Ghayour-Mobarhan, S. Hashtarkhani, S. Karimian, R. S. Dastjerdi, B. Eghbali, E. Seyfi, and B. Kiani, "Colorectal cancer risk factors in north-eastern iran: A retrospective cross-sectional study based on geographical information systems, spatial autocorrelation, and regression analysis," *Geospatial health*, vol. 14, no. 2, 2019.

[29] H. Shabanikiya, S. Hashtarkhani, R. Bergquist, N. Bagheri, R. VafaeiNejad, M. Amiri-Gholanlou, T. Akbari, and B. Kiani, "Multiple-scale spatial analysis of paediatric, pedestrian road traffic injuries in a major city in north-eastern iran 2015–2019," *BMC public health*, vol. 20, no. 1, pp. 1-11, 2020.

[30] P. Tabari, H. Shabanikiya, N. Bagheri, R. Bergquist, S. Hashtarkhani, F. Kiani, A. Mohammadi, B. Kiani, "Paediatric, pedestrian road traffic injuries in the city of Mashhad in north-eastern Iran 2015–2019: a data note," *BMC research notes*, vol. 13, pp. 1-3, 2020.

[31] H. Khodadadi, and V.Derhami, "Improving Speed and Efficiency of Dynamic Programming Methods through Chaos," *Journal of AI and Data Mining*, Shahrood University of Technology, vol. 9, no. 4, pp. 487-496, 2021.

[32] J. J. Rebello, "*Matlab central file exchange*," 2021. [Online]. Available: https://www.mathworks.com/matlabcentral/ fileexchange/47359-viterbi-algorithm

کدگذاری آدرس فارسی: یک روش مبتنی بر تجزیه LALR و برنامه‌نویسی پویا

علیرضا مازوچی[۱]، سارا بوربور[۲]، محمدرضا غفرانی[۱] و سعیده ممتازی[۱،*]

[۱] دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی‌تکنیک تهران)، تهران، ایران.

[۲] مرکز آمار و رصدخانه شهری تهران، تهران، ایران.

ارسال ۲۰۲۲/۱۲/۱۱؛ بازنگری ۲۰۲۳/۰۱/۱۱؛ پذیرش ۲۰۲۳/۰۳/۱۸

چکیده:

تبدیل آدرس پستی به یک مختصات جغرافیایی، که اصطلاحا کدگذاری جغرافیایی نامیده می‌شود، به‌عنوان یک ابزار مفید در بسیاری از کاربردها مورد استفاده قرار می‌گیرد. توسعه یک کدگذار جغرافیایی برای کشورهای در حال توسعه که از قالب آدرس‌دهی استاندارد پیروی نمی‌کنند با چالش‌های جدی همراه خواهد بود؛ علاوه بر چالش‌های معمول در تمام وظایف پردازش زبان طبیعی، می‌توان به دادگان مرجع ناکافی و فقدان استاندارد نام‌گذاری یکپارچه معابر اشاره کرد. در این مقاله یک کدگذار جغرافیایی فارسی به نام TehranGeocode ارائه می‌شود. براساس دانش نویسندگان، TehranGeocode اولین کدگذار جغرافیایی برای زبان فارسی است. با توجه به ساختار غیراستاندارد آدرس‌های فارسی، یک آدرس باید به قطعات کوچک شکسته شود، اطلاعات هر قطعه از دادگان مرجع جمع‌آوری گردد، و برای یافتن مختصات متناظر با آدرس، اطلاعات جمع‌آوری شده به یکدیگر متصل شوند؛ برای دست‌یابی به این هدف، سامانه پیشنهادی بر اساس تجزیه آدرس و برنامه‌نویسی پویا توسعه داده شده است. چارچوب پیشنهادی قادر است ۸۳٪ آدرس‌ها را با خطای کمتر از ۳۰۰ متر در نقشه پیدا کند که برای این وظیفه با چالش‌های مذکور نتایج امیدوارکننده‌ای است.

کلمات کلیدی: کدگذاری جغرافیایی، فارسی، تجزیه آدرس، آدرس پستی، پردازش زبان طبیعی، الگوریتم ویتربی.