Collecting data with postal address on the Internet

Valentina R. Fedorova¹ and Roman K. Fedorov^{2[0000-0002-2944-7522]}

¹ Irkutsk State University, Irkutsk, Russia ² Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk 664033, Russia fedorov@icc.ru

Abstract. An information system has been developed for collecting data with postal addresses. The system makes it possible to increase the efficiency of publishing new information in the form of maps. Found addresses require expert judgment for the suitability and usefulness for publication. The software system can be applied to texts imported from different sources, such as Microsoft Word documents, PDF documents, social networks, etc.

Keywords: GIS, parsing postal address, web scraping, web data extraction, web information extraction.

1 Introduction

Now there are many different systems that present information on the cartographic maps. For example 2GIS, Yandex.Maps, Google Maps, etc. Basically, these systems include a city map with an address plan and a description of organizations and services. There are also geoportals of municipal and regional authorities, which provide information about the economy, resources, infrastructure, etc. The main disadvantage of these systems is low efficiency, new information appears after a long time on the map, or does not appear at all. A huge amount of information is published on the Internet every day in the form of text with addresses. For example, news sites publish articles within a few hours after an event, posts with comments on specific events are posted on social networks, and articles about the any work appear on municipal and regional portals every day. Therefore, it is useful to automatically collect data on the Internet and publish them on a map.

2 Review of existing methods

Let us consider the existing text processing methods that search for postal addresses of geographic objects.

Parsing addresses with "fuzzy regular expressions". The main idea of this method is that it is necessary to create a dictionary with possible spellings of streets. Estimating analogy of names is carried out according to Levinstein distance [1]. Any expres-

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

sion and subexpression is enclosed in parentheses, the type of expression is determined by the character following the opening parenthesis. For example, the symbol "=" means that words must be present either in the given order or in reverse, and expressions in parentheses with a "?" sign mean optional parts, i.e. if they are absent, they are skipped and do not affect the error count. This method takes into account spelling errors, various permutations in street names, for example, st. October Revolution and st. Revolution October, as well as omission of any part of the name. The disadvantage of this method is that it is necessary to manually create expressions for specific streets, and also only the string with the address is submitted for parsing, and not the full text [2].

Parsing of postal addresses using the FIAS database [3]. In this method, the strings containing the address are divided into address elements, then each element is checked in turn for the presence in the FIAS database. If an element is in the database, then its level in the hierarchy is remembered, and the next element is searched for with a larger hierarchy value and a fixed PARENTGUID equal to the GUID of the previous found element. But this method does not provide for parsing the text, it can process only a line with a correctly formed address [4].

Parsing the postal address using neural networks [5]. In this method, the address is encoded into a number. A code is assigned for each category of information strings, numbers, abbreviations, etc. Next, the list of addresses is scanned, and addresses matching unique patterns are extracted from them. After that, each template indicates to which category the analyzed part of the address belongs. Thus, a training dataset is prepared. The disadvantage of this method is that training a neural network requires a large training dataset with marked-up text.

3 Program system for collecting data with postal address

A software system has been developed that collects data on the Internet, searches for postal addresses and publish them on a map. The software system consists of three components:

- component for collecting data on the Internet designed to download HTML pages, save in the file system and parse and search for links to other HTML pages;
- component for extracting postal addresses from text designed to parse the text of HTML pages and extract postal addresses;
- component for geocoding addresses converts a postal address into geographic coordinates using geocoding services (for example, OpenStreetMap, Yandex, etc.).

Let's consider more detail the component of extracting postal addresses from text. The scheme of the component work for extracting addresses from text is shown in Fig. 1.



Fig. 1. The scheme of the component work for extracting addresses from text.

Initialization. At this stage, data about settlements, streets and houses are downloaded to arrays. It should be considered that, the name of a settlement may consist of several separate parts, and, the names of settlements will not always be written in the nominative case. Therefore, another array is created in which names will be stored in parts, and the endings will also be cut off, that is, "Большое Голоустное" will be in two separate items, "Больш" and "Голоустн" (In Russian, word endings change). At the same time, in an array with names divided into parts, each element of the array has a link to the full name. An example of the distribution of settlements over two arrays is shown in Fig. 2.



Fig. 2. Name arrays.

Dividing into tokens. The file with the text of the HTML page is read and divided into tokens. Then each token is filtered and the HTML markup is discarded.

Extracting sentences from the text. The postal address is assumed to be within one sentence. The text is divided into sentences, taking into account abbreviations.

Search for a postal address. The input of the algorithm is a sentence represented by a set of tokens. Each token in the sentence is checked for presence in arrays with settlements (cities) and streets. If there is such a value, then the array element index is stored. If there are accepted names of settlements or their abbreviations near the current token, then this token is checked for presence only among settlements. Also, if there is a generally accepted street name or its abbreviation, this token is checked for presence only among the streets. Each token can have several possible matches in the arrays. To store this information, a special object is used - the sentence context, which

consists of possible settlements, streets and houses (see Fig. 3). If there are settlements in the context, the search of streets is performed only within these settlements.



Fig. 3. Sentence context.

It is assumed that there is the following sentence: "Planned repair work is being carried out at the address Русская Аларь, ул. Заречная, 4". Since each token is matched in the database, the following values will be included in the context for the "Русск" token: settlements "Русская Аларь" and "Русский Мельхитуй", and street "Русская". There are two values for the "Аларь" token: "Аларь" and "Русская Аларь". Since there is a generally accepted abbreviation of the street "ул" (like st.) before the "Заречная" token, this token will be checked for presence only in the array with streets, therefore only "Заречная" street will be included in the context. Since there is only "Заречная" street in the settlement "Русская Аларь", it will refer only to it, and there is indeed a house with number 4 on this street, therefore, in the context, the house will refer to "Заречная" street. Context items that do not refer to anything will be removed from it.

In the case of a compound name of a settlement or a street, each part of it is checked in the array; a situation may arise when there are several possible items in the array. If these item names consist of several parts, then the presence of all these parts in the sentence is checked. If a complete match is found, then the remaining items are discarded (see Fig. 4).



Fig. 4. Example of parsing compound name.

Suppose that the sentence contains the name of the settlement "Русская Аларь". Since each token is matched separately, the following items are available in the database: "Аларь" and "Русская Аларь" for the "Аларь" token, as well as two items: "Русская Аларь" and "Русский Мельхитуй" for the "Русская" token. For all items from the database, the presence of all its constituent parts in the sentence is tested. Among all these items, only the "Русская Аларь" remains.

After all settlements and streets have been saved in context, the house number is searched for. Each token stores its position in sentence. The house number is assumed to immediately close to the street, so the sentence string is truncated. In the substring a house number is matched by a regular expression. The matched numbers (for example 267/5, 18A, 144r/3) are checked for presence in the array of houses placed on the streets which are in the context.

After all the sentence tokens have been processed, the context is analyzed. In the context of the sentence, there can be several related combinations of settlements, streets, and houses. If at least one part of the address is missing in the combination, then the combination is not complete and is discarded. Initially, all possible variants of settlements are added to the context, but as soon as a street is located, all those settlements which do not have such street are removed from the context. If the street is not linked to any settlement, it will also be removed from the context.

After removing incomplete combinations, there may still be several variants in the context, so it is necessary to identify the most relevant ones. For this, the estimation is introduced for each element of the context. It is preferable to choose compound names of settlements or streets. If the settlement or street is a compound name, then their score is increased by 2 points, otherwise, if the name is single, then - 1 point. The estimation is carried out according to the formula (2), where x_1 is the estimation of the settlement, which takes a value equal to 2 if the name of the settlement is composite, and 1 - otherwise, x_2 is the estimation of the street, which takes a value equal to 2 if the name street is composite, and 1 otherwise.

$$f(x_1, x_2) = x_1 + x_2 \tag{1}$$

This is done in order not to miss a compound name, for example, if the sentence contains "Октябрьской революции" Street, then two meanings will appear in the context: "Октябрьской революции" Street and "Революции" Street. Of these two streets, the compound name will be the most relevant, so it gets more points in the estimation. After the scoring procedure, only the values with the highest score remain in the context. If the context is fully assembled, that is, it contains all three components of the address available in one instance, then the address is saved.

4 Approbation

During the work of the address search information system, 12918 URLs were found, 995 HTML pages were downloaded to the device. The downloaded HTML pages contained 681 addresses. In fig. 5 a map with tagged addresses and URLs attached to them is shown.



Fig. 5. Found addresses.

On URL https://irkobl.ru/news/floods.php (see Fig. 6) the system found addresses that are collection points for humanitarian aid to victims of floods in the Irkutsk re-

gion. Thus, the information system automatically generated a map of the collection points for humanitarian aid (see Fig. 7).

Перечень пунктов сбора гуманитарной помощи пострадавшим гражданам в результате		
паводков в Иркутской области		

№ п/ п	Наименование учреждения	Адреса пунктов сбора гуманитарной помощи
1	Комплексный центр социального обслуживания населения Балаганского района	рп. Балаганск, Орджоникидзе, 12, (39548)50024, (39548)50408
2	Комплексный центр социального обслуживания населения Баяндаевского района	с. Баяндай, ул. Борсоева, д.11
3	Комплексный центр социального обслуживания населения г. Бодайбо и Бодайбинского района	г.Бодайбо, ул.Октябрьская ,21 "А" тел.: 8(39561)5-10-48; сот: 89246367245(заведующая отделением срочной помощи - Исакова Елена Валерьевна)
4	Комплексный центр социального обслуживания населения Боханского района	п. Бохан, ул. Школьная, д.6 п.,(39538)81517
5	Комплексный центр социального обслуживания населения г. Братска и Братского района	г. Братск, ул. Южная, д. 81 (каб. 24), тел. (3953) 45-89-81
		г. Братск, ул. Мечтателей, д. 17A (каб. 3), тел. (3953) 36-57-07
6	Комплексный центр социального обслуживания населения "Веста"	г. Ангарск, 189 квартал, д. 15а, (3955)541308
7	Комплексный центр социального обслуживания населения Жигаловского района	п. Жигалово, ул. Партизанская, 56; тел: (39551)3-12-99

Fig. 6. HTML page https://irkobl.ru/news/floods.php.



Fig. 7. The collection points for humanitarian aid.

5 Conclusion

The developed information system for collecting data with postal addresses makes it possible to increase the efficiency of publishing new information in the form of maps. Found addresses require expert judgment for the suitability and usefulness for publication. The software system can be applied to texts imported from different sources, such as Microsoft Word documents, PDF documents, social networks, etc.

The work was carried out with the support of RAS (projects: AAAA-A17-117032210079-1, AAAA-A19-119111990037-0), RFBR (projects:18-07-00758-a, 17-57-44006-Mong-a) and Ministry of Science and Higher Education of the RF, the grant for implementation of large scientific projects on priority areas of scientific and technological development (project no. 13.1902.21.0033). Results are achieved using the Centre of collective usage «Integrated information network of Irkutsk scientific educational complex».

References

- 1. Levenshtein V.I.: Binary codes with corrected dropouts, insertions and character replacements. Reports of the USSR Academy of Sciences (1965).
- Parsing addresses with fuzzy regular expressions, https://habr.com/ru/post/192518, (last accessed 2020/05/08).

- 3. Parsing postal addresses from a string in C#, https://habr.com/ru/post/232347, (last accessed 2020/05/08).
- 4. Parsing the postal address into components, https://basegroup.ru/community/articles/addresses, (last accessed 2020/05/08).
- Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).