PROCEEDINGS OF SCIENCE

A Novel Fuzzy Chinese Address Matching Engine Based on Full-text Search Technology

Xiaojing Yao[1][2]
Institute of Remote Sensing and Digital Earth
National Engineering Research Center for Remote Sensing Applications,Beijing,100101, China
E-mail: yaoxj@radi.ac.cn

Xiang Li
Institute of Remote Sensing and Digital Earth, Beijing,100101, China
E-mail: lixiang_whu@163.com

Ling Peng
Institute of Remote Sensing and Digital Earth
National Engineering Research Center for Remote Sensing Applications,Beijing,100101, China
E-mail: plqiqi@126.com

Tianhe Chi[3]
Institute of Remote Sensing and Digital Earth
National Engineering Research Center for Remote Sensing Applications,Beijing,100101, China
E-mail: chith@126.com

The ability to locate addresses is one of the most important features in an urban geographic information system. Since Chinese geocoding problem cannot be handled by the European geocoding method, some Chinese scholars did special researches on Chinese geocoding. Current researches all focus on address standardizations and models, and pay less attention to the user input and result control. We designed a novel fuzzy Chinese address matching engine to give a freedom of user input and result control. The engine is composed of an index builder and a retrieval locator based on full-text search. Furthermore, three kinds of fuzzy match methods (Searching-Box Fuzzy Single Match, One-to-One Fuzzy Single Match, and Table-Form Fuzzy Batch Match) are implemented. Through more than 50,000 pieces of address-point data from eight districts of Beijing for testing, this engine compared to traditional database retrieval system shows obvious advantages: (1)It has higher match efficiency for dealing with large data; (2)It offers greater freedom on user input and result control. In the quality test, when the threshold of fuzzy matching degree is higher than 0.75, the accuracy rate reaches to 100%, with matching rate 83% and recall rate 92% respectively.

CENet2015
12-13 September 2015
Shanghai, China

[1] Speaker
[2] Corresponding Author
[3] This research was supported by the fire control management and rescue command system based on Internet of things (2011FU125Z24).

1. Introduction

The ability to locate addresses is one of the most important features in an urban geographic information system. Geocoding, also called address match, which uses a geographical description to map a space position with X and Y coordinates [1], is the key to solve address location problem.

European countries start researches on geocoding earlier than China. For example, the project of "Topologically Integrated Geographic Encoding and Referencing (TIGER)" in America [2], the "Postal Code Address Data (PCAD)" in Canada [3], the "Geocoded National Address File (GNAF)" in Australia [4] all belong to this category. Besides, lots of commercial organizations such as Google, Yahoo, Microsoft and ESRI, also released their own free geocoding engines for the government work, city planning and facility location.

However, the European geocoding technology cannot be used in China directly because of the reasons: (1) Chinese address standardizations and geocoding database lack integrated planning. For example, in 1988, Beijing firstly proposed a series of urban address data collecting rules [5-6] in the "Geocoding Project for Beijing". Later, Shanghai, Zhejiang and Hong Kong also did similar work, and published masses of different standardizations. So far, there still exists an insurmountable barrier to build uniform standards about geocoding. (2) Compared to English address expression, Chinese expression has more freedom and no space. European geocoding engine cannot handle Chinese geocoding problem directly. Therefore many Chinese scholars do researches on Chinese geocoding and present some practical models, such as Hierarchical Address Match Models [7, 8], Information Retrieval Address Match Models [9] and Human Intelligence Address Match Models [10].

Nevertheless, current researches all focus on address standardizations and models, and pay less attention to the user input and result control. On one hand, users always input address mixed with un-normal terms. Limited result is returned if the address construction is over-pursued. On the other hand, different volume of input data require different match accuracy, efficiency and operation styles. In order to solve the above problems, this paper proposes a novel fuzzy Chinese address matching engine composed of an index builder and a retrieval locator based on full-text search. This engine has three matching modes: Searching-Box Fuzzy Single Match, One-to-One Fuzzy Single Match, and Table-Form Fuzzy Batch Match. Since this engine offers simple address matching components and gives a factor to adjust the results, it can be easily used to develop complex geocoding services.

The paper is organized as follows. The core algorithm of full-text search technology - Correlation Score Algorithm is presented firstly in Sect. 2. Section 3 discusses the framework of fuzzy Chinese address matching engine. We propose the implement of the index builder and the retrieval locator. The retrieval locator contains three fuzzy matching functions, which is the most import part of the paper. Section 4 presents a series of experiments to demonstrate the efficiency and the accuracy of the proposed function. Finally, the study is summarized in Sect. 5.

2. Correlation Score Algorithm based on Vector Space Model (VSM)

Full-text search engine creates index and records the frequency and location for each word in a file, so it's much faster to pinpoint the user input. Chinese full-text search technology includes invert index, word segmentation, information retrieval and etc. It's mainly used to match key word in the website. It's also apt to match address due to the similar process. This paper selects Lucene engine to customize our address matching engine because of its good extendibility and concurrency support.

The kernel of Lucene is the correlation score mechanism. It calculates the correlation score between the input query and the document. The algorithm adds adjustable stimulating factors based on VSM [11], which can enhance the flexibility of the matching algorithm. The formula is below:

$$\begin{aligned}&Score(q,d)=\\&coord(q,d)\times queryNorm(q)\times\\&\sum_{k\text{ in }q}[tf(k\text{ in }d)\times idf(k)^2\times k.getBoost()\times norm(k,d)]\end{aligned}\qquad(2.1)$$

In Eq. (2.1), k means key word in query q or document d. $coord(q,d)$ stands for the ratio between the matched key words in q and the key words in q; $queryNorm(q)$ stands for the normalized factor of q. The longer the query is (it means q containing more key words), the smaller the factor value is. $tf(k\text{ in }d)$ stands for the frequency of k in d. $idf(k)$ stands for the frequency of inversed documents. The less likely the key word k appears in the whole database, the bigger the value is. $k.getBoost()$ stands for the weight of the key word k. $norm(k,d)$ stands for the normalized factor of d. The longer the document is (it means d containing more key words), the smaller the value is.

In conclusion, if a key word appears more frequent in the active document than in others, the score of the key word will be much higher to the active document than to others. In addition, if the matched key words of a query sentence hold a higher percentage in the active document than in others, the score of the query will be much higher as well. Based on the formula, each address in the database can be treated as a document. Then after the word segmentation and invert indexed process, the above algorithm can be used to calculate the matching degree between the user input and each address piece from the database. This formula is also the theoretical foundation of the fuzzy Chinese address matching engine.

3. The Framework of Fuzzy Chinese Address Matching Engine

The fuzzy Chinese address matching engine has two main threads: (1) The creation of geography index; (2) Geography searching and location. The framework is as Fig. 1.

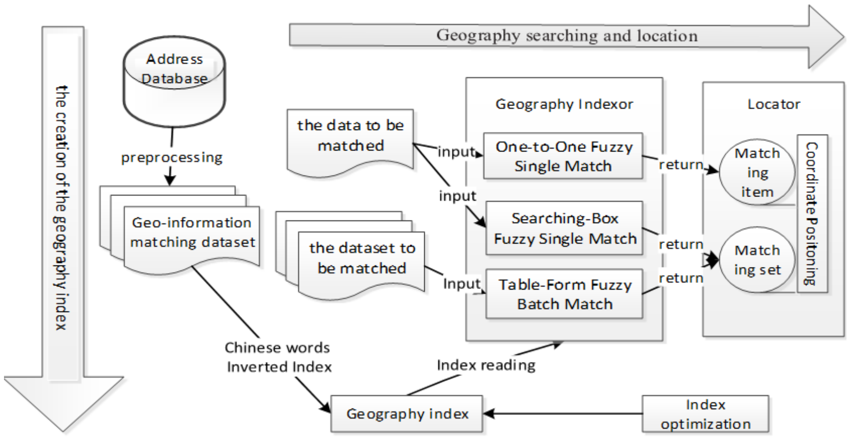
Geography searching and location
the creation of the geography index
Address Database
preprocessing
Geo-information matching dataset
the data to be matched
the dataset to be matched
Geography Indexor
One-to-One Fuzzy Single Match
Searching-Box Fuzzy Single Match
Table-Form Fuzzy Batch Match
input
input
Input
return
return
return
Locator
Matching item
Matching set
Coordinate Positoning
Chinese words Inverted Index
Index reading
Geography index
Index optimization

Figure 1: The framework of fuzzy Chinese address matching engine

3.1 Creation of Geography Index

Geography index is created for the whole address database. It puts a piece of “dictionary clothes” on the address database, so it avoids catastrophic traversals during matching. The creation of geography index happens when updating the address database. The whole process is divided into 3 steps:

1. Preprocess the address database. We first filter the non-standard addresses in the original database and normalize them, then encode the addresses according to their hierarchal elements. After that, the normalized address can by expressed by a series of codes. In order to improve the matching rate, each normalized address record is supplied with alias, English abbreviation and etc. The item which can be located should contain fields including address code, address name, location description (such as standard address and administrative area),

remark (such as abbreviation and alias) and location coordinates (X, Y coordinates). Accordingly, the geographic coordinates table, the administrative area table, the street table, the building table are built. Among them, the geographic coordinates table is the only table storing coordinates and the completed "related codes", so other tables can be connected logically.

2. Create the word segmentation dictionary. The address segmentation is similar to the ordinary sentence segmentation, but it needs to think about the characters of the address composition. First, most of the addresses can be treated as the combination of different hierarchical place names, so an address can be expressed as follows: address = place name 1 + place name 2 +…+ place name 3. For example, a Chinese address expression "Beijing City Chaoyang District jianguomenwai Street 1 No." can be divided into "Beijing City", "Chaoyang District", "Jiangguomenwai Street", "No. 1". When the administrative level goes down, the location specificity increases. This kind of specificity can be evaluated by their occurring probability in the database, so it's necessary to identify these independent place name units. Second, many studies about address match shows that, a normalized hierarchical place name usually consists of common name and special name. For example, "Beijing City" consist of one special name "Beijing" and one common name "City". The special name can be accurately described as a geography object, serving as the main matching word. The common name can mark single work unit, serving as the cutting word. Using the standard address as an example, the best cutting should be "Beijing/ City/ Chaoyang/ District/ Jianguomenwai/ Street/ 1/ No.". According to this feature, in order to increase the segmentation accuracy of common name and special name, it's needed to put the common geography segmentation words, as well as administrative division names, community names, and street names into the word segmentation dictionary. Common geography segmentation words are as follows:

Administrative area: Province, city, autonomous region, special administrative region, district,

alliance, autonomous prefecture, village, town, sub-district office etc.

Community: li, plot, park, lane, apartment, house, garden, cottage etc.

Street: road, avenue, street, alley, hutong etc.

Number: number, #, building, dormitory, room, hall etc.

Geographic object: mansion, square, center, bar, restaurant, hotel|, bureau, company etc.

Spatial direction: east, west, south, north, compound direction(such as southeast, northwest)

Spatial relation: contain, disjoint, joint, overlay, cross

Word segmentation dictionary is not only for the creation of the geography index, but also for the user input segmentation. Spatial orientation and spatial relation segmentation words can provide the correction information of a location expression. They make the result closer to the real geography location description. For example, "No.1, Jianguomenwai Street, Chaoyang District, Beijing City, 10m east" is not a normal location, because there's an offset description "10m east" in the sentence. This offset description can be segregated out and used to calculate the real geography location.

3. Build the geography indexer. First, a "Document" object *d* is created, and the address data are written into the "IndexWriter" of *d*. Second, the fields of address code, address name, location description, remark are selected as geography index fields. An appropriate Chinese tokenizer (for example, Paoding Tokenizer), along with the word segmentation dictionary, are used to build "Inverted Index". At last, the grouping index, caching index and incremental index of Lucene are used to optimize the index.

3.2 Geography Searching and Location

In practical applications, fuzzy match is more widely used than exact match. Fuzzy match includes three modes: Searching-Box Fuzzy Single Match, One-to-One Fuzzy Single Match, and Table-Form Fuzzy Batch Match.

Searching-Box Fuzzy Single Match means finding the similar result set from the database after input a query sentence. The result elements rank in descending order according to the correlation score. Usually, users have the right to re-choose. The first element in result set has the highest score, which is called “the most relevant solution”. One-to–One Fuzzy Single Match means finding the target object from the database after input a query sentence. However, just using the most relevant solution as the result of One-to-One Fuzzy Singe Match is improper because it may not be “the target solution”. For example, if a user intend to find “Guangtai Animal Pharmaceutical Company” by inputting “Guangtai Pharmaceutical Company”, the most relevant solution is “Guangtai Biology Pharmaceutical Company”. Either uncompleted input or insufficient portion of matched words will cause the disagreement between “the target solution” and “the most relevant solution”. Under this condition, manual interaction is necessary. Table-Form Fuzzy Batch Match is the repetition of One-to-One Fuzzy Single Match. In the process of mapping non-spatial table-form data, users pay more attention to how many “target solutions” can be returned from the whole table-form data, rather than how many “relevant solutions” can be returned from one record of the data.

3.2.1 The Algorithm of Searching-Box Fuzzy Single Match

The query fields are the whole geography indexed fields or some of them, and the returns are the top *n* results with the most highest correlation scores. The matching process uses the “MultiFieldQueryParser” index analyzer to do multi-field fuzzy match. The algorithm is below.

Algorithm 1. The Algorithm of Searching-Box Fuzzy Single Match (fuzzySingleMatch)

```
Input:
- fields denotes the query fields
- q denotes the query sentence
- n denotes for the number of results
- $T_{Chinese}$ denotes a Chinese Tokenizer
Output:
- the top n results with the most highest correlation scores $R = \{r_1, r_2, ...r_n\}$
$R \leftarrow \varnothing$
Create a MultiFieldQueryParser P assigned $T_{Chinese}$ and fields parameters
Use $P_{T,fields}$ to parse q and get a Query Q
Create an IndexSearcher $S_Q$ to do the query task
Fetch the top n Score Documents by $S_Q$ and store them in $D_{scoreDocs} = \{d_1, d_2, ...d_n\}$
for each d in $D_{scoreDocs}$ do
    $R = R \cup \{\text{the index result of } d\}$
end for
return R
```

3.2.2 The Algorithm of One-to-One Fuzzy Single Match

One-to-One Fuzzy Single Match has three possible situations. First, the engine finds out the target solution, which means a successful match. Second, the query fetches several results when the query sentence is uncompleted. The most relevant solution may not be the target solution, so manual operation is needed. Third, the engine cannot find the target solution, which means a failed match. We need to judge the completeness of the input firstly to identify the second situation, and judge the threshold to distinguish the first and the third situation. The second step needs a judgment factor. Lucene will make the score divided by the maximum score if it’s greater than 1. Although the normalized factor of query $queryNorm(q)$ and the normalized factor of document $norm(k,d)$ are introduced when calculating scores, it makes no sense to compare the scores of the most relevant solutions from different users’ inputs. To solve this problem, we introduce a factor called Fuzzy Matching Degree（μ_{FMD}） based on the correlation score algorithm. The formula is as follows:

$$\mu_{FMD} = S_1 / S_2 \qquad (3.2)$$

In Eq.(3.2), S_1 denotes the score of the most relevant solution, S_2 denotes the exact match score when the current most relevant solution is re-calculated. μ_{FMD} ranges from 0 to 1. $r < \mu_{FMD} \le 1$ means a successful match(r is an empirical threshold given by user). Especially, $\mu_{FMD} = 1$ means an exact match. The algorithm (Algorithm 2, containing 26 lines) contains two key steps:

Keyword stream completeness detection(line 3-10)

After segment the query sentence into key words, we put each key word into "BooleanQuery" index analyzer assigned "MUST" parameter to get the "IndexResults" including all keywords. If the size is more than 1, then it needs the user interaction; otherwise it passes the detection and goes to the threshold detection.

Keyword stream threshold detection(line 11-28)

If the result set passes the completeness detection, we put each keyword into "BooleanQuery" index analyzer assigned "SHOULD" parameter to get the "IndexResults" including at least 1 keyword. If the size of the results set is more than 0 and the μ_{FMD} of the most relevant solution is larger than *r*, then it passes the detection; else if the size is equal to 0, or the size is more than 0 with the μ_{FMD} of the most relevant solution less or equal to *r*, it fails.

Algorithm 2. The Algorithm of One-to-One Fuzzy Single Match(oneToOneFuzzySingleMatch)

Input:

- *fields* denotes the query fields
- *q* denotes the query sentence
- *r* denotes the threshold of μ_{FMD}
- $T_{Chinese}$ denotes a Chinese Tokenizer

Output:

- the target solution object $O_{target}(r_{target}, f)$, where r_{target} indicates the target index result, *f* indicates the matching situation flag, $f = 0$ indicates for the third situation, $f = 1$ indicates for the second, and $f = 2$ indicates for the first.

```
1: $O_{target}(r_{target}, f) \leftarrow \varnothing$
2: Create a QueryParser P assigned $T_{Chinese}$ and fields parameters
3: Use $P_{T,fields}$ to parse q and get a BooleanQuery Q
4: Q assigned "MUST" parameter for keyword stream completeness detection
5: Create an IndexSearcher $S_Q$ to do the query task
6: Fetch the top 2 Score Documents by $S_Q$ and store the results in set $D_1$ % In order to save the processing time, only get the top 2 results%
7: if size($D_1$) > 1 then
8:     $O_{target}.f \leftarrow 1$
9:     $D_1 \leftarrow fuzzySingleMatch(fields, q, 10)$
10:    $O_{target}.r_{target} \leftarrow$ the selected node from $D_1$
11: else
12:    Q assigned "SHOULD" parameter for keyword stream threshold detection
13:    Fetch the top 2 Score Documents by $S_Q$ and store the results in set $D_2$ % In order to save the processing time, only get the top 2 results%
14:    if size($D_2$) = 0 then
15:        $O_{target}.r_{target} \leftarrow \varnothing$
16:        $O_{target}.f \leftarrow 0$
17:    else
18:        $\mu_{FMD} = getFMD($the first node of $D_2)$
19:        if $\mu_{FMD} > r$ then
20:            $O_{target}.r_{target} \leftarrow$ the first node of $D_2$
21:            $O_{target}.f \leftarrow 2$
```

```
    else
     $O_{target}.r_{target} \leftarrow \varnothing$
     $O_{target}.f \leftarrow 0$
    end if
  end if
end if
return $O_{target}(r_{target}, f)$
```

3.2.3 The Algorithm of Table-Form Fuzzy Batch Match

Rather than regurgitating, Table-Form Fuzzy Batch Match is an iteration process of One-to-One Fuzzy Single Match.

4. Results and Analysis

The experiment data is used to test our methods consists of 50000 pieces of address-point data from eight districts of Beijing. We considered the efficiency of the engine rather than the errors caused by the incompleteness of data.

Efficiency test and analysis

In this experiment, we repeated each algorithm 5 times and took the average value. The result is below.

time(t/s)
DataVolume(million)
	One-to-One Fuzzy
	Single Match
		Searching-Box Fuzzy
		Single Match
			Database SQL
			search

1	0.01209	0.01201	2.21872
2	0.01351	0.01296	4.38291
3	0.01462	0.01373	6.19681
4	0.01549	0.01431	8.59281
5	0.01593	0.01459	10.21781

Table 1: The comparison of our methods and SQL searching test

We can conclude from the table above that, the matching engine proposed in the paper has obvious higher efficiency than the common SQL data system. The reversed index avoids a lot of traversals, so the effect of the basic data amount can be ignored. This advantage is more remarkable in a batch match.

Quality test and analysis

We get a piece of enterprise table-sheet from government data system containing about 3000 items with enterprise name and address description fields. We repeated the fuzzy batch match method 5 times and took the average value. Fig. 1.2 shows the accuracy rate, recall rate and matching rate changes when assigning different μ_{FMD} threshold.

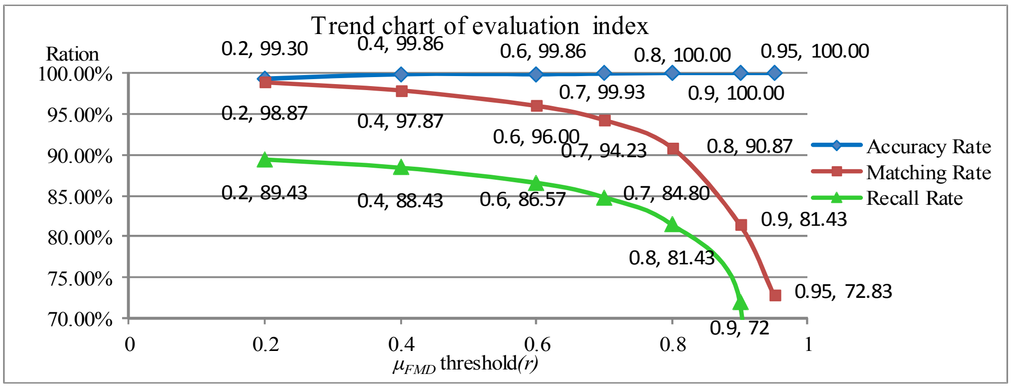

Figure 2: Evaluation index trend chart of fuzzy batch match

There's a negative correlation between accuracy rate and recall rate. With the increasing of *r* (μ_{FMD} threshold), the recall rate and matching rate are decreasing. In practical application, accuracy rate is relatively more important, so we should first guarantee the accuracy rate, then the recall rate and matching rate. In Fig. 1.2, we know that when *r* is higher than 0.75, the accuracy rate reaches to 100%, with matching rate 83% and recall rate 92% respectively. The threshold meets the needs of practical application. We can also adjust the threshold according to different situations.

5 Conclusion

This paper proposes a novel Chinese fuzzy address matching engine based on full-text search with three kinds of fuzzy match methods: Searching-Box Fuzzy Single Match, One-to-One Fuzzy Single Match, and Table-Form Fuzzy Batch Match. The core idea of this engine is: by creating index for the address segments, we get the appearance frequency of each segment in the address database, so there's no need to analyse the hierarchy roles of the address element during match. In addition, we creates a factor called μ_{FMD} to control the accuracy rate and matching rate for Fuzzy Batch Match. The result shows that these algorithms, compared to the traditional methods, have higher match efficiency and give higher degree of freedom to user input.

Chinese address match is a complex problem. The match engine in this paper focuses on the flexibility of the user input and result control. It lacks the consideration of error tolerance, such as house number match, abbreviation, input mistake, which will be improved in our future work.

References

[1] T. H. Grayson ,*Address matching and geocoding*[R]. Massachusetts Institute of Technology Department of Urban Studies and Planning (2000).

[2] U.S. Census Bureau TIGER/ Line Shapefiles Technical Documentation[EB/OL]. http://www.census.gov/geo/ maps-data /data/ pdfs/tiger /tgrshp2013/TGRSHP2013_TechDoc.pdf (2013).

[3] Canada Postal Guide - Addressing Guidelines[EB/OL]. http://www.canadapost.ca/tools/pg/manual/PGaddress-e.pdf. 2007.

[4] PSMA Australia Limited G-NAF Product Description[EB/OL]. http://www.psma.com.au/?product=g-naf (2012).

[5] China's state bureau of surveying and mapping[S]. Coding Rules for urban geographical features - City roads road intersections blocks and municipal pipe lines (GB/T14395-2009). Standards Press of China, Beijing (2009).

[6] China's state bureau of surveying and mapping[S]. Rules of coding for address in the common platform for geospatial information service of digital city (GB/T23705-2009). Standards Press of China, Beijing (2009).

[7] H. Yu , Q. Qi, Y. Li, *Study on city address geocoding model based on street*[J]. Journal of Geo-information Science. 15(2), 175-179 (2013).

[8] L. Zhang, S. Wu, *Research on place names and address segmentation in geocoding system*[J]. Science of Surveying and Mapping. 35(2), 46-48 (2010)(In Chinese).

[9] C. Zou, G. Zhu, S. Zhao, *Research on customized query of geographic name and address based on search engine*[J]. Bulletin of Surveying and Mapping. (8), 92-94,124 (2014)(In Chinese).

[10] Z. Song, *Address matching algorithm based on Chinese natural language understanding*[J]. Journal of Remote Sensing. 17(4), 788-801 (2013).

[11] B. Possas, N. Ziviani, W. Meira, B. Ribeiro-Neto, *Setbased vector model: An efficient approach for correlation based ranking*[J]. ACM Transactions on Information Systems. 23(4), 397-429 (2005).