Knowledge-Based Systems 174 (2019) 57–72

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Extracting geographic features from the Internet: A geographic information mining framework

Ying Zhang [a,*], Qunfei Ma [a], Yao-Yi Chiang [b], Craig Knoblock [b], Xin Zhang [c], Puhai Yang [a], Minghe Gao [a], Xiang Hu [a]

[a] School of Control and Computer Engineering, North China Electric Power University, 2 Beinong Road, 102206, Beijing, China
[b] Dana and David Dornsife College of Letters Arts and Sciences, University of Southern California, Los Angeles, USA
[c] School of Computer Science and Technology, Changchun University of Science and Technology, 130022, Changchun, China

ARTICLE INFO

Article history:
Received 26 October 2018
Received in revised form 21 December 2018
Accepted 23 February 2019
Available online 14 March 2019

Keywords:
Geographic information mining framework
Place entity extraction
Place-name dataset
Geographic information mining

ABSTRACT

In this paper, we propose a Geographic Information Mining framework to contribute some exploratory results concerning harvesting the featured place information entities from the Web. In the framework, we suggest an iterative geographic information mining model reflecting the data evolution along the mining process. Associating the iterations, we propose a set of methodologies and integrate them into the processing onto solving the critical issues concerning collecting data, filtering irrelevant samples and extracting featured entities. According to the experiments, the contribution brings in a sound systematic solution to enrich the existing digital gazetteers as complete as Google Maps.

1. Introduction

Digital gazetteers are the structured dictionaries of named places. As defined by Goodchild and Hill [1], a gazetteer is composed of three core components, i.e., namely place names, place types and geospatial locations. The primary purpose of a gazetteer is to translate the informal place names and place categories to the formal georeferencing of mathematical schemes and well-known types [2,3]. Seen that the explosion of web-based services on the Internet, more and more gazetteer applications require a very high level of completeness, but the current gazetteers still lack the local place names used in everyday conversations [4,5]. Besides, the existing structured dictionaries only provide static information but do not incorporate frequent updates. Thus, there is a strong need to enrich gazetteers with abundant up-to-date local place entries for improving the timeliness and integrity of the structured geographical data. One critical issue is effectively and efficiently building up place-name datasets.

Conventionally, place-name datasets can be obtained from some structured data sources, such as DBpedia,[1] LinkedGeoData,[2] Wikimapia,[3] Google Places API,[4] and OpenStreetMap.[5] However, these structured sources only provide static information but do not always incorporate the timely updates. Not only that, although some commercial sources, such as the Google Places API, maintain high-quality location data, many restrictions are issued for obtaining and using their data (i.e., usage limits). In contrast, the volume of vernacular publicly available on the Web is enormous and grows rapidly. Meanwhile, the up-to-date information of places is commonly released on the Web and accepts frequent renewing since the unstructured web pages. The great utility of the Web in conducting studies on the data collection and extraction has been widely recognized [6–8].

In this paper, we firstly propose a Geographic Information Mining (GIM) framework inspired by the CRISP-DM (Cross Industry Standard Process for Data Mining) methodology [9] for extracting geographic information. The GIM framework enables us to build up the multiple dimensions of modeling geographic information mining processes, i.e., the phase-oriented model, the generic/specific task-oriented model and the instance-oriented model. Associated with search engines, each of the models specify and arrange the focal phases, tasks and instances as the entries integrating the potential methodological solutions to contribute the effort onto mining the geographic information from web pages. Under the phase-oriented modeling dimension, the information mining process consists with as a serial of phases. Each of the phases specifies a sequential business reflecting the evolution from raw data to structured knowledge. Seeing that information mining would not be completed when a solution is just deployed, we specially bring in an iterative scheme of structuring the procedural modules (i.e., tasks and instances) under the

* Corresponding author.
E-mail address: yingzhang@ncepu.edu.cn (Y. Zhang).

[1] http://wiki.dbpedia.org/Downloads.
[2] http://linkedgeodata.org/sparql.
[3] http://wikimapia.org/api.
[4] https://developers.google.com/places/.
[5] http://www.openstreetmap.org.

https://doi.org/10.1016/j.knosys.2019.02.031
0950-7051/© 2019 Elsevier B.V. All rights reserved.

task-oriented and instance-oriented modeling dimensions. The iterative scheme reveals the sustaining evolution of knowledge discovery through gaining and utilizing the previously obtained mining achievement(s) during mining information [10], and it also exposes that information mining is an open process taking advantage of various solutions to collect, extract and treat data samples.

As the foundation for constructing GIM framework, CRISP-DM methodology, as a bridge between the gap of business problems and data mining objectives, provides a standard reference model to translate business problems into a set of data mining tasks and is independent of technological aspects. It is inclined to improve the effectiveness and efficiency of managing data mining projects. In contrast, GIM framework proposed in this paper is designed for organizing the potential methodological solutions of resolving geographic information mining issues from web pages and it supports to load the compatible methods with the iterative modeling dimensions. In other words, GIM could be adopted as a technological roadmap toward extracting the accurate and timely geographic features, associated locations and feature types from freely available online data to build the place-name datasets, which mainly concerns the key issues as follows:

The *first issue* is to collect and identify the appropriate place types for composing the keywords in the format of <Street Names><City names><Place Types> to search for the initial the geographic information.

The *second issue* is to filter out the "noisy" information mixed with the obtained search results for preserving the meaningful and valuable information.

The *third issue* is to extract the necessary geospatial locations from the various web pages [11] in consideration of the distinct structures in different web pages.

The *fourth issue* is to address the problem of place name disambiguation (also called toponym disambiguation), i.e., to accurately identify the place names as low ambiguous as possible.

To completely present our research progress for readers' better understanding onto the contribution in this paper, we briefly clarify the previous study in the following Section 2. Around the relevant research topics, we discuss the related work in the same section. In Section 3, we elaborate the GIM framework and propose a set of methods integrated into the framework for building place-name datasets relying on the previous research result. Then in Section 5, we expound a set of experiments to verify the proposed contribution and make a comprehensive analysis onto the experimental results. In the last section, we conclude the research contribution with highlighting the characteristics of the proposed framework and state the future work concerning improving the performance for enriching gazetteers.

2. Previous study and related work

2.1. The previous work

Our previous work proposed in [12] provided a set of preliminary solutions/results corresponding to the mentioned critical issues being concerned with the proposed GIM framework.

(1) To compose the proper searching keywords in the format of <Street Names><City names><Place Types> with web search engines, the previous work collect the place types through receiving the manually input.

(2) To filter out the webpages containing many real-estate listings for obtaining more usable searching results, the previous work selected to query the Google search engine with the recently sold house addresses (the addresses can be collected from any real-estate website). With the returned results providing the domain names of popular real estate websites, we are able to eliminate the "noisy" webpages.

(3) To retrieve place addresses from the unstructured webpages, we presented an algorithm to parse the place address as a line starting with a number and containing the city name as an address in the condition where the length of the line is less than a given threshold. Meanwhile, we also used a similar means to determine whether or not a combination of two lines represents one address.

(4) To extract place names exactly from the given webpages, we simply assumed that the corresponding webpage title was the place name if the returned webpage contained only one address and the address was the same as the address used to query the search engine API.

With answering the four critical issues, the above-introduced work brings our inceptive reflection of establishing the GIM framework in this paper. On this basis, we will depict the framework with highlighting the progress with the contrast between the previous work and current contribution in Section 3.

2.2. Related work review

A considerable number of researchers have collected data for generating gazetteers from structured Internet sources [13–15]. The approach mentioned in [16] extracted points of interest from a set of popular Web sources including DBpedia, OpenStreetMap, Wikimapia, Google Places, Foursquare, and Eventful. The first two sources provide SPARQL endpoints, and the latter four sources offer RESTful API. Blessing and Schütze [17] used Google APIs to query the Web and generate place name variations. Their queries had the names of places as positive terms and had the stopwords as negative terms. The stopwords are used to exclude webpages that make the query result noisy. Gelernter et al. [18] presented a method to enrich a gazetteer by identifying sources of novel local gazetteer entries in crowdsourced OpenStreetMap and Wikimapia geotags. However, these volunteered geographic information are not complete and up-to-date. Through comparing the result obtained from the methods brought in by the GIM framework,we find that both OpenStreetMap and Wikimapia, as the data sources are not as latest and complete as ours.

Datasets mined with web crawling tools can be considerably different from data sets retrieved via the exposed query APIs. Thus, many researchers took advantage of search engines to query the Internet to extract information for enriching gazetteers [19–22]. For instance, Uryupina [23] put forward an approach to the automatic acquisition of geographical gazetteers from the Internet. A new gazetteer can be learned from a small set of pre-classified examples by applying bootstrapping techniques. However, bootstrapping could produce a drift toward a noisy result due to the inferred examples. Brindley et al. [24] discovered neighborhood place names from addresses found in web pages. They extracted postal addresses from the Web and created relatively simple linguistic models to produce neighborhood definitions. Popescu et al. [25] presented an automated technique for creating and enriching geographical gazetteers. Their technique merges disparate information from Wikipedia, Panoramio, and Web search engines to identify geographical names.

With the mined data, named entity disambiguation is necessary to identify and determine the appropriate sense of entities for further computation corresponding to the context [26]. Wang et al. [27] presented a topic model for named entity disambiguation in the field of community question answering. The model performed learning in an unsupervised manner but could take advantage of weak supervision signals estimated from the metadata of community question answering and knowledge bases. Hung and Chen [28] proposed a word sense disambiguation (WSD) method for sentiment classification. This paper argued that the task of word sense disambiguation should be done before a proper sentiment score or sentiment orientation for a word in SentiWordNet can be justified. It built WSD-based SentiWordNet lexicons by taking advantage of three WSD techniques based on the context of word-of-mouth documents. Gutiérrez et al. [29] proposed an unsupervised approach to solve semantic ambiguity issue based on a multidimensional network through analyzing different lexical resources. Concerning the place name disambiguation issue, the researches related to gazetteers enrichment identify the issue as Toponym Resolution (TR) [30]. The three directions in TR employed heuristic techniques, machine learning and semantic Web separately [31,32]. Despite their simplicity, the methods based on heuristics have shown a relatively good performance in some text domains [33,34]. Two types of the research works took advantage of machine learning. Some researches usually depend on supervised approaches that require annotated datasets. For instance, Overell and Rüger [35,36] proposed co-occurrence models generated from Wikipedia to solve the place name disambiguation problem by supervised learning techniques. In order to take advantage of the unlabeled data and avoid the need of annotation, a number of methods use a small lexicon such as a small gazetteer [37–40] as seeds to extract place names. In contrast, without using a gazetteer, DeLozier et al. [41] presented a toponym resolver that used the geographic profiles of the local clusters to build a system that computes the overlap of all geo-profiles in a given text span. For semantic aspect, Zaila and Montesi [42] built a geographic ontology GeoNW by combining GeoNames, WordNet and Wikipedia resources, and based on this GeoNW, an algorithm for solving toponym disambiguation was presented. Different from the discussed traditional methods, we suggest to avoid ambiguity problem by using queries of <Street Names><City names><Place Types>. Our provided solution does not suffer from the problem of ambiguous place names since the three components in the given queries mutually restrict each other. It means that given a specific address and a place type, the possibility that the extracted place names are ambiguous is small.

3. The GIM framework

In this paper, we propose a Geographic Information Mining (GIM) framework for locating our contributed methodologies as a systematic solution to the various issues concerned in building up place-name dataset (Seen in Fig. 1). The framework provides a set of modeling dimensions elicited by the contribution from [9] in terms of a hierarchical complexity of processing potential geographic information. The modeling dimensions are:

(1) **Phase-oriented modeling dimension**: it provides the scheme of modeling the total geographic information process with four phases, i.e., *objective definition*, *iterative processing*, *evaluation and refinement* and *deployment*. The phases imply a serial of critical business carrying out one after another to achieve the preset milestones. The dimension reflects the evolution process from raw data to structure knowledge, during which we employ the conventional process of knowledge discovery and data mining methodologies (e.g., [43] and [44]) and build up the phase-oriented reference model. Specially, the potential iterations are programmed between "iterative processing" phase and "evaluation and refinement" phase in consideration of the fact that geographic information mining would produce some unforeseen findings and often require the continuous computations.

(2) **Generic task-oriented modeling dimension**: it unfolds the "iterative processing" phase and provides a hierarchical modeling scheme of decomposing the critical activities during the "iterative process" phase. Under the dimension, one or more groups of tasks are organized and each group is designed as an implementation of geographic information mining methodology to cover one kind of geographic information mining scenario. In the framework, the groups could be created and combined dynamically according to the specific problem to ensure the flexibility of leading various types of geographic information solutions.

(3) **Specific task-oriented modeling dimension**: Within one group, i.e., referring to an adopted methodology during mining geographic information, the procedures are arranged as a number of task suites. Each suite consists with three actions that are interrelated in an iterative manner and specifies the determined actions iteratively carrying out the methodology in a particular engagement. In other words, one group of task suites is modeled as an iteration chain.

(4) **Instance-oriented modeling dimension**: Referring to each action within one task suite, it is configured with the specific parameter(s) and reflects the actual occurrence during the execution of geographic information mining, i.e., an instance is acting the corresponding actions and meanwhile keeping the coordination with the other two instances. The instances are designed and categorized as *Collection (C)*, *Treatment (T)* and *Extraction (E)* to form a serial of iterations, which implies the essential behaviors of coping with data.

During constructing the systematic solution of mining geographic information under the GIM framework, we take advantage of web search engines for multiple times to gather the intermediate data associating extracting and mining the place entities, such as the place types.

As follows, we will depict the modeling regulation of the GIM framework and elaborate the proposed methodological solutions to solve the particular research issues simultaneously.

3.1. Geographic information mining process

Along the geographic information mining process, a set of raw data should be assembled for the further intermediate data entities collection and extraction to achieve the final place-name dataset establishment, i.e, implementing the structured knowledge discovery. We model the process with four phases (i.e., objective definition, iterative processing, evaluation and refinement, and deployment) reflecting the critical activities executed in turn (see Fig. 1).

Objective definition. With analyzing the raw data toward discovering the potential interests in it, we arrange the "objective definition" as the first phase to determine the problem specification(s) and compose the strategy of coping with the problem(s). Both the specification(s) and the strategy would elicit the objectives stated in form of questions:

(1) *What kinds of data organization to be built up?*
(2) *What kinds of hidden information to be mined?*

The first question concerns the necessities of dealing with data for proceeding with employing the further procedures of mining target information, while the second question specifies the structure and content of the final results through mining procedures. In this paper, we kick off the practice under the GIM framework by answering the questions mentioned above as follows:

(1) The raw data mainly originates from the web pages returned by search engines that receive the keywords in the format of <Street Names><City names><Place Types>. Most of the preliminary data would emerge as unstructured since they are gathered through employing search

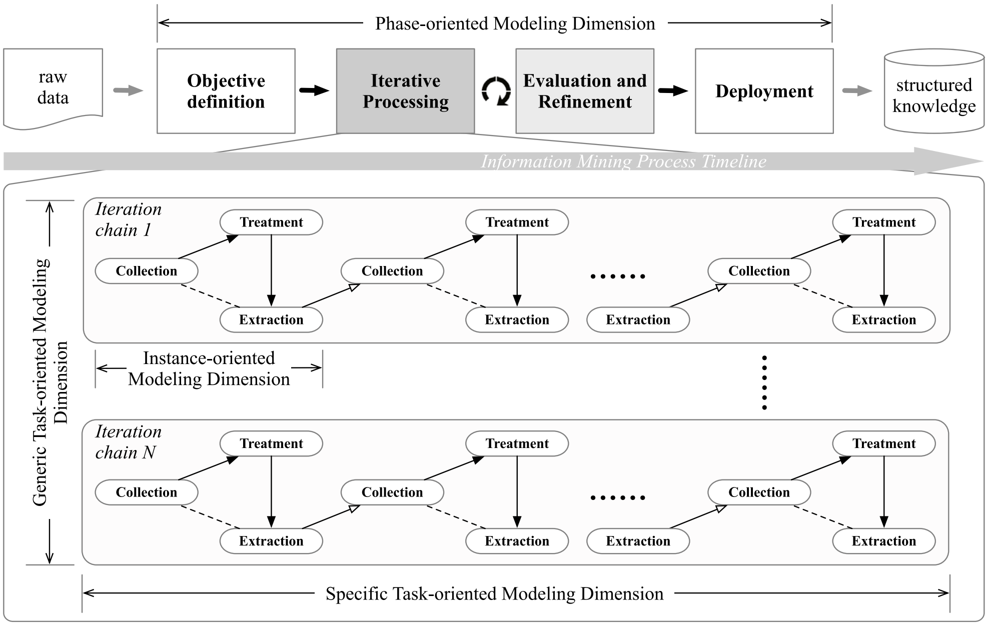

Phase-oriented Modeling Dimension
raw data
Objective definition
Iterative Processing
Evaluation and Refinement
Deployment
structured knowledge
Information Mining Process Timeline
Iteration chain 1
Treatment
Collection
Extraction
......
Instance-oriented Modeling Dimension
Iteration chain N
Generic Task-oriented Modeling Dimension
Specific Task-oriented Modeling Dimension

Fig. 1. The GIM framework.

engines providing plenty of web pages. Some filtering and formatting procedures would thus be adopted. Through the intermediate procedures, the data would evolve into the structured entities. In this paper, the mined place-name entities are organized as JSON-encoded entities.

(2) Through collecting the preliminary data associated with employing search engines and filtering the irrelevant web pages, we retrieve the place addresses from the valid pages and then mine the place names with taking advantage of both title and abstracts of the target web pages provided by search engines. The place names are the outcome to achieve our target of enriching place-name datasets.

Iterative processing. With the answers obtained through the objective definition phase, the second phase, i.e., "iterative processing", then starts. The phase reflects and arranges the critical activities of coping with data according to the previously specified objective definition, and it brings in the problem-solving pattern of iteratively executing activities toward the objectives. During the iterative processing phase, the iterations are implemented with the *phase-oriented ones* and *task-oriented ones*.

Evaluation and refinement. The phase-oriented iterations are implemented with the close link between the iterative processing and the evaluation and refinement phases. The iterations indicate that the processing made onto data would be performed multiple times according to the conclusion made from the "evaluation and refinement" phase. In other words, it is essential to evaluate the produced outcome from the iterative processing phase for determining whether the objectives could be achieved. In this paper, we propose an approach of constructing place-name datasets proceeding to develop the previous research work [12] to solve the critical issues concerned during the iterative processing phase.

Since the task-oriented iterations would be unfolded during the iterative processing phase and meanwhile reflect the activities under the task-oriented and instance-oriented modeling dimensions, we will not continue with expounding them and disturb the introduction to general geographic information mining process and we will elaborate the corresponding issues of task-oriented iterations in Section 3.2.

Deployment. Through the iterative processing and the evaluation and refinement phases, the place-name entities are provided as the meaningful outcome and organized as the input to the "deployment" phase, through which we are enabled to gain the structured knowledge for the further business objectives. In this paper, we visualize the place-name entities on maps to reflect the intuitive outcome from mining geographic information and then are going to contribute the built place-name dataset to enrich the existing gazetteers.

3.2. Intermediate data processing

In this section, we resolve the iterative processing phase and depict the in-line iterative procedures of collecting, treating and extracting data under the task-oriented and instance-oriented modeling dimensions. As illustrated by the geographic information mining process in Fig. 1, the iterative processing phase consists with one or more generic tasks, i.e., a set of generic tasks could be dynamically arranged corresponding to the conventional problem scenarios during mining geographic information under the generic task-oriented modeling dimension. With analyzing the representative literature devoted to enriching gazetteers through data mining [12,19,20,22,34], we outline four generic tasks as follows:

- *To seek for potential resources*: it refers to collect the initial raw data according to the objectives of mining information.
- *To clean and structure samples*: it refers to preliminarily treat the collected data to filter the irrelevant pieces, structure the data entities and/or reformat the data, etc.
- *To expose featured components*: it refers to process the samples with some methodologies toward geographic information mining objectives and assistant highlighting the features for the further process.
- *To retrieve target entities*: it refers to extract the target data results as the information mining outcome.

Under different overall geographic information mining requirements, the four generic tasks might not be strictly arranged as the above-prescribed order and might be omitted in some cases, but they could cover all potential problem scenarios and support to employ new methods.

Each of the generic tasks consists with a set of specific tasks that are designed and settled in consideration of the actual particular situations, i.e., one specific task describes one meaningful triplet containing three procedural actions organized iteratively. Between the specific tasks, the upstream one provides the temporary output to the downstream one, and the latter consumes the intermediate input and continues with propagating the effort onto the further specific task(s) if necessary. An iteration chain reflecting a serial of interrelated specific tasks is formed corresponding to each generic task.

One triplet is constituted with three instance that are configured with particular parameters, i.e., *collection*, *treatment* and *extraction* instances (we denote them as a *CTE iteration* for short). The collection instance contains the requirement specifications of dealing with initial data and is responsible for gathering the necessary data samples. Then the treatment instance proceeds with processing the collected data with cleaning, constructing, combining and/or formatting operations. Next, the extraction instance starts to extract the potential features associated with the specifications and targets brought in by the collection instance. According to the regulation of the generic tasks, one or more CTE iterations would be arranged to ensure enough effort made for answering the requirements.

4. Automatic construction of place-name datasets under GIM framework

In this section, we present a set of methodologies of solving the critical issues for automatically constructing place-name datasets and integrate them into the geographic information mining process of the GIM framework, through which we suggest an application model of the framework.

Based on the objective definition of the proposed research work stated in Section 3.1, we then proceed to consider the corresponding critical issues and identify the initial generic tasks according to the actual research scope.

4.1. Web page collection

To collect web pages that contain place information, we need to first compile a list of search queries for querying a search engine API. The search engine queries consist of three parts in this work, namely, *Street Name*, *City Name*, and *Place Type*. To prepare our queries, we first give a specific city name and then extract street names of this city from OpenStreetMap (OSM) [45,46]. OpenStreetMap is a leading VGI (Volunteered Geographic Information) project, aiming at creating a free editable map of the world. It has over 1.6 million registered users, and nearly 30% of them have made actual contributions to the maps [47,48]. We implement a technical solution to automatically download data from OSM as an XML file containing three primitive data types, namely, Node, Way, and Relation. Each of the three primitive data types is associated with a set of tags. A tag is basically a pair (key, value) that complements the information corresponding to the primitive data types. Fig. 2 gives a snippet of the OSM file. We extract the street names from those tags whose keys equals to "*addr:street*" and whose values are the specific street names (such as "Main Street" and "N Sepulveda Boulevard") as shown in Fig. 2. Besides, we could also choose some alternative geospatial sources to extract street names, for example, Wikimapia.

For business types, in the previous work, we manually prepare a list of popular place types, such as Restaurant, Bar, etc. In this work, we present an algorithm to generate a set of place types automatically. The corresponding process is shown in Fig. 3. Large numbers of place types have been collected from Google Places APIs.

Given the above, the generic task of web page collection consists with two specific tasks: *place type similarity calculation* and *node similarity calculation*.

We firstly calculate the similarity of each pair of types based on the WordNet as shown in Algorithm 1 (seen in the 5th line of Algorithm 1).

$$sim(x, y) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \tag{1}$$

In Eq. (1), x and y represent two types selected from the input list, l is the shortest path length between x and y, and h is the depth of subsumer in the hierarchical semantic nets [49]. $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are the parameters which scale the contributions of shortest path length and depth, separately. For WordNet, the optimal values for the proposed parameters are: $\alpha = 0.2$ and $\beta = 0.5$, as presented by Li et al. [50]. Correspondingly, the similar types are placed in one group as shown in Fig. 4.

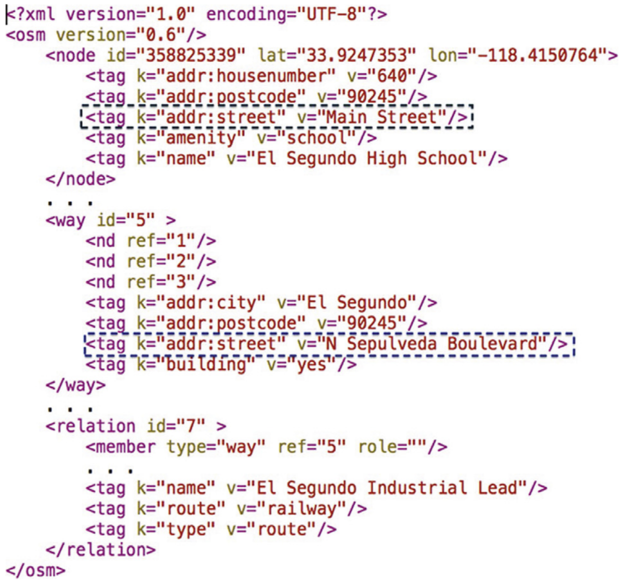
Then we calculate the sum of similarities for each node in one group. For example, compared with the other nodes in one group, the similarity sum of node 4 is the largest one. Thus, node 4, which represents its neighbors such as nodes 1,8 and 12, will be output as illustrated in Fig. 5. The other nodes such as 2, 6, 7 and 10 will be put back to the second task, i.e., node similarity calculation.

Algorithm 1: Generating a set of place types

```
    Input: allTypes[]=a collection of all types extracted from
           Google Places API
    Output: resultList[]=a proper scale data set of place types
 1  List[]  typeList=deleteNoneBusinessTypes(allTypes)
 2  // step 1: generate different similarity groups
 3  for each type x in typeList do
 4      for each type y in typeList do
 5          $sim(x, y) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$
 6          if $(sim(x, y) < threshold)$ AND $(groupList(y).has(x) \neq True)$ AND $(groupList(x).has(y) \neq True)$ then
 7              groupList(x).append(y);
 8  //step2: calculate the group center for each group g
 9  first=g[1];
10  second=g[2];
11  num=the number of types in one group;
12  switch num of do
13      case 1
14          output first;
15      case 2
16          if $frequency(first) > frequency(second)$ then
17              output first;
18          else
19              output second;
20      otherwise
21          $center = \arg\max_{v \in g} \sum_{u \in g} weight(u, v)$
22  output center;
```

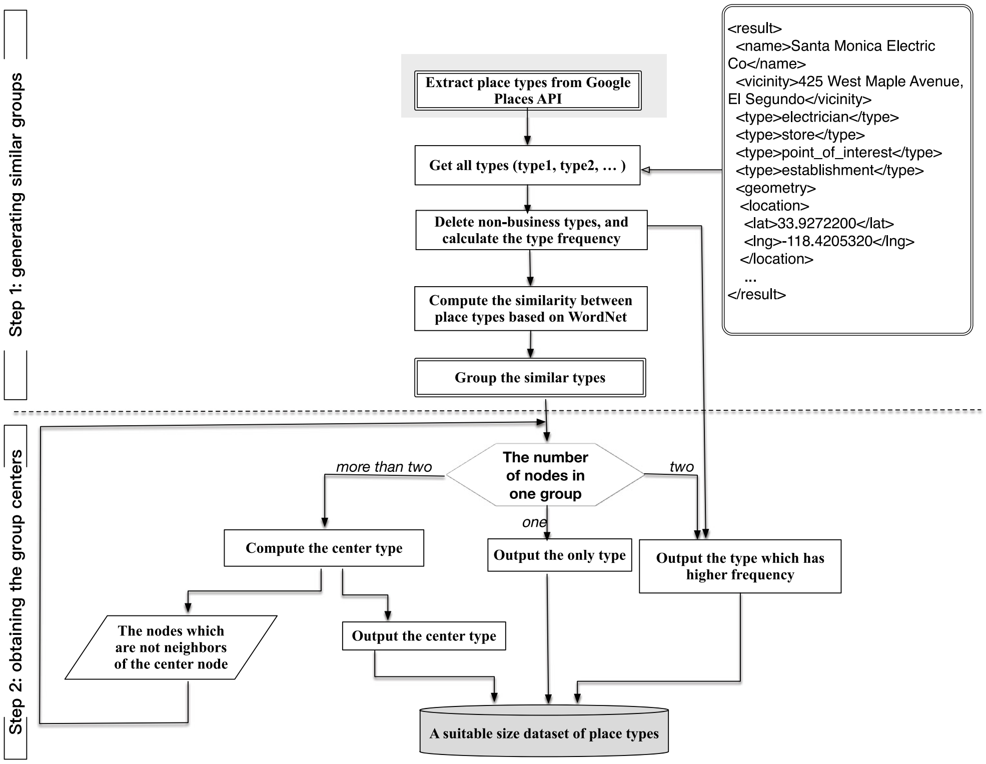
4.2. Irrelevant web page filtering

Although search engines are able to return as many results as we want, many of them are irrelevant. For this work, many



```
<?xml version="1.0" encoding="UTF-8"?>
<osm version="0.6"/>
    <node id="358825339" lat="33.9247353" lon="-118.4150764">
        <tag k="addr:housenumber" v="640"/>
        <tag k="addr:postcode" v="90245"/>
        <tag k="addr:street" v="Main Street"/>
        <tag k="amenity" v="school"/>
        <tag k="name" v="El Segundo High School"/>
    </node>
    . . .
    <way id="5" >
        <nd ref="1"/>
        <nd ref="2"/>
        <nd ref="3"/>
        <tag k="addr:city" v="El Segundo"/>
        <tag k="addr:postcode" v="90245"/>
        <tag k="addr:street" v="N Sepulveda Boulevard"/>
        <tag k="building" v="yes"/>
    </way>
    . . .
    <relation id="7" >
        <member type="way" ref="5" role=""/>
        . . .
        <tag k="name" v="El Segundo Industrial Lead"/>
        <tag k="route" v="railway"/>
        <tag k="type" v="route"/>
    </relation>
</osm>
```

Fig. 2. A snippet of OpenStreetMap.


Step 1: generating similar groups
Extract place types from Google Places API
Get all types (type1, type2, …)
Delete non-business types, and calculate the type frequency
Compute the similarity between place types based on WordNet
Group the similar types
Step 2: obtaining the group centers
The number of nodes in one group
more than two
two
one
Compute the center type
Output the only type
Output the type which has higher frequency
The nodes which are not neighbors of the center node
Output the center type
A suitable size dataset of place types

```
<result>
  <name>Santa Monica Electric
Co</name>
  <vicinity>425 West Maple Avenue,
El Segundo</vicinity>
  <type>electrician</type>
  <type>store</type>
  <type>point_of_interest</type>
  <type>establishment</type>
  <geometry>
   <location>
    <lat>33.9272200</lat>
    <lng>-118.4205320</lng>
   </location>
   ...
</result>
```

Fig. 3. The generation process of a set of place types.

real estate listings returned from search engines are big obstacles and should be coped with through executing irrelevant web page filtering task. We take advantage of machine learning to train a classifier to identify the real-estate web pages. A classifier suitable for this task should have two main characteristics. On one hand, it should not be sensitive to the ratio of the negative samples to the positive ones because the training data are often imbalanced (i.e., more positive samples than negative ones). On the other hand, the classifier should work well based on the limited training samples. Inspired by Vapnik [51], we select the Support Vector Machine (SVM), which is known to be insensitive to the size of training samples, to answer the two desired requirements. In the training set, we configure the real-estate web pages as positive samples, and the others as the negative ones. This type of problem situation can be supported by a traditional two-class-SVM. However, the two-class-SVM classifiers might be overly biased toward one class of samples when the ratio of the negative examples to the positive examples is not balanced. To cope with this issue, one-class classification solutions are introduced.

Bernhard Scholkopf in [52,53] developed an algorithm that returns a function f that takes the value +1 in a “small” field capturing most of the data points and −1 elsewhere. Thus, the strategy behind it is to map the data into the feature space

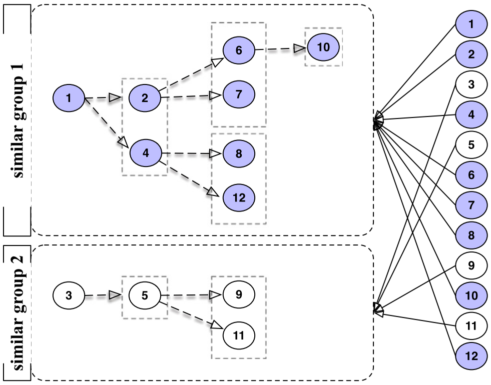


Fig. 4. Group the similar place types.

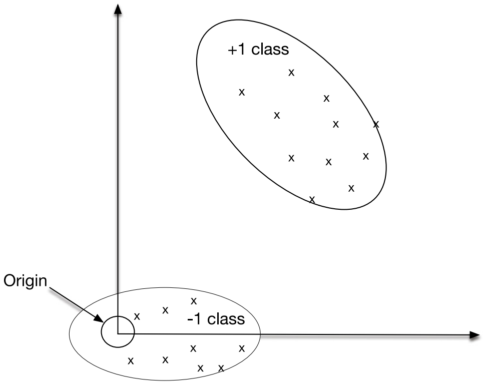


Fig. 6. One-class classification problem.

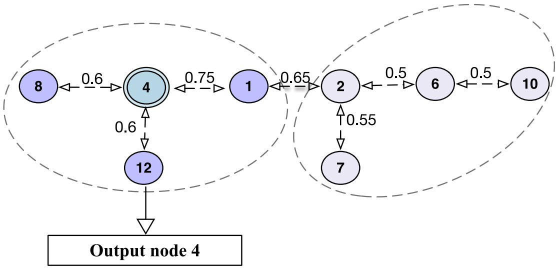


Fig. 5. Compute the group center for types.

according to the selected kernel and to separate them from the origin with the maximum margin. Users can choose different types of kernel functions which generate a variety of nonlinear estimators in input space. As shown in Fig. 6, the algorithm tries to separate dataset from the origin with maximum margin, and the corresponding problem is denoted as below:

$$\min_{\omega\in F,\varepsilon\in R^t,\rho\in R}\frac{1}{2}\|\omega\|^2+\frac{1}{vl}\sum_i \xi_i-\rho \tag{2}$$

subject to

$$(\omega\cdot\Phi(x_i))\geq\rho-\xi_i,\xi_i\geq 0 \tag{3}$$

$v \in (0,1)$ is a parameter representing the percentage of the negative samples within the training dataset. Nonzero slack variables ξ_i are penalized in the objective function. According to the dual problem presented by expression (4) and (5), we can get the decision functions shown in formula (6) and (7). In this work, we select Radial Basis Function (RBF, described by formula (8)) as the kernel function because of its good performance to be discussed in Section 5.

$$\min_{\alpha}\frac{1}{2}\sum_{ij}\alpha_i\alpha_j k(x_i,x_j)\quad subject\quad to\quad 0\leq\alpha_i\leq\frac{1}{vl},\sum_i\alpha_i=1 \tag{4}$$

$$k(x,y)=(\Phi(x),\Phi(y)) \tag{5}$$

$$f(x)=\mathrm{sgn}(\sum_i\alpha_i k(x_i,x)-\rho) \tag{6}$$

$$\rho=(\omega\cdot\Phi(x_i))=\sum_j\alpha_j k(x_j,x_i) \tag{7}$$

$$K(x_i,x_j)=\exp(-\gamma\left\|x_i-x_j\right\|^2),\gamma>0 \tag{8}$$

Given the SVM model, we then need to extract the features for real-estate websites. For this work, we collect the real-estate URLs and extract their common features. We pick up seven features from the common attributes: (1) the number of strings composing a URL; (2) the ordinal number of the string containing the first number in the given URL; (3) the position (i.e., the ordinal number) of the first number character in that string; (4) whether the string containing the first number is the longest one; (5) the position (i.e., the ordinal number) of "www"; and (6) the position (i.e., the ordinal number) of "com". For instance, the corresponding characteristic vector corresponding to an instance URL "https://www.zillow.com/homedetails/16080-Running-Deer-Trl-Poway-CA-92064/16828157_zpid/" is expressed as <7, 6, 1, 1, 2, 4>. We use "//", "/", and "." to separate the URL string, and take "-" as the separator between words. The procedures of webpage filtering module are shown in Fig. 7.

4.3. Geographic feature exposure

Place name and place address are two features of geographic information during extracting place-name entities for building a gazetteer. However, extracting both of them is a challenge because of the variety of web page structures. A lot of related research work use the statistic methods based on the vast corpus to disambiguate geographic features. Others adopt predefined rules to filter out place names. In contrast, we take advantage of the titles and abstracts generated by Google (demonstrated in Fig. 8) to extract place names.

Therefore, we identify and plan two specific tasks for exposing geographic features as *place addresses retrieve* and *place name disambiguation*.

During the execution of the place addresses retrieve task, some challenges affect the performance of address extraction from web pages. For instance, one or more elements in the address may be missing. It may also have some extra text to indicate routing information [54]. The wide variation in abbreviation will also affect the performance of extraction. In order to address such problems, we present a new method shown in Algorithm 2.

Since we focus on American addresses in this work, we propose a common pattern for American address styles, i.e., "*number +(direction)+street name+street type+city name+state name+ country name*". According to Algorithm 2, we get the position of city name in the webpage content at first, and then set the city name as the center of strings, and extract its following words to obtain the state name and country name, and retrieve its

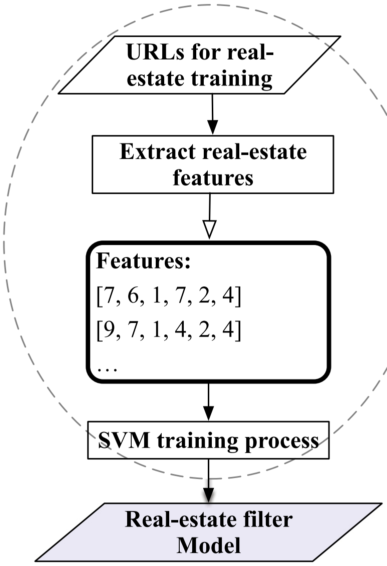
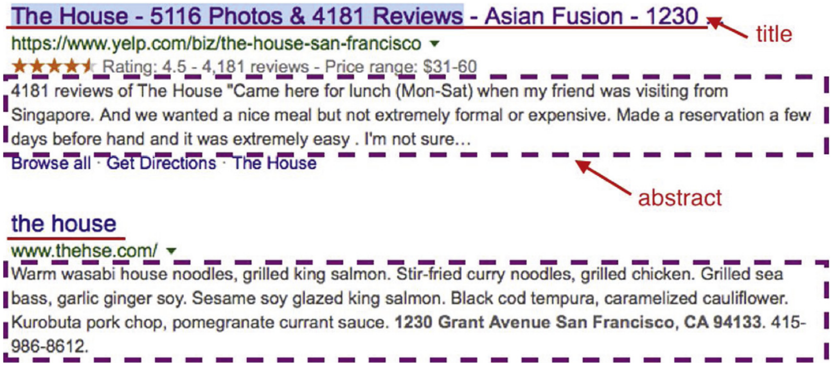


Fig. 7. Real-estate filter.


The House - 5116 Photos & 4181 Reviews - Asian Fusion - 1230
https://www.yelp.com/biz/the-house-san-francisco
★★★★★ Rating: 4.5 - 4,181 reviews - Price range: $31-60
4181 reviews of The House "Came here for lunch (Mon-Sat) when my friend was visiting from Singapore. And we wanted a nice meal but not extremely formal or expensive. Made a reservation a few days before hand and it was extremely easy . I'm not sure...
Browse all · Get Directions · The House

the house
www.thehse.com/
Warm wasabi house noodles, grilled king salmon. Stir-fried curry noodles, grilled chicken. Grilled sea bass, garlic ginger soy. Sesame soy glazed king salmon. Black cod tempura, caramelized cauliflower. Kurobuta pork chop, pomegranate currant sauce. **1230 Grant Avenue San Francisco, CA 94133.** 415-986-8612.

Fig. 8. Titles and abstracts generated by Google.

previous words to check whether the previous words are the street type, street name, direction or numbers based on the given address pattern. The street dictionary is critical for the algorithm, as shown in Fig. 9, which includes two parts, namely, Street Types and Direction. Street Types consist of various types such as "*street*", "*avenue*", and "*drive*" with their corresponding abbreviations "*st*", "*ave*", and "*dr*". Direction is comprised of "*north*", "*south*", "*west*", "*east*" and their abbreviations. For instance, in the street dictionary, one record written as "*'Rice Boulevard':'direction':None, 'type':2*" means that for street *'Rice Blvd'*, there is no direction information and the street type is the second type in the type list. As mentioned in Section 4.1, the street names are retrieved from OpenStreetMap on the condition that a specific city name is given in advance. Thus, the relationships among city name, state name and country name can be obtained. Correspondingly, we can identify them immediately by looking up the state name dictionary and country name dictionary by Algorithm 2 when the state name and the country name are missing in the web content. In addition, we use two flexible variables denoted as m and n in the algorithm to solve the problem concerning the extra information in address texts.

During the execution of place name disambiguation, a large number of researchers took advantage of gazetteer to identify place names. There is a common weakness of these methods that rely on the gazetteers is that the gazetteers are highly incomplete. Some other researchers used machine learning for toponym resolution. Due to the limited availability of annotated corpora, such approaches cannot avoid the instinct problems as that the supervised machine learning method and the semi-supervised machine learning method have.

To reduce the dependence on both the gazetteers and the annotated corpora, we employ search engines to extract titles and abstracts that are used to resolve place names as Fig. 10 depicts.

The previously extracted place addresses are used as the input data for disambiguating the place names. Given one specific address, we utilize search engines to retrieve those titles including the building numbers and the street names as the same as the search keyword has. Then we extract substrings which start from the beginning up to the building numbers as the potential place names from the extracted titles. To check whether these potential place names are real place names, we take both the corresponding addresses and the potential place names as the keywords during the next round of searching process. Meanwhile, this method can also address the problem that one place is mapping to the various place names. In the second searching process, we extract both abstracts and URLs associated with the web page titles. URLs are utilized to filter out the real-estate websites as mentioned in Section 4.2. The common substrings from the web page titles will be extracted when the searching address is contained in either titles or their corresponding abstracts. The common substring which gets the highest similarity score is taken as a place name.

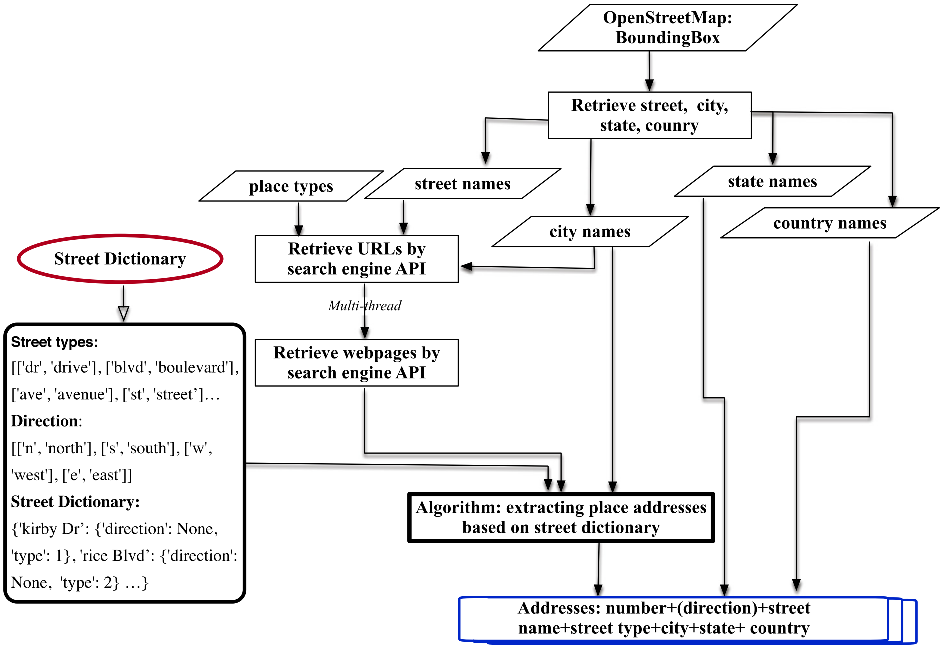

OpenStreetMap: BoundingBox
Retrieve street, city, state, counry
place types
street names
state names
city names
country names
Street Dictionary
Retrieve URLs by search engine API
Multi-thread
Retrieve webpages by search engine API
Street types:
[['dr', 'drive'], ['blvd', 'boulevard'], ['ave', 'avenue'], ['st', 'street']…
Direction:
[['n', 'north'], ['s', 'south'], ['w', 'west'], ['e', 'east']]
Street Dictionary:
{'kirby Dr': {'direction': None, 'type': 1}, 'rice Blvd': {'direction': None, 'type': 2} …}
Algorithm: extracting place addresses based on street dictionary
Addresses: number+(direction)+street name+street type+city+state+ country

Fig. 9. Extraction of address data from webpages.

4.4. Place-name entity extraction

With the identified geographic features through the previous tasks, the definition of place name is given by formula (9) and (10). *count(com_str)* denotes the number of common string com_str among the titles, while *len(com_str)* represents the length of *com_str*. ω_1 and ω_2 are two weights which represent the impact factor of the count number and the length separately. Such a two-layer searching process finally produce the place names with the greatly improved precision of place name disambiguation.

$$place\_name = \underset{\substack{com\_str(i)\\ i\in\{i,\ldots,k\}}}{\arg} \max(score(com\_str(i))) \tag{9}$$

$$\begin{cases} score(com\_str(1)) = \omega_1 \times count(com\_str(1)) \\ \qquad + \omega_2 \times len(com\_str(1)) \\ \ldots \\ score(com\_str(k)) = \omega_1 \times count(com\_str(k)) \\ \qquad + \omega_2 \times len(com\_str(k)) \end{cases} \tag{10}$$

4.5. Visualizing presentation

In this paper, we obtain the place-name entities through the one-round iterative processing phase and then visualize the entities by displaying the extracted addresses within the geographic coordinates to deploy the achievement according to the GIM framework.

To visualize places, we use a geocoding tool (e.g., Google Fusion Tables) to convert the extracted address information into geographic coordinates, and then place the markers on a map. Google Fusion Tables is an experimental data visualization web application to gather, visualize, and share data tables. After uploading the generated dataset to the Fusion Tables, it can auto-detect the location data in the table and illustrate them on a map. We can also customize which data appears and how it is displayed. As illustrated in Fig. 11, we customized three kinds of pins. The green pins represent the place names with geographic information obtained by the proposed approach. The blue pins

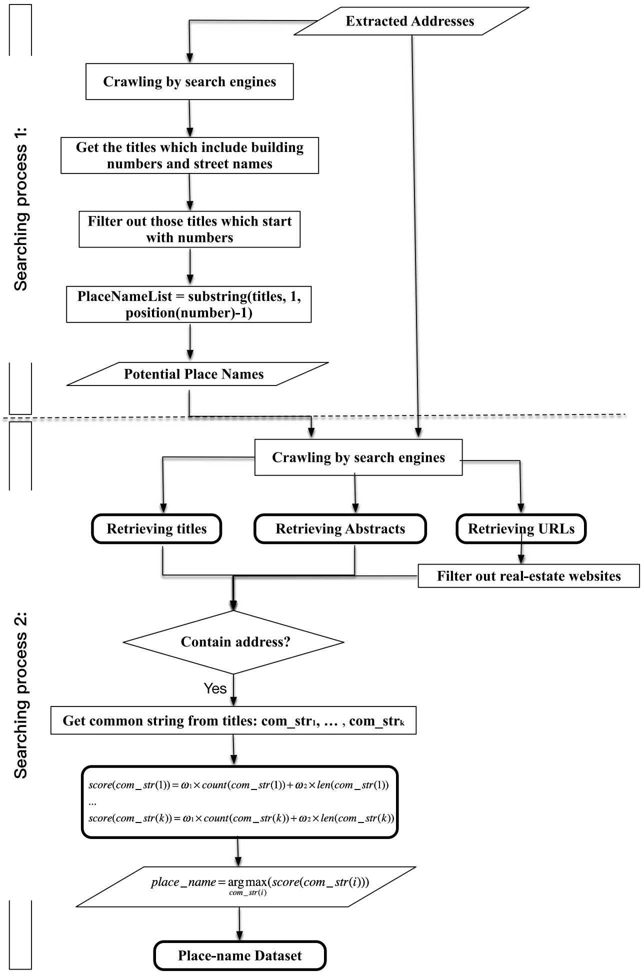

Extracted Addresses
Searching process 1:
Crawling by search engines
Get the titles which include building numbers and street names
Filter out those titles which start with numbers
PlaceNameList = substring(titles, 1, position(number)-1)
Potential Place Names
Searching process 2:
Crawling by search engines
Retrieving titles
Retrieving Abstracts
Retrieving URLs
Filter out real-estate websites
Contain address?
Yes
Get common string from titles: com_str1, … , com_strk
score(com_str(1)) = ω1×count(com_str(1)) + ω2×len(com_str(1))
…
score(com_str(k)) = ω1×count(com_str(k)) + ω2×len(com_str(k))
place_name = argmax(score(com_str(i))) com_str(i)
Place-name Dataset

Fig. 10. Procedures of place name disambiguation.

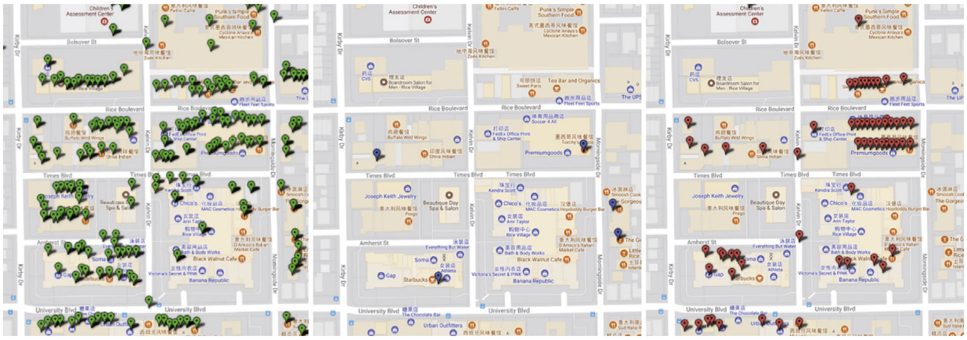


Fig. 11. Geocoding and visualizing the extracted place-name dataset in maps (The green pins represent the place names with geographic information obtained by our approach. The blue pins denote those places extracted from OpenStreetMap, and the red pins describe the place information retrieved from Wikimapia) . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

denote those places extracted from OpenStreetMap, while the red pins describe the place information retrieved from Wikimapia.

Algorithm 2: Extracting Place Address

```
Input:
Street Dictionary[] sdList; pages[]= A collection of web pages;
Street Types[] stList; Street Direction[] sdirList;
State Names[] snList; Country Names[] cnList;
Output: Address[] address;
contentPages[]=filterHtmlTags(pages);
//find the position of city name in the webpage content
location=position(city_name);
address.addCity(city name);
//judge whether the following word is a state name
if string(location+1,1) is in snList then
    address.addState(string(location+1,1));
else
    Look up dictionary for state name;
    address.addState(state name);
//judge whether the following words contain country name
if string(location+m,m+1) is in cnList then
    address.addCountry(string(location+m,m+1));
else
    Look up dictionary for country name;
    address.addCountry(country name);
//judge whether the previous word is a street type
if string(location-1,1) is in stList then
    location =location-1;
    address.addType(string(location-1, 1));
    //judge whether the previous n words is a street name
    if string(location-n, n) is in snList then
        location=location-n;
        address.addName(string(location-n, n));
        //judge whether the previous word is a street direction
        if string(location-1, 1) is in sdirList then
            location=location-1;
            address.addDirection(string(location-1, 1));
    //judge whether the previous word is a house number
    if string(location-1, 1) is a digital number then
        address.addNumber(string(location-1, 1));
return address;
```

Table 1
The comparison between the previous work and the proposed contribution.

Issue	The previous work	The proposed contribution
To compose searching keywords	Manually providing place types	Suggesting the WordNet-based similarity computing method
To filter irrelevant webpages	Querying Google search engine using recently sold home address	Employing SVM to train real-estate filtering model
To retrieve place addresses	Parsing the place address as a line starting with a number and containing the city name as an address (line length $<$ threshold)	Proposing a new pattern for place name recognition
To extract place names	Assuming the webpage title was the place name	Providing a two-layer searching process

In Table 1, we summarize the whole research work with comparing the previous work and the current contribution under the GIM framework.

In Section 4, we depict the particular process of adopting the GIM framework to automatically construct place-name datasets by proposing the methods for solving the critical problems during geographic information mining. We thus derive an application model based on the GIM framework (cf. Fig. 12).

5. Verification and experiments

In this section, we are going to evaluate whether the GIM framework integrated with the methods would meet the requirements and specifications brought in by the issue of enriching place-name datasets from mining geographic information from Internet. To answer the question "*Are we building the things right?*" [55] frequently posed during verifications, we structure the verification paradigm by following the process proposed by Sargent [56] but slightly modifying it to fit our work in this paper (seen in Fig. 13).

In the paradigm, the *problem entity* refers to the target phenomena that we are going to analyze and model. In this paper, the phenomenon is that there is plenty of data reflecting the potential geographic features and available information within Internet. The *conceptual model*, on one hand, brings in our modeling achievement of representing the problem entity and, on other hand, presents our reflection onto organizing the suite of methodologies for mining geographic information from web

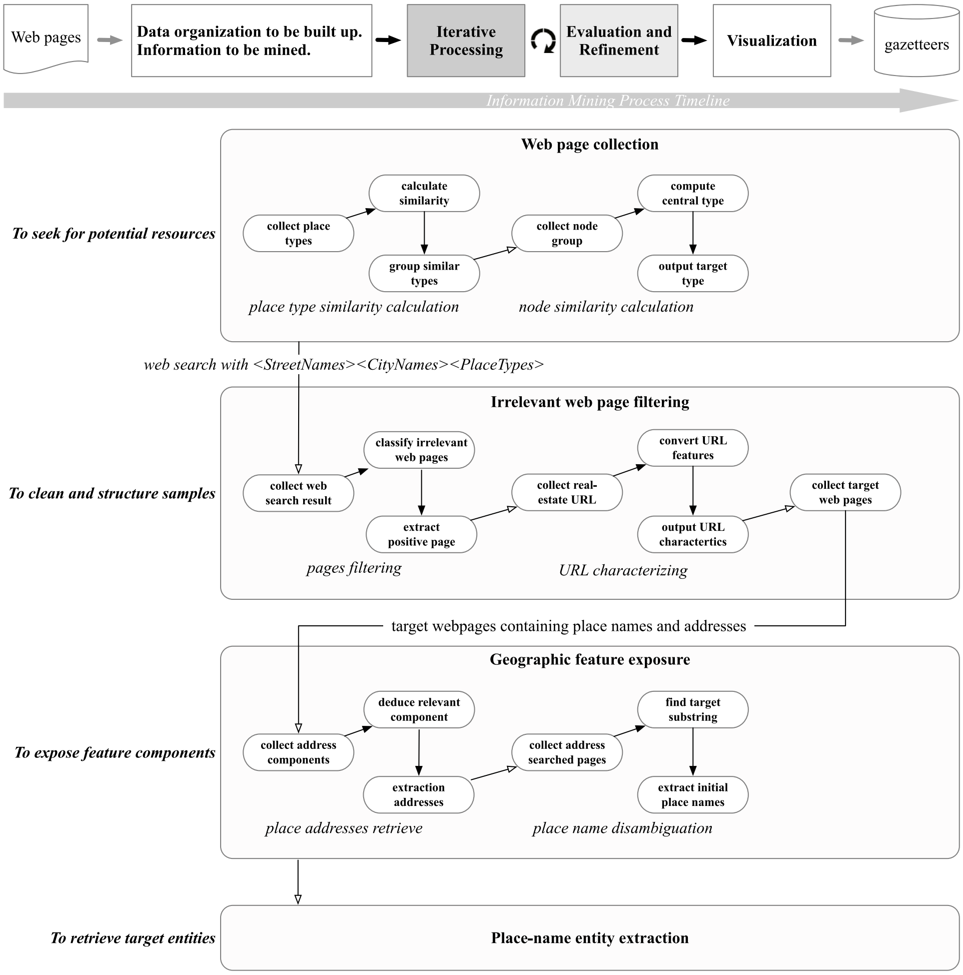

Web pages
Data organization to be built up. Information to be mined.
Iterative Processing
Evaluation and Refinement
Visualization
gazetteers
Information Mining Process Timeline
Web page collection
To seek for potential resources
collect place types
calculate similarity
group similar types
collect node group
compute central type
output target type
place type similarity calculation
node similarity calculation
web search with <StreetNames><CityNames><PlaceTypes>
Irrelevant web page filtering
To clean and structure samples
collect web search result
classify irrelevant web pages
extract positive page
collect real-estate URL
convert URL features
output URL charactertics
collect target web pages
pages filtering
URL characterizing
target webpages containing place names and addresses
Geographic feature exposure
To expose feature components
collect address components
deduce relevant component
extraction addresses
collect address searched pages
find target substring
extract initial place names
place addresses retrieve
place name disambiguation
To retrieve target entities
Place-name entity extraction

Fig. 12. Application model of GIM framework.

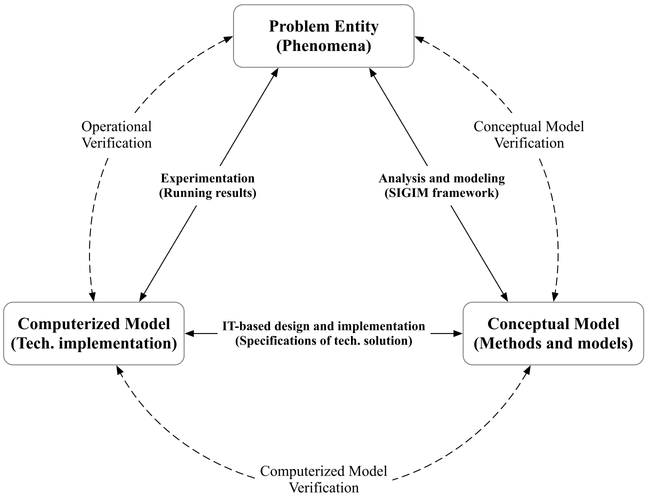

Problem Entity (Phenomena)
Operational Verification
Conceptual Model Verification
Experimentation (Running results)
Analysis and modeling (SIGIM framework)
Computerized Model (Tech. implementation)
IT-based design and implementation (Specifications of tech. solution)
Conceptual Model (Methods and models)
Computerized Model Verification

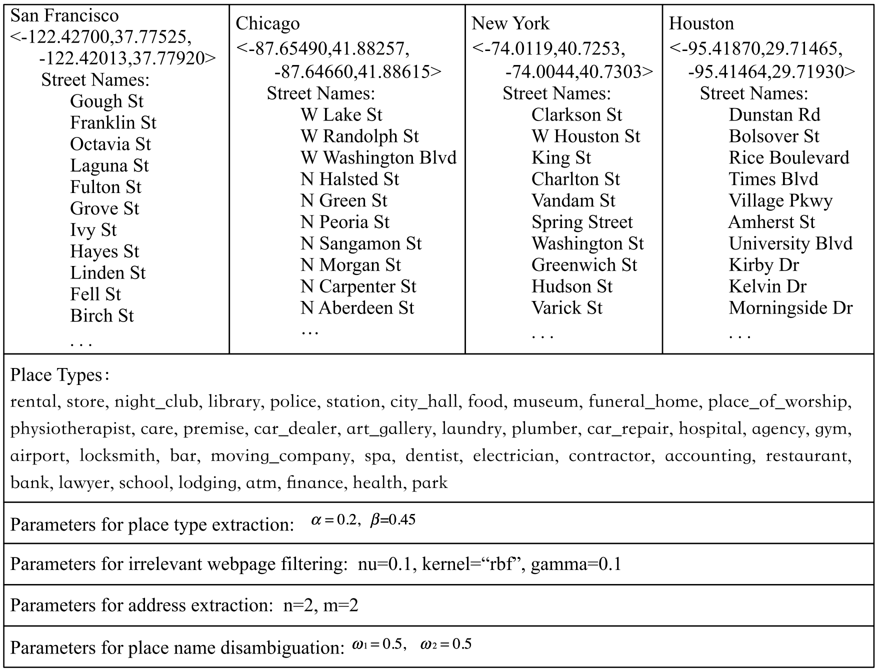
Fig. 13. Verification paradigm.
Source: Adapted from [56]

pages. The *computerized model* is the technical implementation designed and developed based on the specifications of methods and models from the conceptual model.

Through the *conceptual model verification*, we justify whether the methods and models are correct when reflecting the problem entity. In this paper, we analyze the characteristics of web pages as well as the feasibility of dealing with the web pages with employing search engines and then ponder upon organizing a serial processes of mining geographic information associated with search engines. To verify the framework, we re-explore our previous work in [12] and align its proposed solutions to the geographic information mining process of the GIM framework to prove the correctness and feasibility of the conceptual model.

Through the *computerized model verification*, we justify whether the specifications derived from the conceptual model could be correctly covered with the technical implementation. In this paper, we identify a set of key issues as the requirements of constructing the place-name datasets. Relying on the previous work in [12], we design the algorithms and compose the executable methods as the preliminary solutions to the computerized model. The specific details of the solutions would be depicted in the following sections (i.e., Sections 5.1, 5.2 and 5.3).

Through the *operational verification*, we justify whether the running results according to the solutions from the computerized model could respond to treating the problem entity. With putting the parameters of the selected areas (cf. Section 5.1) in the solutions, we compare the output with the other existed datasets. The details would also be stated by the experiments made in the following sections (i.e., Sections 5.1, 5.2 and 5.3).



San Francisco	Chicago	New York	Houston
<-122.42700,37.77525, -122.42013,37.77920> Street Names: Gough St Franklin St Octavia St Laguna St Fulton St Grove St Ivy St Hayes St Linden St Fell St Birch St . . .	<-87.65490,41.88257, -87.64660,41.88615> Street Names: W Lake St W Randolph St W Washington Blvd N Halsted St N Green St N Peoria St N Sangamon St N Morgan St N Carpenter St N Aberdeen St …	<-74.0119,40.7253, -74.0044,40.7303> Street Names: Clarkson St W Houston St King St Charlton St Vandam St Spring Street Washington St Greenwich St Hudson St Varick St . . .	<-95.41870,29.71465, -95.41464,29.71930> Street Names: Dunstan Rd Bolsover St Rice Boulevard Times Blvd Village Pkwy Amherst St University Blvd Kirby Dr Kelvin Dr Morningside Dr . . .

Place Types:
rental, store, night_club, library, police, station, city_hall, food, museum, funeral_home, place_of_worship, physiotherapist, care, premise, car_dealer, art_gallery, laundry, plumber, car_repair, hospital, agency, gym, airport, locksmith, bar, moving_company, spa, dentist, electrician, contractor, accounting, restaurant, bank, lawyer, school, lodging, atm, finance, health, park

Parameters for place type extraction: $\alpha = 0.2$, β=0.45

Parameters for irrelevant webpage filtering: nu=0.1, kernel="rbf", gamma=0.1

Parameters for address extraction: n=2, m=2

Parameters for place name disambiguation: $\omega_1 = 0.5$, $\omega_2 = 0.5$

Fig. 14. Experimental setting.

Table 2
The comparison among kernel functions.

	Precision	Recall	F-Measure
Linear	0.674502712477	0.941919191919	0.786090621707
Poly	0.659574468085	0.939393939394	0.775
RBF	0.959349593496	0.893939393939	0.925490196078
Sigmoid	0.714867617108	0.886363636364	0.791431792559

5.1. Experimental setting

In this section and the following ones (i.e., Sections 5.1, 5.2 and 5.3), we concentrate upon specifying our experimental setup and demonstrating the evaluation results from employing the critical methods under the GIM framework, i.e., the computerized model and operational verifications.

Referring to the adopted one-class-SVM for filtering out irrelevant web pages (cf. Section 4.2), we first give the experimental results to elaborate why we select RBF as a kernel function in this work. Following that, we then illustrate the good performance of the proposed web page filtering solution. Finally, we test the three datasets (i.e., our built up dataset and the two datasets respectively provided by OpenStreetMap and Wikimapia) in terms of *precision*, *recall*, and *F-score* [57,58].

During evaluating the contribution, we selected four areas of the United States to apply the proposed methods. Fig. 14 specifies the experimental settings such as the range of corresponding areas, city names, street types and place types, which are used to query the Google Search Engine. The street names were obtained from OpenStreetMap automatically as discussed in Section 4.1. In this work, all the extracted place types were merged to 40 representative types as shown in Fig. 14. In addition, the specific meanings of parameters for each stage listed below have been discussed in the corresponding sections.

5.2. Evaluation measurement

In this paper, three measures were used to evaluate the performances of the proposed contribution. They are *precision*, *recall* and *F-measure*. We used Eq. (11) to evaluate the precision of the generated place name datasets. It is defined as the ratio of the number of correctly extracted places to the total number of extracted places. *TP* denotes the number of true positives, *TN* represents the number of true negatives, *FP* is the number of false positives, and *FN* means the number of false negatives.

$$P = \frac{TP}{TP + FP} \times 100\% \tag{11}$$

Recall was used to evaluate the missing places through the proposed process (cf. Section 3.1) according to Eq. (12). It is defined as the ratio of the number of correctly extracted places to the total number of places in the ground truth.

$$R = \frac{TP}{TP + FN} \times 100\% \tag{12}$$

We evaluated the harmonic mean of P and R, i.e., F-score, which was calculated by Eq. (13). F-score have a high value only when both P and R have high values and can be seen as a way to find the best compromise between P and R.

$$F_1 = 2 \times (\frac{P \times R}{P + R}) = \frac{2TP}{2TP + FP + FN} \times 100\% \tag{13}$$

5.3. Performance evaluation

5.3.1. Evaluation for real-estate website filtering

To evaluate the performance of the information mining process of the GIM framework, we first test the function of the irrelevant web page filtering task (cf. Section 4.2), whose performance take a significant effect on the GIM framework. In this module, we selected RBF as a kernel function for one-class-SVM because the experimental results illustrated that RBF kernel function got the best results as shown in Table 2. Parameter "*gamma*" is usually set to the ratio of 1 to the number of feature types. In this paper, $gamma = \frac{1}{thenumberoffeaturetypes} = \frac{1}{7} \approx 0.10$. In contrast, "*nu*" is a parameter representing the proportion of negative examples to the whole training set. Actually, "*nu*" is the same as variable "*v*" in formula (1). In this test, "*nu*" is set to 0.1. The training set includes 3710 samples and the testing set

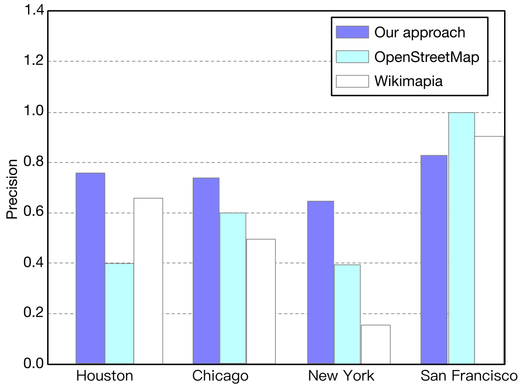


Fig. 15. An average precision comparison among our approach, OSM and Wikimapia.

contains 824 samples. Correspondingly, the precision of webpage filtering module of our method is 0.959, the recall is equal to 0.894, and the value of F-measure reaches 0.925.

5.3.2. Comparison between the explored result, OSM and wikimapia

In the experiments, we selected Google Map as the ground truth because of its high quality. As shown in Fig. 14, we tested four city areas with selecting 10 blocks to evaluate the performance of the methods and models.

Tables 3, 4, 5, 6 present the overall results for all test blocks of four cities. We compared our obtained results in this paper with OpenStreetMap and Wikimapia regarding recall, precision and F-score (Figs. 15, 16, 17). Fig. 15 shows an average precision of each area. We can see that the obtained precision was higher than that of OpenStreetMap and Wikimapia. Compared with a single source, the Internet is much wider although it has various complicated structures. Thus, the methods of our contribution can retrieve large numbers of places. Even compared with Google Maps, our contribution obtained more places as illustrated in Tables 3, 4, 5, 6.

The places listed in Table 7 were extracted by our proposed method from Internet, while they are missing on current Google Maps. For instance, more than one places share the same address “2524 Rice Blvd Houston TX US”, namely, Cookie Bouquet, Cookies by Design, and Maui Brothers Pizza. However, we can get only one place “Pizza L’Vino” from Google Maps. Since we could not walk there to check how many places which shared the same address, we considered Google Maps as ground truth in this work. Therefore, the extra places extracted from Internet were simply taken as negative samples. Correspondingly, the value of TP (True Positive) declined, and the value of FP (False Positive) raised. Based on the definition of precision given by Eq. (11), we can find that this is the main reason why the average precisions of our contribution did not reach as high as 0.9. Since OpenStreetMap and Wikimapia are developed by volunteers, precision of them should be high. However, for many areas, most of places are missed in OpenStreetMap and Wikimapia. Some blocks in the testing areas have no marked places. So, the average precision of OSM and Wikimapia for the testing area is relatively low.

Fig. 16 illustrates that the recall of our contribution was much higher than that of OpenStreetMap and Wikimapia. The number of extracted place names is larger than that of Google Maps. However, since we use Google Maps as the ground truth, the samples that were missed on the current Google Maps were considered as negative ones.

Fig. 17 illustrates that our contribution outperformed OpenStreetMap and Wikimapia in F-score. However, large numbers of the extracted places include more false positives, which prevent our contribution gaining a higher F-score. The false-positives in our contribution could be from two main reasons. First, compared with Google Maps, our methods of the GIM framework extracted more places, so such extra samples were taken as false positives in our work. Thus, the real F-score of our contribution could be higher than test value. Second, the main idea in our contribution for extracting place names was to retrieve the common substrings from titles when each part of searching address is contained in either titles or their corresponding abstracts. Most of the addresses satisfied this rule, but there were some exceptions. For instance, given the query “417 Spring St New York NY US”, our methods incorrectly extracted the place name “First Niagara Bank” from the title since the union of both title and abstract contained every part of the search query. However, the real place address for “First Niagara Bank” is “417 Spring Street Jamestown, NY” as shown in Fig. 18. So, we can see that the performance including precision, recall and F-score of New York city area are not as good as that of other areas. We would address the first problem by manually walking to the selected areas in order to check the correctness of the extra place information. In order to overcome the second problem, we would add additional conditions such as whether or not the street name in the searching query belong to the extracted city.

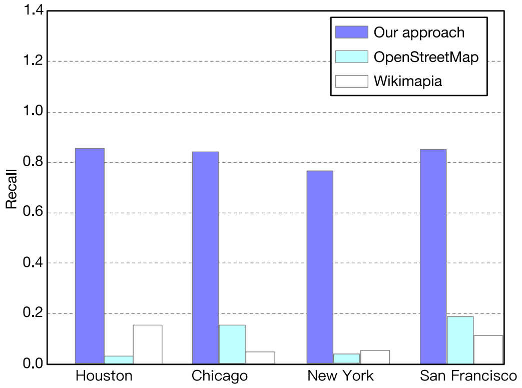


Fig. 16. An average recall comparison among our approach, OSM and wikimapia.

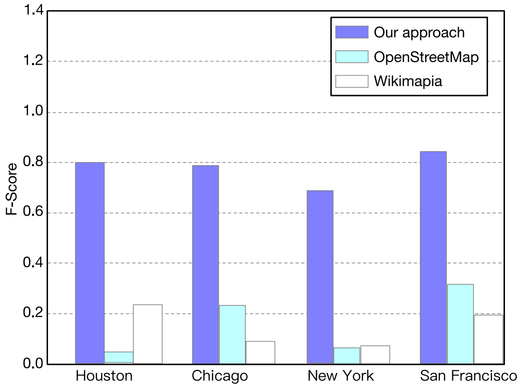


Fig. 17. An average F-score comparison among our approach, OSM and wikimapia.

6. Conclusion

In this paper, we presented *a Geographic Information Mining (GIM) framework for building place-name datasets from Internet to enrich gazetteers with local place names* used in normal life. This work concludes and extends our previous work in [12]. The new work in this paper were comprised of the following parts:

Table 3
The overall results for all the test blocks at Houston.

Precision			Recall			F-score		
Pro-App.[a]	OSM	Wikimapia	Pro-App.	OSM	Wikimapia	Pro-App.	OSM	Wikimapia
73%	0	100%	93%	0	3%	82%	0	6%
81%	0	82%	86%	0	41%	83%	0	55%
80%	0	0	100%	0	0	89%	0	0
88%	100%	67%	82%	6%	12%	85%	11%	20%
83%	100%	73%	83%	3%	23%	83%	6%	35%
60%	0	0	86%	0	0	71%	0	0
59%	0	71%	83%	0	22%	69%	0	34%
60%	100%	100%	79%	11%	5%	68%	20%	10%
86%	100%	100%	96%	4%	40%	91%	8%	57%
91%	0	60%	69%	0	10%	78%	0	17%

[a] It refers to the proposed approach in this paper.

Table 4
The overall results for all the test blocks at Chicago.

Precision			Recall			F-score		
Pro-App.[a]	OSM	Wikimapia	Pro-App.	OSM	Wikimapia	Pro-App.	OSM	Wikimapia
60%	0	0	100%	0	0	75%	0	0
69%	0	0	92%	0	0	79%	0	0
80%	100%	100%	100%	13%	13%	89%	23%	23%
72%	100%	100%	82%	27%	5%	77%	43%	10%
83%	0	0	79%	0	0	81%	0	0
85%	100%	100%	88%	24%	12%	86%	39%	21%
71%	100%	100%	68%	23%	5%	69%	37%	10%
60%	0	0	82%	0	0	69%	0	0
71%	100%	0	63%	50%	0	67%	67%	0
93%	100%	100%	88%	13%	13%	90%	23%	23%

[a] It refers to the proposed approach in this paper.

Table 5
The overall results for all the test blocks at New York.

Precision			Recall			F-score		
Pro-App.[a]	OSM	Wikimapia	Pro-App.	OSM	Wikimapia	Pro-App.	OSM	Wikimapia
67%	0	0	100%	0	0	80%	0	0
86%	0	0	60%	0	0	71%	0	0
90%	100%	14%	79%	4%	4%	84%	8%	6%
63%	100%	0	63%	13%	0	63%	23%	0
48%	0	0	77%	0	0	59%	0	0
63%	0	0	71%	0	0	67%	0	0
50%	0	50%	100%	0	11%	67%	0	18%
70%	100%	67%	88%	13%	25%	78%	23%	36%
54%	100%	13%	64%	3%	4%	57%	5.6%	7%
57%	0	14%	65%	5%	5%	59%	6%	6%

[a] It refers to the proposed approach in this paper.

Table 6
The overall results for all the test blocks at San Francisco.

Precision			Recall			F-score		
Pro-App.[a]	OSM	Wikimapia	Pro-App.	OSM	Wikimapia	Pro-App.	OSM	Wikimapia
71%	100%	100%	79%	16%	16%	75%	28%	28%
79%	100%	100%	92%	17%	25%	85%	29%	40%
100%	100%	100%	82%	18%	9%	90%	31%	17%
75%	100%	100%	86%	14%	14%	80%	25%	25%
78%	100%	100%	89%	21%	16%	83%	35%	28%
93%	100%	100%	84%	22%	3%	88%	36%	6%
71%	100%	100%	81%	33%	9%	76%	50%	17%
100%	100%	100%	71%	14%	14%	83%	25%	25%
82%	100%	0	100%	22%	0	90%	36%	0
83%	100%	100%	96%	12%	4%	89%	21%	8%

[a] It refers to the proposed approach in this paper.

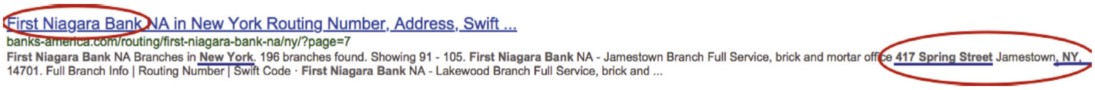

First Niagara Bank NA in New York Routing Number, Address, Swift ...
banks-america.com/routing/first-niagara-bank-na/ny/?page=7
First Niagara Bank NA Branches in New York. 196 branches found. Showing 91 - 105. First Niagara Bank NA - Jamestown Branch Full Service, brick and mortar office 417 Spring Street Jamestown, NY, 14701. Full Branch Info | Routing Number | Swift Code · First Niagara Bank NA - Lakewood Branch Full Service, brick and ...

Fig. 18. A snippet of false sample generated by our approach.

Table 7
The missing places of Google Maps in Houston.

Place name	Place address	Google maps	Internet
Cookie Bouquet	2524 Rice Blvd Houston TX US	0	1
Cookies by Design	2524 Rice Blvd Houston TX US	0	1
Maui Brothers Pizza	2524 Rice Blvd Houston TX US	0	1
Texadelphia	2520 Rice Blvd Houston TX US	0	1
Box&Box Custom Jewelers	2514 Rice Blvd Houston TX US	0	1
Persnickety Houston TX	2504 Rice Blvd Houston TX US	0	1
Merle norman cosmetic studio	2513 Rice Blvd Houston TX US	0	1

(1) We create a geographic information mining process under the GIM framework that provides the multiple modeling dimensions of processing potential geographic information to resolve the critical issues in building up place-name datasets. Especially, the process organizes the activities in a set of iteration chains under the multiple modeling dimensions to maintain the flexibility and adaptability of the GIM framework.

(2) We presented a WordNet-based similarity computing method to merge the extracted place types with Google Places APIs in order to automatically generate a set of place types on a proper scale. One the one hand, such a proper size of place-type collection brought much more complete searching results. On the other hand, they did not decrease the searching efficiency.

(3) We took advantage of machine learning (i.e., one-class SVM) to train a classifier in order to identify the real-estate web pages (contain mostly residential addresses). The experimental results illustrated that the filtering process got a relatively high precision (0.959) and F-measure (0.925).

(4) We presented a new algorithm to detect the existence of place locations from natural language text and to disambiguate them, i.e., geoparsing to make the place location disambiguation.

(5) In order to refrain from heavily depending on the annotated datasets, we seek the possibility and feasibility of learning the necessary information from the extracted initial web pages to disambiguate place names.

The experimental results showed that our contribution outperformed OpenStreetMap and Wikimapia in terms of precision, recall and F-score.

Our current and future work in this area focuses on three main directions. The first is to improve both precision and recall of the geographic feature extraction module. This can be done by exploiting the DOM structure of a web page. Besides, we can also take advantage of the ontology concepts to define a flexible way to establish semantically richer relationships between place types. The richer relationships between place types can provide better search queries to search engine and thus improve the recall. The second is further analyzing and processing the retrieved web pages to extract much more useful knowledge such as the historical information of places. In addition, we need to explore new methods to parse various address patterns for different countries. The presented method can be applied to different areas in the world when the corresponding address parsers are built. Lastly, we will focus on further analyzing and processing the retrieved web pages to extract much more useful knowledge such as the historical information of places.

Acknowledgment

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2018MS024 in part by the National Natural Science Foundation of China under Grant 61305056, and in part by the Overseas Expertise Introduction Program for Disciplines Innovation in Universities (Project 111) under Grant B13009.

References

[1] M.F. Goodchild, L.L. Hill, Introduction to digital gazetteer research, Int. J. Geogr. Inf. Sci. 22 (10) (2008) 1039–1044.
[2] L.L. Hill, Core elements of digital gazetteers: placenames, categories, and footprints, in: International Conference on Theory and Practice of Digital Libraries, Springer, Berlin, Heidelberg, 2000, pp. 280–290.
[3] Hu Yingjie, Geospatial semantics, in: Bo Huang, Thomas J. Cova, Ming-Hsiang Tsou, et al. (Eds.), Comprehensive Geographic Information Systems, Elsevier, Oxford, UK, 2017.
[4] C. Davies, I. Holt, J. Green, J. Harding, L. Diamond, User needs and implications for modelling vague named places, Spatial Cognition Comput. 9 (3) (2009) 174–194.
[5] L. Hollenstein, R. Purves, Exploring place through user-generated content: using flickr tags to describe city cores, J. Spatial Inf. Sci. 2010 (1) (2010) 21–48.
[6] M. Lapata, F. Keller, The web as a baseline: Evaluating the performance of unsupervised web-based models for a range of NLP tasks, in: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004, 2004.
[7] K. Stock, C. Cialone, Universality, Language-variability and individuality: defining linguistic building blocks for spatial relations, in: International Conference on Spatial Information Theory, Springer, Berlin, Heidelberg, 2011, pp. 391–412.
[8] F. Twaroch, R. Purves, C. Jones, Stability of qualitative spatial relations between vernacular regions mined from web data, in: Proceedings of Workshop on Geographic Information on the Internet, Toulouse, France, 2009.
[9] R. Wirth, J. Hipp, CRISP-DM: TOwards a standard process model for data mining, in: Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining, Citeseer, 2000, pp. 29–39.
[10] G. Mariscal, O. Marban, C. Fernandez, A survey of data mining and knowledge discovery process models and methodologies, Knowl. Eng. Rev. 25 (2) (2010) 137–166.
[11] W. Li, M.F. Goodchild, R.L. Church, B. Zhou, Geospatial data mining on the web: Discovering locations of emergency service facilities, in: International Conference on Advanced Data Mining and Applications, Springer, Berlin, Heidelberg, 2012, pp. 552–563.
[12] Ying Zhang, Yao-yi Chiang, A Craig Knoblock, Chaopeng Li, Liming Du, Shaowen Liu, Sanja Singh, An automatic approach for building place-name datasets from the web, in: 19th AGILE Conference on Geographic Information Science, 2016, pp. 1–6.
[13] Mateos Carlos Javier Broncano, Carlos Pinilla Ruiz, Rubén González Crespo, Andrés Gaspar Castillo Sanz, Relative radiometric normalization of multitemporal images, Int. J. Interact. Multimedia Artif. Intell. 1 (3) (2010) 54–59.
[14] Taibi Aissa, Baghdad Atmani, Combining fuzzy AHP with GIS and decision rules for industrial site selection, Int. J. Interact. Multimedia Artif. Intell. 4 (6) (2010) 60–69.
[15] Huang Ming, Jiajun Lin, Yong Peng, Xing Xie, Design a batched information retrieval system based on a concept-lattice-like structure, Knowl.-Based Syst. 150 (2018) 74–84.
[16] G. Lamprianidis, D. Skoutas, G. Papatheodorou, D. Pfoser, Extraction, integration and analysis of crowdsourced points of interest from multiple web sources, in: ACM Sigspatial International Workshop on Crowdsourced and Volunteered Geographic Information, Vol. 5, ACM, 2014, pp. 16–23.
[17] A. Blessing, H. Schütze, Automatic acquisition of vernacular places, in: Proceedings of the 10th International Conference on Information Integration and Web-based Applications and Services, ACM, 2008, pp. 662–665.
[18] J. Gelernter, G. Ganesh, H. Krishnakumar, W. Zhang, Automatic gazetteer enrichment with user-geocoded data, in: Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, ACM, 2013, pp. 87–94.
[19] C. Keßler, K. Janowicz, M. Bishr, An agenda for the next generation gazetteer: geographic information contribution and retrieval, in: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances In Geographic Information Systems, ACM, 2009, pp. 91–100.

[20] C.B. Jones, R.S. Purves, P.D. Clough, H. Joho, Modelling vague places with knowledge from the web, Int. J. Geogr. Inf. Sci. 22 (10) (2008) 1045–1065.

[21] R. Purves, P. Clough, H. Joho, Identifying imprecise regions for geographic information retrieval using the web, in: Proceedings of the 13th Annual GIS Research UK Conference, 2005, pp. 313–318.

[22] S. Schockaert, M. De Cock, Neighborhood restrictions in geographic IR, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 167–174.

[23] O. Uryupina, Semi-supervised learning of geographical gazetteers from the internet, in: Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References-Volume 1, Association for Computational Linguistics, 2003, pp. 18–25.

[24] P. Brindley, J. Goulding, M.L. Wilson, A data driven approach to mapping urban neighbourhoods, in: Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2014, pp. 437–440.

[25] A. Popescu, G. Grefenstette, P.A. Moëllic, Gazetiki: automatic creation of a geographical gazetteer, in: Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital Libraries, ACM, 2008, pp. 85–93.

[26] W. Lorenzo, Rubén González Crespo, A. Castillo, A prototype for linear features generalization, Int. J. Interact. Multimedia Artif. Intell. 1 (3) (2010) 59–65.

[27] F. Wang, W. Wu, Z. Li, M. Zhou, Named entity disambiguation for questions in community question answering, Knowl.-Based Syst. 126 (C) (2017) 68–77.

[28] C. Hung, S.J. Chen, Word sense disambiguation based sentiment lexicons for sentiment classification, Knowl.-Based Syst. 110 (2016) 224–232.

[29] Y. Gutiérrez, S. Vázquez, A. Montoyo, Spreading semantic information by word sense disambiguation, Knowl.-Based Syst. 132 (2017).

[30] R. Volz, J. Kleb, W. Mueller, Towards ontology-based disambiguation of geographical identifiers, in: CEUR Workshop Proceedings, 2007.

[31] Xu Haijiao, Changqin Huang, Dianhui Wang, Enhancing semantic image retrieval with limited labeled examples via deep learning, Knowl.-Based Syst. 163 (2019) 252–266.

[32] Zhou Yu, Jianbin Huang, He Li, Heli Sun, Yan Peng, Yueshen Xu, A semantic-rich similarity measure in heterogeneous information networks, Knowl.-Based Syst. 154 (2018) 32–42.

[33] J.L. Leidner, Toponym resolution in text: annotation, evaluation and applications of spatial grounding, in: ACM SIGIR Forum, Vol. 41, No. 2, ACM, 2007, pp. 124–126.

[34] M. Speriosu, J. Baldridge, Text-driven toponym resolution using indirect supervision, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2013, pp. 1466–1476.

[35] S.E. Overell, S. Rüger, Geographic co-occurrence as a tool for gir, in: Proceedings of the 4th ACM Workshop on Geographical Information Retrieval, ACM, 2007, pp. 71–76.

[36] S.E. Overell, S.M. Rüger, Identifying and grounding descriptions of places, in: ACM Workshop on Geographic Information Retrieval, Gir 2006, Seattle, Wa, USA, 2008.

[37] R. Yangarber, W. Lin, R. Grishman, Unsupervised learning of generalized names, in: Proceedings of the 19th International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2002, pp. 1–7.

[38] M. Collins, Y. Singer, Unsupervised models for named entity classification, in: 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[39] S. Cucerzan, D. Yarowsky, Language independent named entity recognition combining morphological and contextual evidence, in: 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, 1999.

[40] R. Yangarber, R. Grishman, P. Tapanainen, S. Huttunen, Automatic acquisition of domain knowledge for information extraction, in: Proceedings of the 18th Conference on Computational linguistics-Volume 2, Association for Computational Linguistics, 2000, pp. 940–946.

[41] G. DeLozier, J. Baldridge, L. London, Gazetteer-independent toponym resolution using geographic word profiles, in: AAAI, 2015, pp. 2382–2388.

[42] Y.L. Zaila, D. Montesi, Geographic information extraction, disambiguation and ranking techniques, in: Proceedings of the 9th Workshop on Geographic Information Retrieval, ACM, 2015, p. 11.

[43] O. Maimon, L. Rokach, Decomposition methodology for knowledge discovery and data mining, in: Data Mining and Knowledge Discovery Handbook, Springer, Boston, MA, 2005, pp. 981–1003.

[44] E.E. Vityaev, B.Y. Kovalerchuk, Relational methodology for data mining and knowledge discovery, Intell. Data Anal. 12 (2) (2008) 189–210.

[45] L. Alarabi, A. Eldawy, R. Alghamdi, M.F. Mokbel, TAREEG: a MapReduce-based web service for extracting spatial data from OpenStreetMap, in: Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ACM, 2014, pp. 897–900.

[46] A. Ballatore, M. Bertolotto, D.C. Wilson, Geographic knowledge extraction and semantic similarity in OpenStreetMap, Knowl. Inf. Syst. 37 (1) (2013) 61–81.

[47] P. Neis, A. Zipf, Analyzing the contributor activity of a volunteered geographic information project—The case of OpenStreetMap, ISPRS Int. J. Geo-Inf. 1 (2) (2012) 146–165.

[48] M. Haklay, How good is volunteered geographical information? a comparative study of openstreetmap and ordnance survey datasets, Environ. Plann. B 37 (4) (2010) 682–703.

[49] Y. Li, D. McLean, Z.A. Bandar, K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Trans. Knowl. Data Eng. (8) (2006) 1138–1150.

[50] Y. Li, Z.A. Bandar, D. McLean, An approach for measuring semantic similarity between words using multiple information sources, IEEE Trans. Knowl. Data Eng. 15 (4) (2003) 871–882.

[51] V. Vapnik, The Nature of Statistical Learning Theory, Springer science and business media, 2013.

[52] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471.

[53] B. Schölkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, J.C. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, 2000, pp. 582–588.

[54] Z. Yu, High accuracy postal address extraction from web pages, in: Masters Abstracts International, Vol. 45, No. 05, 2007.

[55] W.L. Oberkampf, C.J. Roy, Verification and Validation in Scientific Computing, Cambridge University Press, 2010.

[56] R.G. Sargent, Verification and validation: verification and validation of simulation models, J. Simul. 7 (1) (2010) 12–24.

[57] M. Andrea Rodriguez, M.J. Egenhofer, Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure, Int. J. Geogr. Inf. Sci. 18 (3) (2004) 229–256.

[58] I. Muslea, S. Minton, C.A. Knoblock, Active learning with strong and weak views: A case study on wrapper induction, in: IJCAI, Vol. 3, 2003, pp. 415–420.