Geospatial Data Mining on the Web: Discovering Locations of Emergency Service Facilities

Wenwen Li[1], Michael F. Goodchild[2], Richard L. Church[2], and Bin Zhou[3]

[1] GeoDa Center for Geospatial Analysis and Computation, School of Geographical Sciences and Urban Planning, Arizona State University, Tempe AZ 85287
Wenwen@asu.edu
[2] Department of Geography, University of California, Santa Barbara
Santa Barbara, CA 93106
{good,church}@geog.ucsb.edu
[3] Institute of Oceanographic Instrumentation, Shandong Academy of Sciences
Qingdao, Shandong, China 266001
senosy@gmail.com

Abstract. Identifying location-based information from the WWW, such as street addresses of emergency service facilities, has become increasingly popular. However, current Web-mining tools such as Google's crawler are designed to index webpages on the Internet instead of considering location information with a smaller granularity as an indexable object. This always leads to low recall of the search results. In order to retrieve the location-based information on the ever-expanding Internet with almost-unstructured Web data, there is a need of an effective Web-mining mechanism that is capable of extracting desired spatial data on the right webpages within the right scope. In this paper, we report our efforts towards automated location-information retrieval by developing a knowledge-based Web mining tool, CyberMiner, that adopts (1) a geospatial taxonomy to determine the starting URLs and domains for the spatial Web mining, (2) a rule-based forward and backward screening algorithm for efficient address extraction, and (3) inductive-learning-based semantic analysis to discover patterns of street addresses of interest. The retrieval of locations of all fire stations within Los Angeles County, California is used as a case study.

Keywords: Emergency service facilities, Web data mining, information extraction, information retrieval, ontology, inductive learning, location-based services.

1 Introduction

Although it has only been 22 years since its advent, the World Wide Web (WWW) has significantly changed the way that information is shared, delivered, and discovered. Recently, a new wave of technological innovation - the emergence of Web 2.0 and Web 3.0, such as social networks, government surveillance and citizen sensors [5] - has led to an explosion of Web content, and brought us into the era of Big Data [18]. Statistical reports show that by 2008 the amount of data on the Internet had reached 500bn gigabytes [21], and the indexed Web contained at least 11.5 billion pages [6].

S. Zhou, S. Zhang, and G. Karypis (Eds.): ADMA 2012, LNAI 7713, pp. 552–563, 2012.

This information explosion on the Web poses tremendous challenges for various information-retrieval tasks [12].

Within this massive amount of data, identifying location-based information, such as street address and place name, has become very popular, due to the desire to map this information from cyberspace to the physical world. As a type of spatial data, locations of emergency service facilities are especially important in protecting people's lives and safety, and for government agencies to provide real-time emergency response. Taking fire stations as an example, besides the aforementioned functions, insurance companies need the locations of all fire stations within and near a community to determine the insurance costs to be paid by a household. Decision-makers long for this location information to obtain the urban footprint of each fire station and to plan the optimal placement of fire stations within a region.

Presently, most Internet uses obtain WWW information from search engines [16]. However, the commercial search engines such as Google are designed to index webpages on the Internet instead of considering location information that has smaller granularity as an indexable object. Therefore, these search engines always lead to a low recall rate in search results. Fig.1 shows the search results for fire stations within the city of Santa Barbara, CA, from Google. Red pinpoints are the Googled results and blue pinpoints are the actual locations of fire stations within that city. It can be seen that except for *C* (to the west of the green arrow) overlapping with its actual location (blue pinpoint), all of the results are irrelevant.

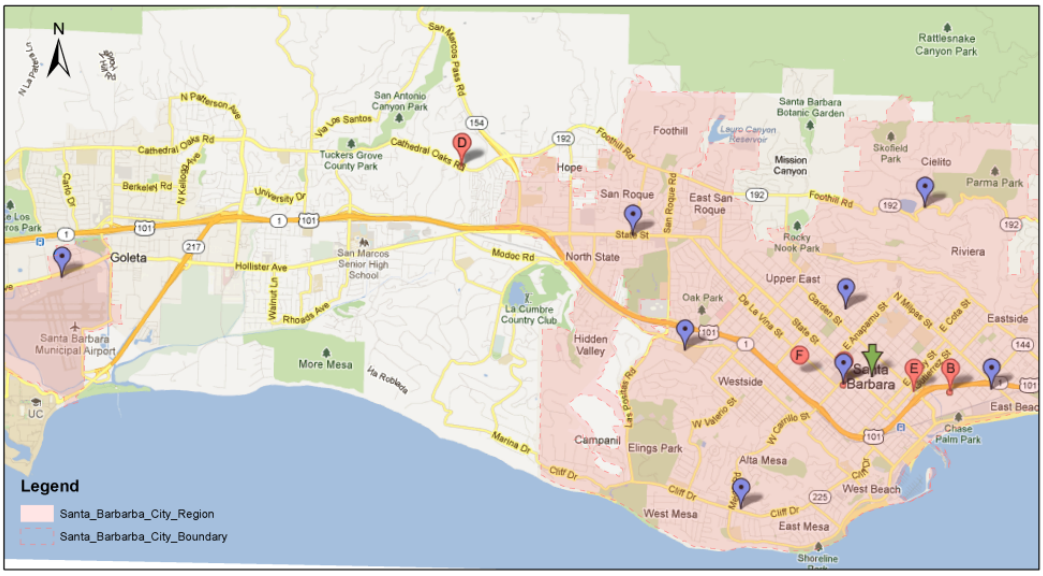

Fig. 1. Results of a Google search for locations of fire stations in the city of Santa Barbara, CA. The pink region shows the geographic extent of Santa Barbara.

In order to retrieve location-based information on the ever-expanding Internet and its almost-unstructured Web data, there is a need of an effective Web-mining mechanism that is capable of extracting desired data on the right webpages within the right scope. In this paper, we report our efforts in developing a knowledge-based Web-mining tool, CyberMiner, that adopts geospatial ontology, a forward- and backward-screening algorithm, and inductive learning for automated location information

retrieval. The retrieval of street addresses of all fire stations within Los Angeles County, California is used as a case study because the County's fire department is the one of the largest and most sophisticated fire departments in the US and its fire services are provided through different tiers of local governments [4].

2 Literature

Spatial data mining on the Web, aiming at discovering spatial data or patterns of data, is a part of the Web geographic-information retrieval (GIR) task. In GIR, studies of exploiting geographic aspects of the Web can be categorized by their purposes. One is georeferencing, a way to attach geographical scope to webpages [1]. By identifying the associations between the general content of a webpage and the location information within it, the search engines are believed to better handle location-based queries [9], such as "All Starbucks in Nanjing, China". Another category (also the focus of this paper) is to obtain location information of certain subjects from the Internet. The goal is to build up a global spatial database to support queries and decision-making.

Both of the categories require automatic extraction of location information from the source code of a webpage. Cai et al. [2] present a method combining domain ontology that defines street name, street type, and city name, and a graph matching to extract addresses from the Web. Taghva et al. [19] apply a Hidden Markov Model (HMM) to extract addresses from documents. However, its target is OCR (Optical Character Recognition) of text, such as a digital copy of a check, instead of dynamic and unstructured Web documents. Yu [23] compared a number of address-extraction methods, including rule-based approaches (regular expression and gazetteer-based approach) and machine-learning approaches (word *n*-gram model and decision-tree classifier) and determined that machine-learning approaches yield a higher recall rate. Loos and Biemann [9] proposed an approach to extracting addresses from the Internet using unsupervised tagging with a focus on German street addresses. These methods provided promising results, however they all suffered from limitations, such as heavy computation load in [2] and low recall rates in [9] and [23]. In this study, we propose an efficient rule-based method combining central keyword identification and a forward- and backward-screening algorithm for address extraction in real time.

In addition to the address extraction, there is also a need for a software agent that is able to automatically pick the right webpages to visit (those having a higher possibility of containing an address of interest) until a spatial database containing the complete information is built up. A typical tool to accomplish this task is a spatial Web crawler, which follows hyperlinks and realizes information extraction from webpages as the process continues. A previous work is [13], which developed a Web crawler to discover the distributed geospatial Web services and their distribution pattern on the Internet. In [13], the determination of whether an extracted URL is a target is straightforward, because those geospatial Web services have a specialized interface. But in our work, the addresses of the desired type have a vague signature, therefore a more intelligent analysis is needed.

In the next sections, we discuss in detail the establishment of the Web crawling framework and use the discovery of all fire stations within Los Angeles County, CA as a case study.

3 Methodological Framework

3.1 Web Crawling Process

Fig. 2 shows the state-transition graph for retrieving location data on the Web. Circles represent states, in which different types of webpages (seed webpages, interlinks, target webpages, and unrelated ones) are being processed. The transition of state is triggered by processing an outgoing link in the webpage being visited. To design an effective Web crawler, it is important to first determine which webpages can be designated as crawling seeds. The proper selection of crawling seeds is important for the whole crawling process because good seeds are always closer to the target webpages in the linked graph of the Web. Second, it is important to identify the target webpage. A target webpage is the one containing the needed location information (in our case, it is the street address for each fire station). This requires the ability to extract all possible existences of street addresses on a target webpage and the ability to filter out those not of interest. It is also important to decide which interlinks will be given higher priority to access during Web crawling. As Fig. 2 shows, a webpage referred to by an interlink may contain hyperlinks to a target webpage or another interlink. An unrelated webpage which comes from an interlink should be filtered out. In the next sections, we will discuss the solutions to these three problems.

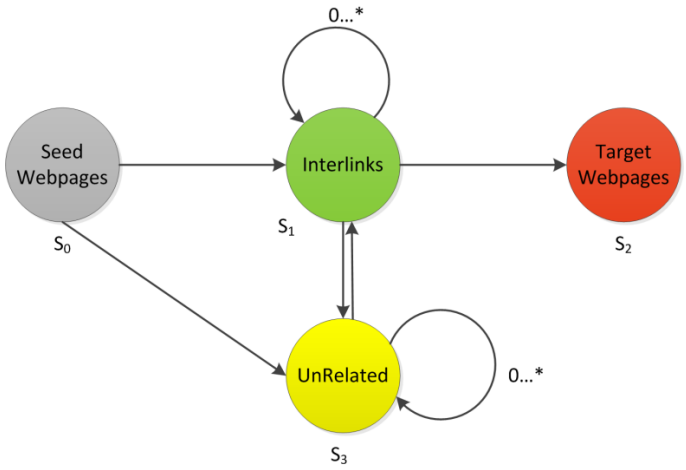

Fig. 2. State transition graph in a Web crawling process

3.2 Geospatial Taxonomy to Aid the Determination of Crawling Seeds

The crawling seeds are the URLs from where the crawling process starts. Seeds selection greatly affects crawling scope, time, and results. Bad seeds may lead to time-consuming crawling without any desired information found. As public-service facilities are mostly operated by local governments (except for a few volunteer service providers) and their locations are publicly listed on the government's website, it is necessary to obtain a knowledge base of the political divisions of the U.S. Fig. 3 is a taxonomy describing such divisions at class level. A class is a subnational entity that

forms the U.S. For example, both "County" and "Incorporated Place" are entities (classes) in the taxonomy, and an "Incorporated Place", also known as a "Municipality", is a subdivision (subclass) of a "County". For use in a crawling process, a class needs to be initiated. For example, given the question "How many fire stations exist in Los Angeles (LA) County of California?" one needs to know specifically which subdivisions of LA County have local governments.

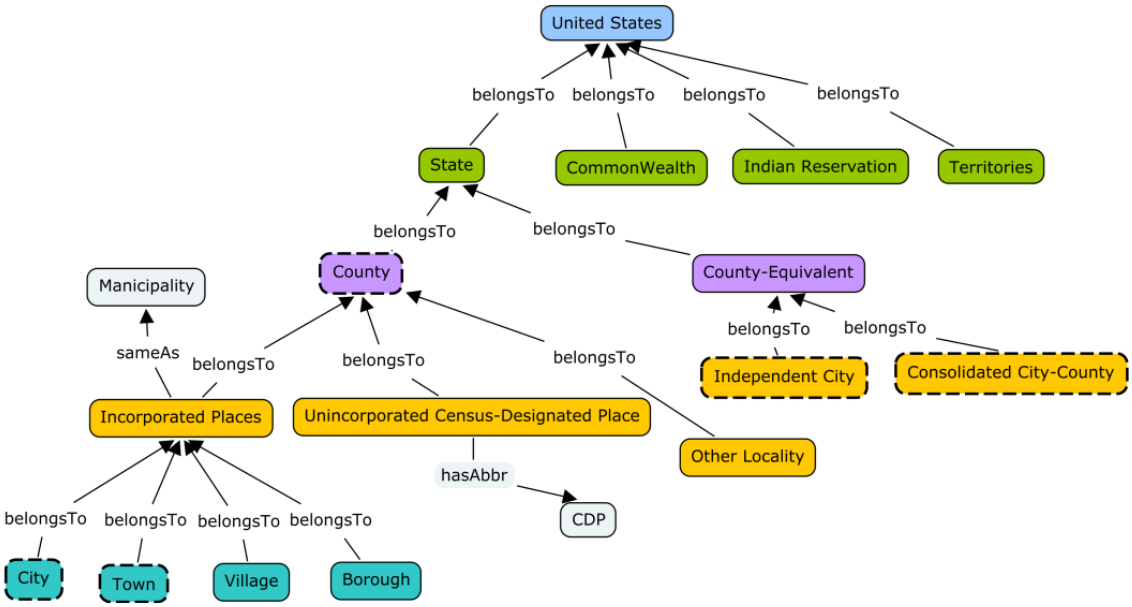

Fig. 3. Geospatial taxonomy of political division of the U.S. Nodes in different colors refer to jurisdictions at different tiers. Words on arrows show the relationship between jurisdictions.

It is worth mentioning that not all subdivisions have local governments, e.g., the CDP always have no government, and for the subdivisions that have a local government, not all of them provide a public service such as fire protection. For example, the fire protection service of Glendora City of California is provided by its parent division Los Angeles County. This would require such subdivisions to be excluded from consideration as crawling seeds. In general, any incorporated area, such as a county, city, town or other county equivalent (dotted boxes in Fig. 3), is more likely to have a local government. These nodes in the geospatial taxonomy are instantiated by the data extracted from the United States Bureau of the Census [20]. The URLs of their official websites are identified from a Google search and populated into the taxonomy. In this way, the starting points of the crawling process can be determined and the scope of search is narrowed to avoid unconcentrated crawling.

3.3 Target Webpage Identification and Street Address Extraction

Target webpages have a prominent characteristic: they contain postal street addresses for public-service facilities. The goal of identifying target webpages is to successfully extract street addresses from them. Though a street address of a public facility can be expressed in multiple ways, a standard form is shown below:

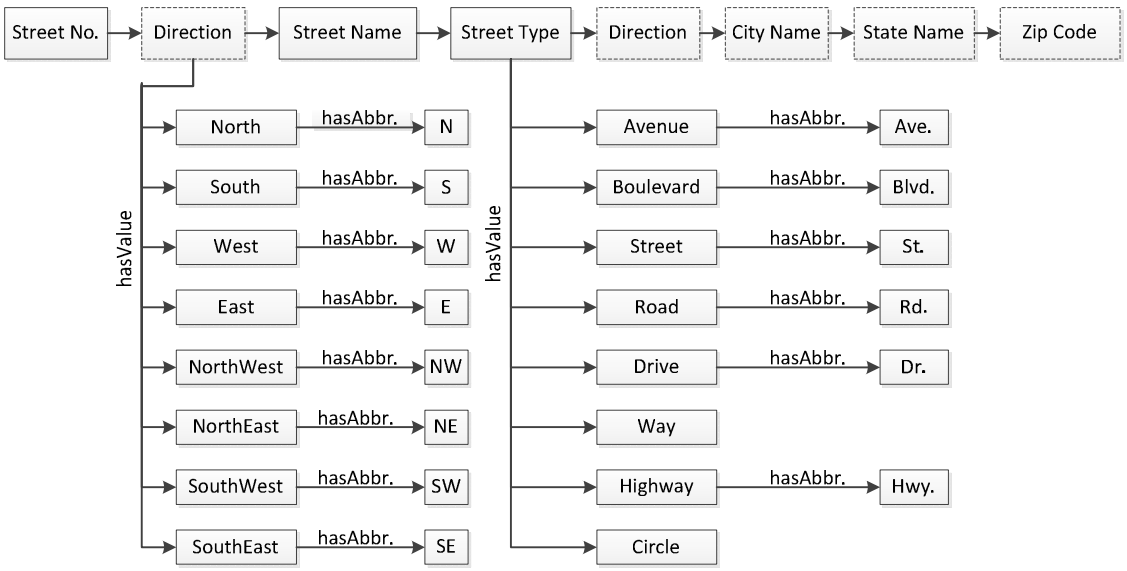

Fig. 4. Common form of street address

In practice, information for direction, city, state name, and ZIP code (boxes with dotted lines in Figure 4) are omitted in a street address. Therefore, "Street Type", such as "Avenue", "Road" and "Highway", becomes a prominent feature (we call it a "central keyword") indicating its existence. Based on a statistical analysis from the 300 known addresses of public facilities, above 90% contain "Avenue", "Boulevard", "Street", "Road", "Drive", "Way", "Highway", or "Circle", and their various abbreviations, such as "Rd" or "Rd." for "Road". Other street types such as "Plaza", "Grove", and "Place" also occur occasionally. Based on this prior knowledge, a street type dictionary was established for quick look-up.

Once the positions of the central keywords on an HTML document are found, backward-screening and forward-screening algorithms are employed to extract the complete address data. Centered on the position of street type p in the text, backward screening inspects the text block before p within a certain radius d_1, aiming at extracting a partial street address including the street number, direction, and street name based on pattern r_1. The forward-screening algorithm inspects the text block after location p within a radius d_2 to find the possible existence of city/county name, state name, and ZIP code as the other part of the address based on pattern r_2. If the extracted text block does not match the given patterns, e.g., the length of the ZIP code is not five or nine digits, even though there is an appearance of a central keyword, it will not be considered as a street address. The definitions of d and r are:

d_1: Distance between p and the location of the foremost digit in the number block closest (before) to location p.

d_2: Distance between p and the location of the last digit of the first number that appears (for detecting 5-digit ZIP code), or the last digit of the second number after p if the token distance of the first and second number block equals 2 (for detecting 9-digit ZIP code, in the format of xxxxx-xxxx).

r_1: regular expression [1-9][0-9]*[\\s\\r\\n\\t]*([a-zA-Z0-9\\.]+[\\s\\r\\n\\t])+

r_2: regular expression "*city-Pattern*"[\\s\\r\\n\\t,]?+("*statePattern*")?+[\\s\\r\\n\\t,]*\\d{5}(-\\d{4})*

Note that a number block in d_1 is defined as a whole word that consists of only numbers and is not a direct neighbor of the street type. The first restriction is to distinguish a street number, e.g., 1034 in "1034 Amelia Ave", from a street name with numbers, e.g., 54 in "W 54th Street". The second restriction is to make sure that street address with name only in numbers can be correctly extracted as well. For example, when "54th Street" is written as "54 Street", the number 54 should be recognized as the street name instead of street number. The "*cityPattern*" in r_1 is the Boolean OR expression of all cities/counties names within our study area (California in this study) and the "*statePattern*" in r_2 is the OR expression of all 50 states. In pattern r_2, we require the appearance of at least city name in the city+state+zipcode pattern for the address identification.

3.4 Semantic Analysis of Addresses

Section 3.3 describes how to extract addresses from an HTML webpage. Apparently, not all addresses are locations of public-service facilities (in our case, fire stations), even though they are extracted from the websites within the domain of city/county governments. Therefore, it is necessary to clarify that an address is truly referring to that of a fire station. We term the set of correct identifications the *positive class* of identifications; the *negative* class is the set of identified addresses that are not fire stations.

To classify an address into a positive class or a negative class, we adopt C4.5 [15], which is a widely used machine-learning algorithm based on decision-tree induction [7]. The basic strategy is to select an attribute that will best separate samples into individual classes by a measurement, 'Information Gain Ratio', based on information-theoretic 'entropy' [11]. By preparing positive and negative examples as a training set, we can produce a model to classify addresses automatically into positive and negative categories. But what semantic information should be used for constructing the training set? In this work, we assume every address on a webpage has navigational information to indicate the semantics of an address, and this navigational information is positioned in a text block right before the position of the address. For example, on the webpage http://www.ci.manhattan-beach.ca.us/Index.aspx?page=124, the fire station address "400 15th Street" has navigational information "fire stations are: Station One" right before it. On the webpage http://www.desertusa.com/nvval/, the address "29450 Valley of Fire Road, Overton, Nevada 89040" has navigational text "Valley of Fire State Park" before its position. Therefore, upon the detection of an address occurrence, we extracted its navigational text block (block size = 40 characters) and used it for semantic analysis.

We prepared four groups of 11 attributes for navigational text *T* of each address as follows:

Attributes A-D (keyword attributes): the existence of keyword "*fire station*", "*fire*", "*station*", and "*location*" within *T*. 1 means Yes, 0 means No.

Attribute E (statistical attributes): the ratio of number of keywords that exist and the total number of keywords in *T*. The value has range of (0,1].

Attribute F (pattern attributes): the existence of "*location*" followed by a number or a pound or hash symbol plus a number. 1 means Yes, 0 means No.

Attributes G-K (keyword attributes): the existence of keyword "fire station", "fire", "station", "location", or "address" in the title of each webpage.

Attributes A-D are based on the keyword frequencies of the navigational information of addresses. Attribute F is also based on the observation that most fire departments list the station number in the navigational text of a fire-station address. Attributes G-K are auxiliary attributes to complement information in the direct navigational text.

Through a semantic analysis based on the C4.5 algorithm, we analyzed the navigational text of 310 addresses, in which 191 are actually of fire stations. These addresses were crawled from the official government websites of Mesa, AZ, Columbus, OH, San Francisco, CA, and some pre-crawled cities in L.A. County. The decision rules shown in Figure 5 were extracted.

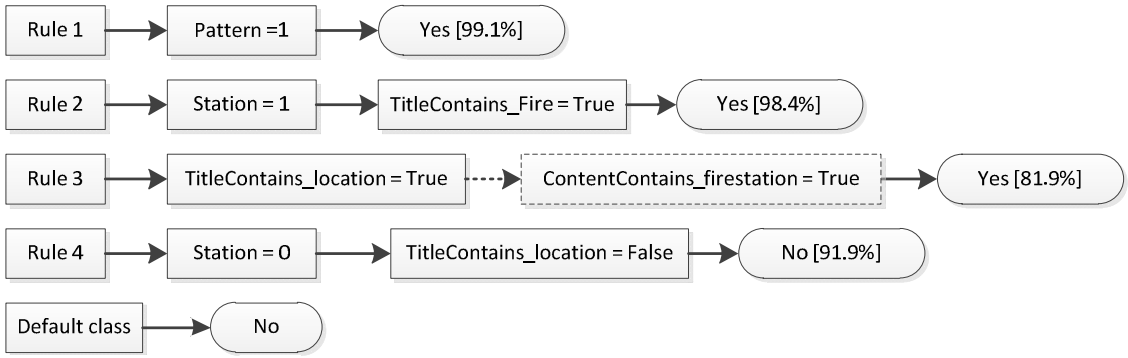

Fig. 5. Decision rules of desired addresses by training data based on semantic information

By default, an address will be placed in the negative class unless the pattern "station+number" exists in the navigational text (Rule 1) or the navigational text contains the keyword "Station" and the title of a webpage contains "fire" (Rule 2). Both Rule 1 and Rule 2 yield very high prediction accuracy, at 99% and 98% separately. Rule 3 determines the address of a fire station by recognizing the keyword "station" in the title. This rule only achieves 81.9% accuracy; therefore, to avoid false positives we added a constraint "if the content contains 'fire station' on a webpage" (in the dotted box). In practice, this rule gives improved prediction accuracy. Using these rules, we made predictions on whether an address is of interest.

4 Implementation

In the previous section we discussed three key techniques for detecting the existence of location information and to predict whether the information is of interest. In this section, we discuss the software architecture of CyberMiner (Fig. 6), which

implements the proposed techniques to enable the automatic Web-mining process. *Crawling entry* is where the crawler starts to work. The seeding Web URLs are fed into the entry and a number of initial conditions such as politeness delay and number of threads to start are configured. Politeness delay ensures that the crawler behaves politely to remote servers. The multi-threading strategy is to increase the utilization of a single CPU core, leveraging thread-level parallelism [13]. We currently set the thread number to be 4, considering the relatively small scale of crawling.

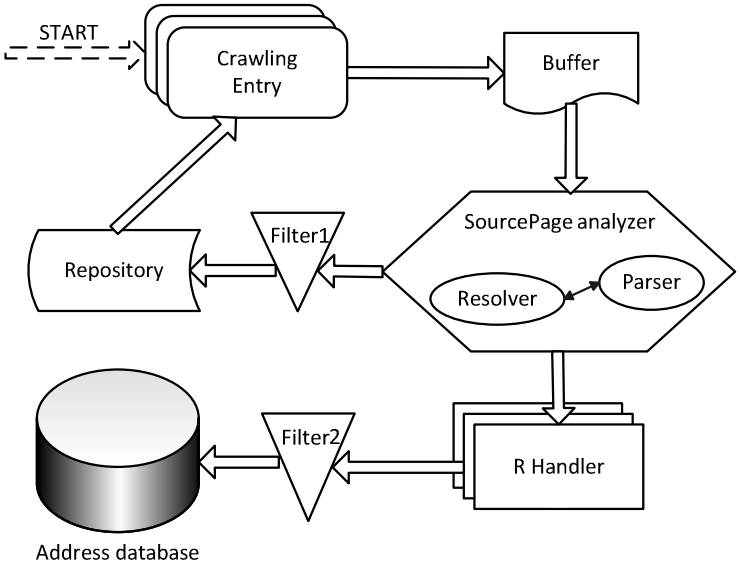

Fig. 6. Crawler Architecture (Adapted from [13])

Buffer selectively caches Web source code linked by URLs for address extraction. *Source page analyzer* is used to analyze the Web source code that has been cached in the buffer, to extract all outgoing links and convert relative URLs to absolute URLs. These URLs go to the *filter1*, which filters out the URLs that have been visited before, within another domain from its parent URL, and the webpages of those having very low possibility to contain the desired addresses, such as a URL of a .css file or an image file. This strategy guarantees the domain purification of the crawling tree inherited from one seed webpage. It also avoids unnecessary cross-domain crawling to reduce search cost.

The *repository* maintains all URLs crawled using a first-in-first-out (FIFO) queue, the head of which is always the URL to be crawled next. Every time a Webpage is being visited, the *R Handler* is initiated to extract all possible addresses from its source code and sends these addresses to *filter 2*, in which the non-desired addresses and duplicated addresses are disregarded based on semantic analysis. The target addresses are inserted into the *Address Database*. This process will continue until the FIFO queue is empty or until it reaches the maximal depth for crawling (the depth is measured by the number of steps between the current URL and its seed URL).

5 Results and Analysis

Fig. 7 shows all the 346 fire stations that have been discovered by CyberMiner within the boundary of Los Angeles County of California, US. These stations are widely distributed in 88 cities within the County, and they show a denser distribution in the cities than that in the rural region (generally the northern part of LA County). The areas without fire station coverage are mountains (the Santa Monica Mountains in the southwest of LA County and the San Gabriel Mountains in northern LA County). To evaluate the performance of CyberMiner, we measured the ratio between the number of retrieved fire locations m to the total number of relevant records on the Web n. This ratio is also known as the recall rate. Here, m equals 346. To compute n (n=392), we visited the websites of all city/county governments that have a fire department and manually annotated the street addresses of all fire stations listed on each city's government website. Although 46 stations failed to be detected, our CyberMiner still achieves very satisfying recall rate – at 88%. To share the results of this research, the data and map have been made public through a Web application http://mrpi.geog.ucsb.edu/fire, on which the locations and addresses of fire stations are provided.

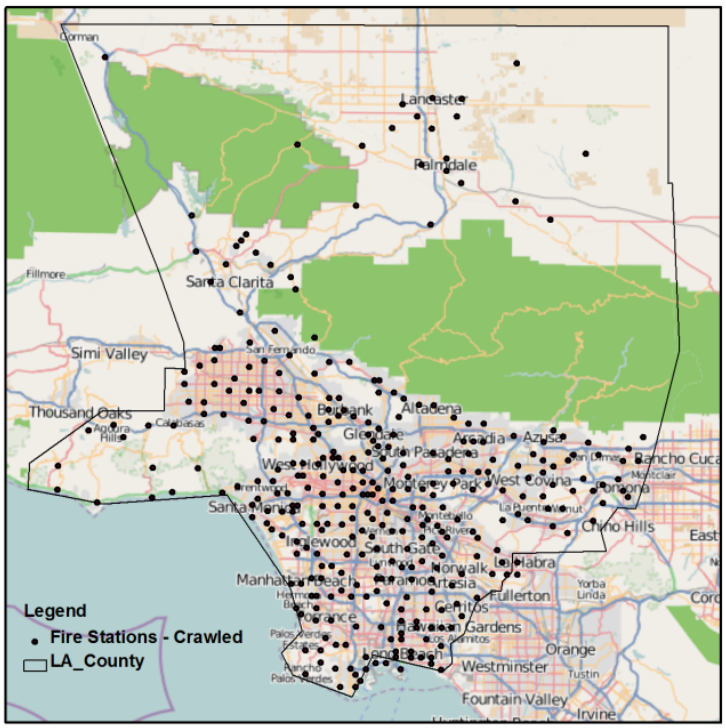

Fig. 7. Locations of all fire stations obtained by CyberMiner

6 Conclusion and Discussion

In this paper, we proposed a method and a software tool CyberMiner for automatic detection and extraction of location information of interest from the WWW. The

proposed geospatial taxonomy, containing hierarchical subdivisions of US governments, restrains the search scale to the domains that are most likely to contain the data in need, thereby greatly reducing search cost. The methods of pattern-based address extraction and inductive learning-based prediction contribute to the recall rate. Although locating fire stations is used as a case study in the paper, the proposed work is easily extendable to search for locations of other emergency/public-service facilities, such as police stations and wastewater treatment plants. The proposed algorithms, such as forward and backward screening for automatic address extraction, are beneficial to the general area of GIR. Moreover, this work goes one step further from previous address extraction research [2][9][23], in that it is not only able to extract addresses in general correctly, but it is also able to classify types of address based on the proposed semantic analysis.

In the future, we will continue to improve the performance of the CyberMiner, especially in its tolerance to errors in an address. Since the core of the proposed address detection algorithm is the determination of the central keyword – the street type - the algorithm had a hard time identifying it when it was not contained in the address type dictionary. Addressing this issue requires enriching the dictionary to include a complete list of street types in the US. Another aspect is the quality control of the search results. As our long-term goal is to establish an address database of emergency service facilities in the whole US, evaluating the correctness of these addresses would need tremendous human effort. To resolve this problem, we plan to take the advantage of the power of citizen sensors. That is, by providing a VGI (volunteered geographic information) platform, we encourage participation from the general public in providing feedback and correction of missing or mislocated information.

References

1. Buyukokkten, O., Cho, J., Garcia-Molina, H., Gravano, L., Shivakumar, N.: Exploiting geographical location information of Web pages. In: Proceedings of Workshop on Web Databases (WebDB 1999) held in Conjunction with ACM SIGMOD 1999, Philadephia, Pennsylvania, USA (1999)
2. Cai, W., Wang, S., Jiang, Q.: Address Extraction: Extraction of Location-Based Information from the Web. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) APWeb 2005. LNCS, vol. 3399, pp. 925–937. Springer, Heidelberg (2005)
3. Chang, G., Healey, M.J., McHugh, J.A.M., Wang, J.T.L.: Mining the World Wide Web, vol. 10. Kluwer Academic Publishers, Norwell (2001)
4. Glendora: City of Glendora Government Website (2012), `http://www.ci.glendora.ca.us/index.aspx?page=896` (last Access Date: July 27, 2012)
5. Goodchild, M.F.: Citizens as sensors: the world of volunteered geography. Geo Journal 69, 211–221 (2007)
6. Gulli, A., Signorini, A.: The indexable web is more than 11.5 billion pages. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, pp. 902–903. ACM, Chiba (2005)
7. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco (2001)

8. Kofahl, M., Wilde, E.: Location concepts for the web. In: King, I., Baeza-Yates, R. (eds.) Weaving Services and People on the World Wide Web, pp. 147–168. Springer, Heidelberg (2009)
9. Loos, B., Biemann, C.: Supporting web-based address extraction with unsupervised tagging. In: Data Analysis, Machine Learning and Applications 2008, pp. 577–584 (2008)
10. Li, W., Goodchild, M.F., Raskin, R.: Towards geospatial semantic search: exploiting latent semantic analysis among geospatial data. International Journal of Digital Earth (2012), doi:10.1080/17538947.2012.674561
11. Li, W., Yang, C.W., Sun, D.: Mining geophysical parameters through decision-tree analysis to determine correlation with tropical cyclone development. Computers & Geosciences 35, 309–316 (2009)
12. Li, W., Yang, C., Zhou, B.: Internet-Based Spatial Information Retrieval. In: Shekhar, S., Xiong, H. (eds.) Encyclopedia of GIS, pp. 596–599. Springer, NYC (2008)
13. Li, W., Yang, C.W., Yang, C.J.: An active crawler for discovering geospatial Web services and their distribution pattern - A case study of OGC Web Map Service. International Journal of Geographical Information Science 24, 1127–1147 (2010)
14. Ligiane, A.S., Clodoveu Jr., A.D., Karla, A.V.B., Tiago, M.D., Alberto, H.F.L.: The Role of Gazetteers in Geographic Knowledge Discovery on the Web. In: Proceedings of the Third Latin American Web Congress, p. 157. IEEE Computer Society (2005)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
16. Rogers, J.D.: GVU 9th WWW User Survey, vol. 2012 (2012), http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-1904/ (last Access Date: July 27, 2012)
17. Sanjay Kumar, M., Sourav, S.B., Wee Keong, N., Ee-Peng, L.: Research Issues in Web Data Mining. In: Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, pp. 303–312. Springer (1999)
18. Szalay, A., Gray, J.: Science in an exponential world. Nature 440, 413–414 (2006)
19. Taghva, K., Coombs, J., Pereda, R., Nartker, T.: Address extraction using hidden markov models. In: Proceedings of IS&T/SPIE 2005 Int. Symposium on Electronic Imaging Science and Technology, San Jose, California, pp. 119–126 (2005)
20. USCB: GCT-PH1 - Population, Housing Units, Area, and Density: 2010 - State - Place and (in selected states) County Subdivision (2012), http://factfinder2.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=DEC_10_SF11_GCTPH11.ST10 (last Access Date: July 27,2012)
21. Wray, R.: Internet data heads for 500bn gigabytes. The guardian, Vol. 2012. Guardian News and Media, London (2009), http://www.guardian.co.uk/business/2009/may/2018/digital-content-expansion (last Access Date: July 27,2012)
22. Yasuhiko, M., Masaki, A., Michael, E.H., Kevin, S.M.: Extracting Spatial Knowledge from the Web. In: Proceedings of the 2003 Symposium on Applications, p. 326. IEEE Computer Society (2003)
23. Yu, Z.: High accuracy postal address extraction from web pages. Thesis for Master of Computer Science. 61p. Dalhousie University, Halifax, Nova Scotia (2007)