Appl Intell (2016) 44:539–556
DOI 10.1007/s10489-015-0707-5

Enhancing POI search on maps via online address extraction and associated information segmentation

Chia-Hui Chang[1] · Hsiu-Min Chuang[1] · Chia-Yi Huang[1] · Yueng-Sheng Su[1] · Shu-Ying Li[1]

Published online: 15 October 2015

Abstract With the popularity of wireless networks and mobile devices, we have seen rapid growth in mobile applications and services, especially location-based services. However, most existing location-based services like Google Maps and Wikimapia rely on crowd-sourcing or business-data providers to maintain their points-of-interest (POI) databases, which are slow and insufficient. Because most updated information can be found on the Web, the insufficiency of current POI databases can be complemented by automatically extracting POIs and their descriptions from general webpages. In this study, we enhance location-based search on maps via online address extraction and associated information segmentation. Given a POI query that cannot be found on a map, we propose a method for extracting the address from search snippets of the query to exploit information from the Web. We demonstrate the application of sequence labeling to Chinese postal-address extraction and compare the performance with and without Chinese word segmentation. Meanwhile, we also present a novel algorithm for associated information segmentation by making use of a document-object model (DOM) tree structure based on the farthest distinguishable ancestor (FDA) of each address. The FDA algorithm is able to locate associated information for each Chinese address resulting in an improvement from an F-measure of 0.811 to 0.964.

Keywords Location-based service · Chinese postal address extraction · Associated information segmentation · Conditional random field · Record boundary detection

1 Introduction

With the popularity of wireless networks and mobile devices, an increasing number of Web application services are designed to support users in their daily life. As reported in 2012[1], "43 percent of total Google search queries are local". Indeed, the requirements for location-based information—e.g., business stores, navigation, and activities—have increased significantly over the past three years [29]. One of the key component to location-based services is the quantity of POIs (points of interest). However, existing location-based services, such as Google Maps, Garmin, Yelp, Wikimapia and OpenPOIs, are constructed manually or generated from crowd-sourcing, and these sources are limited (or focused on a single domain, e.g., restaurants) and far from sufficient.

On the other hand, the Web has become a medium for publishing information about a wide array of entities, including governments, businesses, persons and locations. The most up-to-date information can often be found on the Web. Compared to the amount of location-based information on the Web, manual tagging often centers on popular POIs, such as restaurants, hotels, and tourist attractions, leading to relatively few annotations in other categories. As stated in [1], manual annotations are not necessarily needed, because the home-pages of businesses or classified directories already contain location-related information.

✉ Chia-Hui Chang
chia@csie.ncu.edu.tw

[1] Computer Science and Information Engineering, National Central University, Taoyuan, Taiwan

[1] http://searchengineland.com/study-43-percent-of-total-google-search-queries-have-local-intent-135428

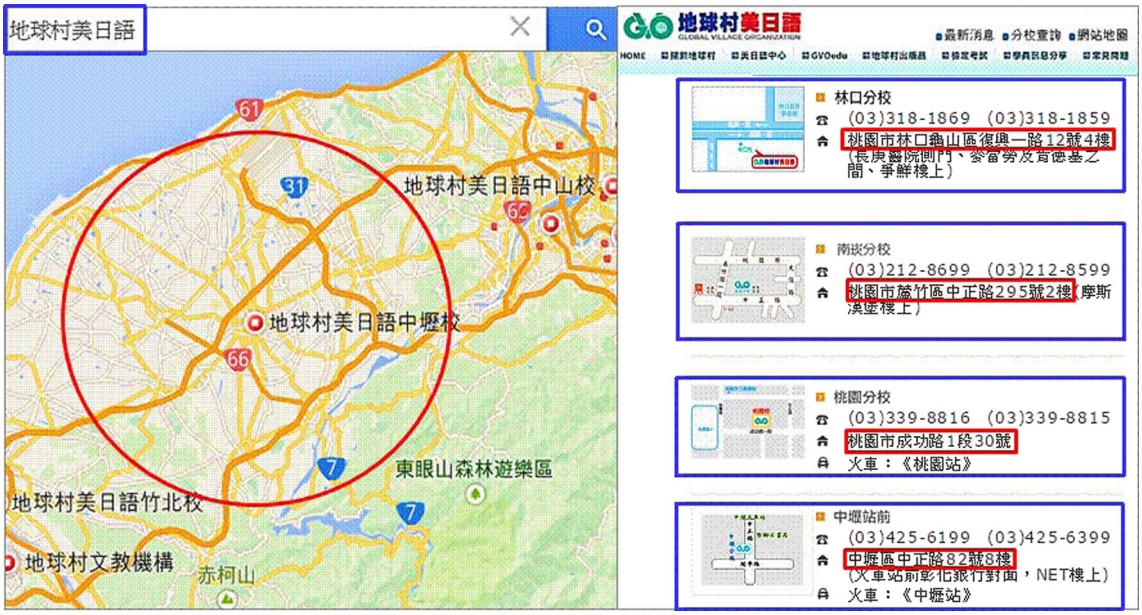

Fig. 1 Google Maps results from a query for "Global Village Organization (地球村美日語)." There is only one school branch in Taoyuan on the map, whereas in fact, there are four school branches in the city

For example, the Global Village Organization (GVO, 地球村美日語)[2] is a business that provides English and Japanese classes with 66 school branches in Taiwan. However, not all school branches are listed on Google Maps. As shown in Fig. 1, there are four branches in Taoyuan, but only one is listed on Google Maps. In fact, if we query Google search engine for "Global Village Organization," we can discover some of these branches from the search snippets and GVO's official website. In this paper, we are interested in finding a method to complement location-based services on the Web by exploiting these webpages.

To enhance local search on maps, we propose an online module to query search engines for potential POIs from search snippets related to users' queries via address extraction. The reason for doing so is that address information is indispensable for positioning on maps and comparably easy to collect and extract. Thus, when a map service fails to return a POI from a query, the online module can still sometimes locate the POI via the address extraction. As shown in Fig. 2, we can easily find several addresses for "Starbucks" by querying "Starbucks branch store" (i.e., 分店、門市). One of the URLs even contains a list of all Starbucks locations. Imagine the use case on a smartphone, users must manually copy a postal address (e.g., 台北市士林區士東路123號) to locate it on a map (see the bottom-right box in Fig. 2). With the proposed online module, there is no need to copy and paste between browsers and map services.

To verify the applicability of the proposal to find potential POIs via online address extraction from search snippets, we use one hundred store names as queries (e.g., "Yamaha music," "Edward Daycare," etc.) to check the potential for obtaining addresses from the top ten search snippets. The results show that 80 % of queries provide the correct addresses in the top ten search snippets using (merely) store names as queries, whereas 10 % of the queries could not be found with Google Maps. Note that the ratio (i.e., the recall) can be further increased by including more snippets, by expanding the query, or by visiting individual pages via the URLs, despite the risk that some addresses may be irrelevant.

In addition to extracting addresses for locating POI locations on maps, we also need the associated information for each POI, as shown in the message box on the bottom-right of Fig. 2. The intention is to provide text snippets for disambiguations of multiple POIs on maps such that users can differentiate various locations using the associated information in conjunction with indexing and enriched POIs for future search. With this function, users can differentiate various marks when multiple addresses are placed on maps.

In this paper, we applied two techniques to accomplish this novel idea: postal-address extraction and associated information segmentation. Postal-address extraction has been studied in various countries—e.g., in Australia [4] and Brazil [6]—and in various languages. In the past, address extraction required large gazetteers that were expensive and unavailable in many countries. Other research applies pattern-based [4, 19, 23] or ontology-based [6, 10] methods to match addresses on webpages. In this paper, we

[2] http://www.gvo.com.tw/

Fig. 2 Search snippets from Google contain many addresses for "Starbucks" in Taipei (*top*). One of the URLs provides a complete listing of Starbucks locations in Taipei (*bottom-left*), whereas the bottom-right box shows the results from the copy and paste operations with the maps

apply machine learning to train a linear-chained conditional random field (CRF) sequence-labeling model for address extraction.

Meanwhile, we also propose a novel technique to segment the associated information for each address when multiple addresses are found on the same page. Unlike previous research [13] based on HTML-tag patterns, which assumes a single format for all (address) data presentation, the proposed technique makes use of the tree structure based on a document-object model (DOM) and the farthest distinguishable ancestor (FDA) of each address to segment address-associated information for each POI. Because the FDA allows the address data to be presented in a different format, the new algorithm can significantly improve the recall performance.

The major contributions of this paper include the following:

- a novel idea to enhance local/POI search on mobile devices via an online extraction process,
- application of machine learning to Chinese postal-address extraction, and
- an effective technique to segment the associated information for each address from a multi-address webpage.

We demonstrate the feasibility of applying state-of-the-art artificial-intelligence techniques to construct an online POI search service via address extraction and associated information segmentation. The experiment shows the proposed techniques achieves a 0.97 F1-measure for Chinese address extraction and the novel algorithm based on a DOM tree structure improves associated information segmentaton from a F1-score of 0.811 to 0.964.

The remainder of the paper is organized as follows. Section 2 describes related work on address and information extraction. Section 3 introduces the system architecture and focuses on online Chinese address extraction and associated information segmentation. The experiment is explained in Section 4. Finally, the experimental results are discussed and conclusions are presented.

2 Related work

Information extraction (IE) is the task of automatically extracting structured information from unstructured text documents, and IE is designed to test how well a machine can understand the messages written in natural language and automate mundane tasks performed by humans. In addition, IE is important for the semantic Web, and it intended to permit the textual information to be shared and reused across applications. Some of the representative conferences on IE include the DARPA-sponsored Message Understanding Conference held from 1987 to 1998, and the Text Retrieval Conference. Many machine-learning techniques, whether supervised or unsupervised, have been studied to solve the IE tasks automatically.

Owing to the difficulty involved, current approaches to IE focus narrowly on restricted domains. Address extraction, which is a specific form of IE, extracts complete addresses and has many social and economic applications. Generally speaking, a complete address comprises the house number, street, city, region, county, state, and zip/postal code. Some components are optional, such as postal codes and directions, and the various components are commonly placed in different orders. To identify the address boundary, an IE program often uses keywords such as "road," "street," or "avenue" to locate candidate strings that might contain a complete address, after which it determines the boundary using various approaches. While candidate string segmentation is not necessary, it usually speeds up the processing procedure. We review recent studies according to their extraction approach: pattern-based and machine-learning methods.

2.1 Pattern-based methods for address extraction

Pattern-based methods generally rely on the gazetteer or ontology to provide names for cities and states. A gazetteer is a geographical dictionary, and it is an important reference for location names. An ontology is a formal representation of a set of concepts within a domain and the relationship between those concepts.

Asadi et al. proposed a pattern-based address-extraction approach that uses several address patterns and relatively little geographical data to extract English-language postal addresses in Australia [4]. They also adopted HTML and vision-based segmentation to improve the overall extraction quality, obtaining an F-measure of 0.83 for a dataset of 1100 webpages and 2030 addresses.

Lin et al. used a layout-format (i.e., vision-based) method to divide input pages into text blocks and verify whether a candidate text block contains a postal address with address patterns [19]. This method has an accuracy of 0.89. Nevertheless, 44 pages for testing were insufficient to demonstrate the feasibility of this method.

Ahlers and Boll tailored a geospatial Web service to recognize full addresses in Germany using a combined extraction and verification process to facilitate mobile pedestrian-focused applications. A geoparser was designed to determine the valid addresses via analysis of the address structure, components, and dependencies, as well as a full database of address-related information including postal codes, city names, and street names[2, 3]. They crawled about approximately 180,000 webpages in Germany and extracted 25,000 addresses that matched the definition of a complete address.

Cai, et al. designed an address ontology with suite information, municipal locations, and regional positions, and applied a graph-matching technique for English-language address extraction [10]. The result yielded an F-measure of 0.734 on six Yellow Pages and five Yahoo business pages. Borges et al. proposed 11 patterns based on an ontology with basic addresses, complements, location identifiers, and subcategories, such as phone numbers and zip codes [6]. An extraction of Brazilian addresses from a collection of over 4 million webpages was conducted. However, precision and recall were not reported, owing to the large quantity of testing data.

2.2 Learning-based methods for address extraction

Supervised learning methods are widely used in many applications. There are two kinds of supervised learning tasks: classification and sequence labeling. Classifiers such as Naive Bayes, C4.5 decision tree and Support Vector Machines (SVMs) [14] are commonly used learning algorithms designed to assign a label to a given input. Others such as Hidden Markov Model (HMM) [5], Maximum-Entropy Markov Model (MEMM) [21] and Conditional Random Fields (CRF) [22] are developed to assign a label to each member of a sequence of inputs.

Ourioupina (2002) defined a classification problem with six location types (viz., the city, region, country, island, river, and mountain) [24]. An accuracy of 89.1 % was obtained using a C4.5 decision tree classifier, and ten-fold cross-validation is reported. Moreover, [31] extended this framework to create new gazetteers with a small dataset using a bootstrapping approach.

Borkar [7] proposed an automated method for elementizing postal addresses based on the HMM [7]. The motivation was to normalize postal addresses, because each region has evolved its own unique style of writing addresses. This not only enabled better querying, but also provided a more robust way of detecting duplicate addresses and identifying households. The experiment was conducted with three datasets containing 740 US addresses, 2388 student

addresses and 769 customer addresses. Given that a third of the dataset was used for training and the remaining two thirds were used for testing, the results yielded an accuracy of 99.6 %, 89.3 %, and 83 % for three datasets, respectively.

Yu [32] proposed a hybrid method that combines pattern-based and machine learning approaches by classifying each token into four classes: START, MIDDLE, END, and OTHER. Then, they segmented the string from the START label to the END label as a complete address. On a dataset of more than 1700 pages, they showed an F-measure of 0.876, obtained using a C4.5 decision tree.

Chang and Li [13] applied C4.5, SVM, and CRF to BIEO tagging (BEGIN, INTERMENIATE, END, and OTHER) for English-language address extraction. With 14 designed features, the F-measure for Yu's dataset achieved the highest score of 0.913 with CRF, outperforming C4.5 and SVM with 0.876 and 0.903, respectively. Therefore, in this paper, we follow this trend by applying CRF to Chinese postal-address extraction and compare the performance by adopting various techniques for Chinese-language preprocessing.

2.3 Associated information segmentation

In addition to postal-address extraction, associated information segmentation is a new task proposed by [13], dealing with the segmentation of webpages that contain multiple addresses. This task is similar to unsupervised web-data extraction, which has been studied for more than ten years [8, 11, 17] and continues to be of considerable interest in recent years [9, 20, 33, 34] for search-result record extraction. For example, [20] proposed a vision-based text-segmentation method for postal addresses from webpages. [33] studied the problem of segmenting data records of webpages from which they extracted data items/fields. Overall, the research into extracting structured data from the so-called Deep Web usually focuses on record boundary detection and attribute alignment. More surveys of web-data extraction can be found in [12, 17, 28].

However, the problem of associated information extraction from pages with multiple addresses that is discussed here has the advantage of using the extracted addresses as a means for locating the associated information for each address. Moreover, address-associated information does not the require strict extraction of individual attribute-value pairs. Hence, the major problem is to find the record boundary of each address for such listings.

Chang and Li [13] proposed a pattern-growth algorithm based on HTML-tags for associated information extraction from pages with multiple addresses. However, such tag patterns are often fragmented, owing to optional fields and nested data. Even without optional fields, determining the correct boundary of an address' associated information is a problem that remains to be solved. The limitation of Chang and Li's method comes from the assumption that all addresses in a page belong to the same list and form a continuous block without interruption. Therefore, their idea is to grow a tag pattern from each address landmark for matching the associated information block repeatedly. In this paper, we eschew this assumption and make use of a DOM tree structure to focus on the subtrees that encompass the address landmarks to improve the performance of associated information extraction.

3 System architecture

To enhance map services, we propose a map service that integrates Google Maps and Google search engine as shown in Fig. 3. Given a POI query, we expand the query with address keywords, like "市" or "路" or "街" or "號", and branch keywords such as "stores" (分店、門市) to obtain search snippets from Google search engine. Simultaneously, the POI query is sent to Google Maps to locate the POI. By extracting addresses from the search snippets and surrounding context as associated information, we can provide more locations related to the POI query to complement existing map services. To increase the probability of finding

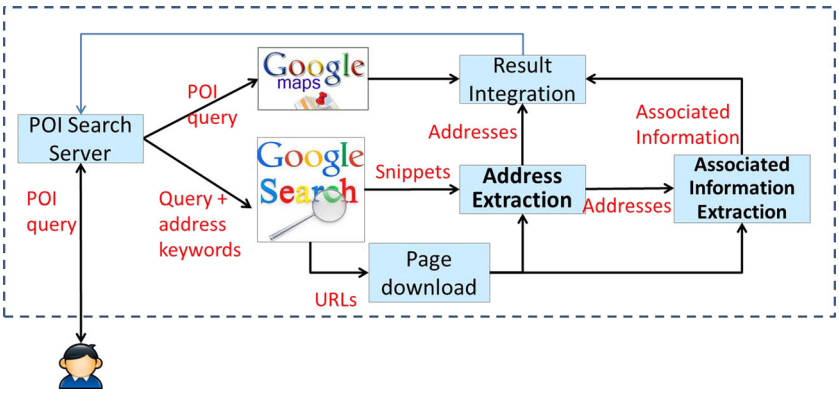

Fig. 3 Overview for the application of online address extraction and the associated information extraction

more addresses, we also visit the URLs from Google search results to download the original pages. Finally, the extracted address with the associated information are integrated with the search results from Google Maps for users. In this paper, we focus on two tasks: Chinese postal-address extraction via sequence labeling and associated information extraction for multiple addresses on one page.

3.1 Postal address extraction

In the past, address-extraction techniques required large gazetteers that were expensive and unavailable in many countries. Recently, some studies have applied structured learning based on decision trees or SVMs to train a model to extract addresses. Because techniques for unstructured learning based on the HMM or CRF have progressed tremendously, we adopt unstructured learning based on sequence labeling to resolve this problem.

The proposed approach for postal-address extraction comprises three steps. As shown in Fig. 4, the preprocessing step includes three modules—namely, eliminating HTML tags, tokenization, and candidate string segmentation. The first module eliminates HTML tags such that the sequence-labeling model can work with different webpages. Then, a set of features is calculated for the learning module to adjust weights for maximizing the log likelihood. During testing (see the dashed lines), the input page will proceed through the preprocessing step and feature extraction. Finally, the trained model is employed to label candidate strings to obtain a complete address.

Figure 5 provides an overview of the machine-learning-based address extraction. The input page first undergoes the HTML-tag elimination process to derive pure text. The task of tokenization then divides the text string into basic processing units, whether via individual Chinese-character segmentation (ICCS) or via Yahoo's Chinese word-segmentation (YCWS) to obtain character or word tokens. Next, candidate address strings are identified with key suffixes of city names, such as "縣," "市," and "鎮," and extended with a window size of eight words or characters. The key point is to transform the address-extraction task into a sequence-labeling problem with BIEO tagging and IO tagging. Finally, the strings are extracted with BIEO tagging, where the beginning label "B" and the ending label "E" are combined with the strings by IO tagging and the maximal scoring subsequence.

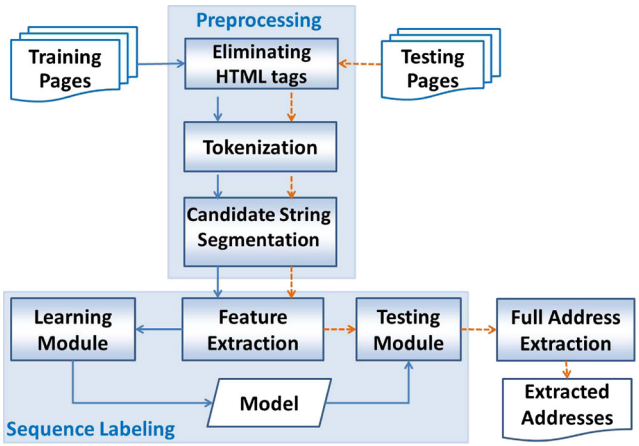

Fig. 4 Flowchart for machine-learning-based address extraction. The solid line denotes the process for the training pages, and the dashed line denotes the process for the testing pages

3.1.1 Tokenization

In English, tokenization is applied to split a string of text into a sequence of tokens based on spaces and punctuations. By contrast, Chinese sentences are sequences of Chinese characters, where each word is composed of between one and four characters without a clear boundary. Therefore, word segmentation is an important step for Chinese-language processing. However, it is also possible to use individual Chinese characters as the basic units for processing. In this study, both Yahoo's Chinese word-segmentation and individual characters for tokenization are used.

- Yahoo Chinese Word Segmentation (YCWS): YCWS is a standard application interface provided by Yahoo for Chinese word segmentation. It integrates linguistics and information retrieval with the enormous corpus on Yahoo.
- Individual Chinese Character Segmentation (ICCS): For this tokenization, each individual Chinese character is understood as a single token.

3.1.2 Candidate string segmentation

To locate candidate strings that might contain an address, key suffixes for city/county/road/street names are used as landmarks, and additional tokens are extended to form a candidate string. Supposing the extended window has w tokens, the system would include w tokens before and after each key character to generate a segment of $2w+1$ tokens. If two candidate strings overlap, they are joined together as one candidate string.

For example, the key suffix for city names "縣" (county) with a window size of eight characters will result in a candidate string such as "學堂地址: 510彰化縣員林鎮山腳路三段," as shown in Fig. 5. Similarly, the key character "鎮" (town) leads to a candidate string of "址:510彰化縣員林鎮山腳路三段2巷6." Because these two candidate segments overlap, they are joined together to form a single candidate segment. The process is repeated

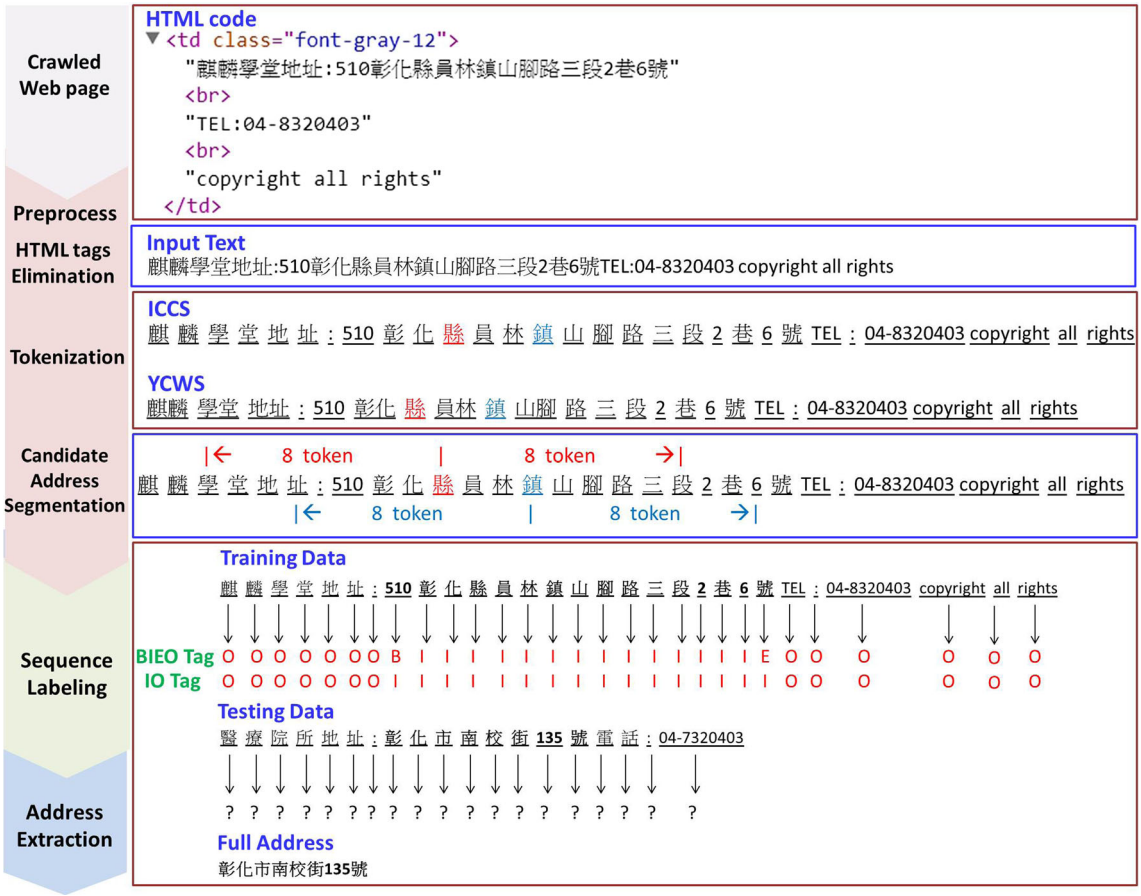

Fig. 5 Example of Chinese address extraction

for other key characters, such as "鎮" (town), "路" (road), and "號" (number). The parameter w of the extended length will be determined by experiments with the corresponding tokenization.

We use administrative divisions, street suffixes, house numbers, and directions as landmarks to segment candidate strings for sequence labeling. Administrative-division suffixes include the county (縣), city (市), town (鎮), township (鄉), district (區), village (村), neighborhood (里), and neighbor (鄰); street suffixes include avenue (道), road (路), street (街), section (段), lane (巷), and alley (弄); and house-number suffixes include the number (號), floor (樓), and room (室). Unlike English-language address extraction in MapMarker [13], we do not use state names and country names, such as "PA," "Florida," "USA," or "CA" as landmarks (see Table 1). The common address suffixes used for Chinese address extraction include a total of eight administrative characters, six street suffixes, and three house numbers. By contrast, there are 29 street suffixes, 70 state names, 16 directional keywords, and 3 country names for English-language address extraction.

3.1.3 Features for sequence labeling

For each token, we enumerate the following features as indicators of an address segment. First, the key tokens for candidate string segmentation are used—i.e., we consider whether a token is a suffix for a county, township, village, street, lane or alley, house number, and building, etc., for a total of seven features. By contrast, MapMarker for English-language address extraction uses only two features—namely, street and direction. Second, a set of dictionaries is constructed to include common location names and countries. For example, Taiwan's 368 counties from Wikipedia (153 townships, 41 towns, and 157 districts) are used as the gazetteer for segmentation-based tokenization using YCWS. By contrast, MapMarker adopts ANNIE (A Nearly-New Information Extraction) system to annotate geographical names in a document [16].

In addition, other contact keywords, such as telephone (縣) and address (市), are among the features, because these tokens are frequently included with an address. Finally, zip codes and room numbers are usually numbers of a fixed length, whereas street numbers often vary in length. As a

Table 1 Common address suffixes used in this paper (top) and with MapMarker (bottom)

Task	Suffix Types	Key Words or Characters
Chinese Address	Administrative division	county(縣), city(市), town(鎮), township(鄉), district(區), village(村), neighborhood(里), neighbor(鄰)
	Street suffix	avenue(道), road(路), street(街), section(段), lane(段), alley(弄)
	House No.	number(號), floor(樓), room(室)
English Address	Street suffix	Street, St., Road, Rd., Avenue, Ave., Alley, Lane
	State name	PA, Florida
	Direction	North, N., South, S., West, W., East, E., N.E.
	Country name	USA, Canada, UK

The paper is adopted these suffixes as landmarks for segmenting candidate strings for sequence labeling

result, six lengths of numbers ranging from one to five, as well as values larger than five are chosen. Table 2 enumerates the features designed for ICCS and YCWS tokenization for Chinese. By contrast, MapMarker does not differentiate various lengths, but rather makes use of letter cases, such as all capitals or all lower case, and the initial capital highlights the word features for English-language postal-address extraction.

There are two tagging methods for information extraction tasks. IO tagging has only two labels: "I" refers to the inside of the extraction target, and "O" refers to the outside. To distinguish the beginning and ending tokens from tokens inside the extraction target, BIEO tagging provides two additional labels for the beginning and ending labels. For example, the corresponding labels for each token in the string "麒麟學堂地址:510 彰化縣員林鎮山腳路三段2巷6號" are shown in Fig 5.

3.1.4 Learning module

After preprocessing and feature extraction, a model is trained with the learning module. This research uses Conditional Random Fields (CRF) as the learning algorithm. The CRF models [18, 30] are undirected graphical

Table 2 Features used for Chinese postal address extraction and MapMarker for English

	Chinese Feature	ICCS	YCWS	English Feature	Example
1	CountyCity	縣,市	桃園縣,台北市	Allcaps	ATM, USA
2	Township	鎮,鄉,區	蘆竹鄉,大安區	Alllower	suite
3	Village	村,里,鄰	三合村,五分里	State	California, CA
4	StreetRoad	道,路,街	八德路,和平街	Street	St., Avenue, Ave
5	LaneAlley	段,巷,弄	三巷,九弄	Initialcaps	Street, Road
6	HouseNo	號	十五號	Digitpun	12th
7	Building	樓,室	六樓,B室	Initial	N.
8	ContactTag	地,址,電,話	地址,電話	Letterpun	Address:
9	Punctuation	、：；•:;.,“”	Punctuation	, - :	
10	ChineseNo	一,二,三,四,五,六	ANNIE Location	True / False	
11	AllDigits	42011,0937137659	Phone	(925)223-25459	
12	DigitLen1	5,6,7	Alldigits	95846	
13	DigitLen2	11,32	Direction	North, N	
14	DigitLen3	420,260	ZipCode	96548, 96584-1547	
15	DigitLen4	5566,1234			
16	DigitLen5	42011			
17	DigitLong	327363,4227151			

The difference between ICCS and YCWS is the length of the segmentation in Chinese

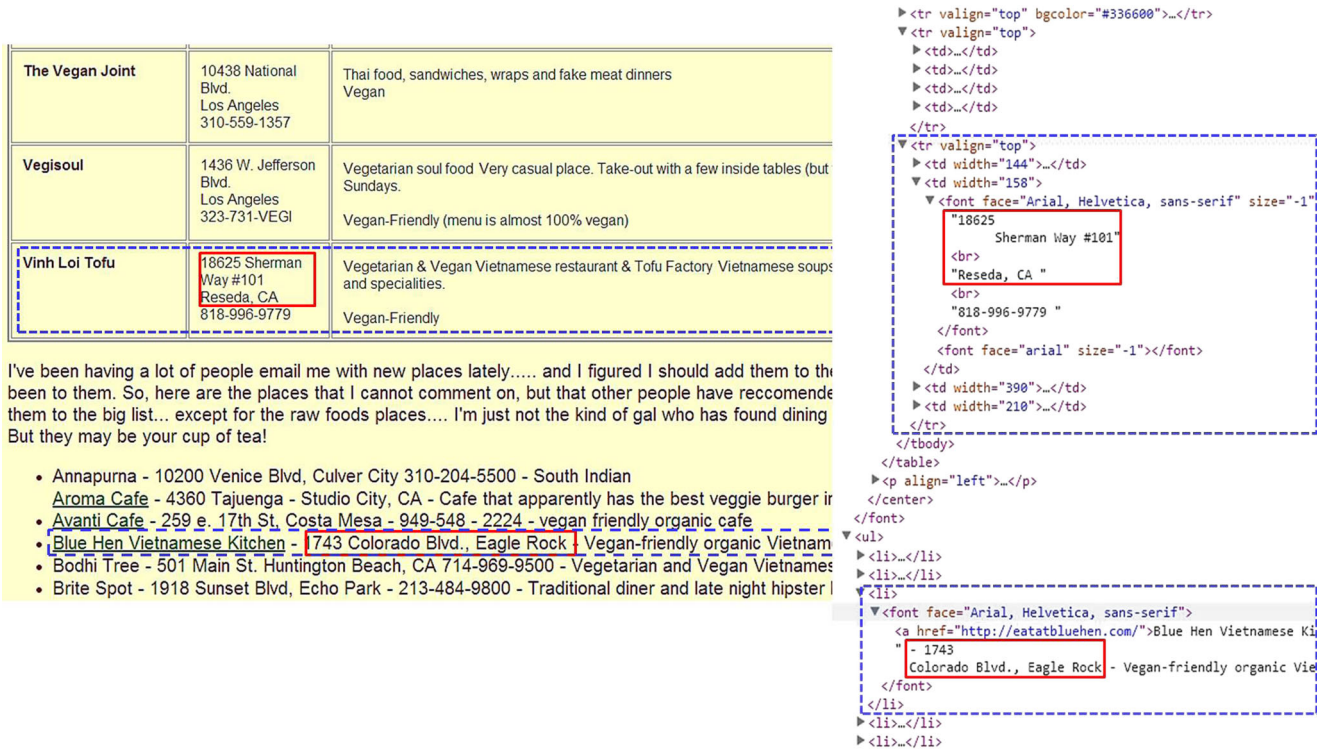

Fig. 6 Two address layouts in a webpage and the corresponding HTML codes (the red blocks denote the detected addresses and the blue dashed blocks denote the corresponding associated information). Such multi-list layout can not be handled by Chang and Li's method which treats just one record pattern

models that define the conditional probability distribution over labeled sequences given a particular observation. Compared with generative models, such as HMM, CRF has the advantage of relaxing the independence assumptions required for calculating the joint probability. In addition, CRF avoids the bias toward states, with fewer successor states than other directed graphical models such as MEMM. CRF is now widely used for sequence segmentation and labeling tasks in many fields such as bioinformatics, computational linguistics, and speech recognition.

A special kind of CRF—one where all the nodes in the graph form a linear chain—is commonly used for information extraction. The problem of learning a linear-chain CRF can be defined as follows, given a sample set of

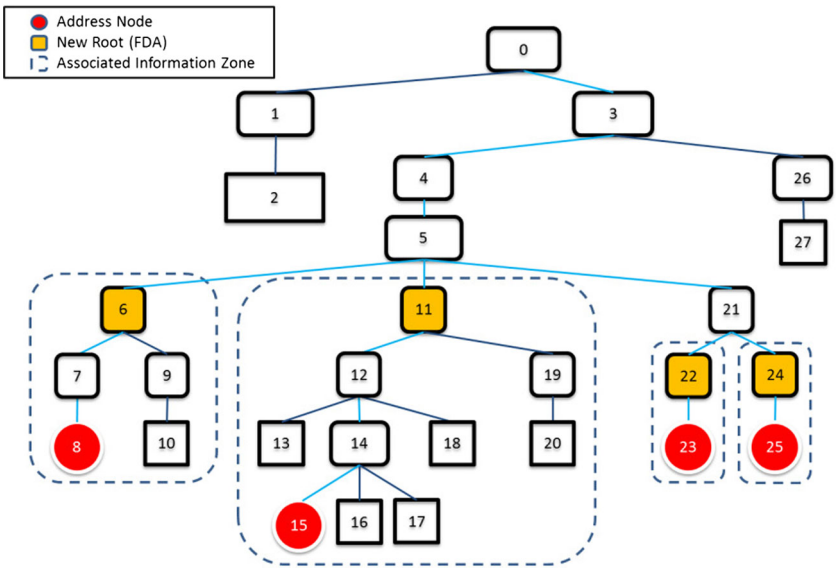

Fig. 7 Associated information block for four addresses

Table 3 Example for the Farthest Distinguishable Ancestor (FDA) algorithm

Address node	Path P	FDA
$t_1 = n_8$	$p_1 = /0/3/4/5/6/7/8$	$a_1 = n_6$
$t_2 = n_{15}$	$p_2 = /0/3/4/5/11/12/14/15$	$a_2 = n_{11}$
$t_3 = n_{23}$	$p_3 = /0/3/4/5/21/22/23$	$a_3 = n_{22}$
$t_4 = n_{25}$	$p_4 = /0/3/4/5/21/24/25$	$a_4 = n_{24}$

training sequences $\{X_1, X_2, \ldots\}$ along with their corresponding label sequences $\{Y_1, Y_2, \ldots\}$. The task of learning is to discover the best possible potential functions, such that the log-likelihood is maximized. Once the potential functions are determined, the inference process can be defined as follows, given a new observable sequence X, find the most likely label sequence Y* for X, i.e. compute $Y^* = argmax_Y P(Y|X)$.

Formally, the conditional distribution of the labels Y given the observations X has the form,

$$P(Y_1, ..., Y_N \mid X) = \frac{1}{Z_X} \prod_i^N \varphi_i^v(Y_i, X) \varphi_i^e(Y_i, Y_{i-1}, X) \quad (1)$$

where $\varphi_i^v(Y_i, X)$ and $\varphi_i^e(Y_i, Y_{i-1}, X)$ are potential functions for the node and edge, respectively. Most sequence-labeling applications use log-linear potential functions.

$$\varphi_i^v(Y_i, X) = exp\left(\sum_j u_j g_j(Y_i, X, i)\right) \quad (2)$$

$$\varphi_i^e(Y_i, Y_{i-1}, X) = exp\left(\sum_k \lambda_k f_k(Y_i, Y_{i-1}, X, i)\right) \quad (3)$$

where g_j and f_k are the node- and edge-feature functions that depend on the labels at the current and previous position, respectively; μ_j and λ_k are parameters to be estimated from the training data, and Z_x is the normalization factor, which takes the form,

$$Z_x = \sum_Y \prod_i^N \varphi_i^v(Y_i, X) \varphi_i^e(Y_i, Y_{i-1}, X) \quad (4)$$

There are several packages available for linear-chain CRF. This paper adopts CRF++ [15], an open-source implementation based on LBFGS (a quasi-Newton algorithm for large-scale numerical optimization). The input file consists of token sequences, with each token represented by the 17 features described in Section 3.1.3. Empty lines are employed to identify sequence boundaries.

In addition to the primal features for each token, CRF++ provides feature templates and special macros that can generate different composite features. There are two types of templates: Unigram and Bigram output labels. The Unigram template makes use of the current output label with primal features coded by macros to generate feature functions. The Bigram template uses the previous output labels, resulting in more feature functions. In this task, both types are applied.

3.1.5 Full address extraction

For BIEO tagging, we extract a segment that begins with a B-label followed by at least one I-label ending with an E-label. For a testing sequence with each token properly labeled, the corresponding segment can be extracted accordingly.

While BIEO tagging has B- and O-tags that can act as sentinels for boundary detection, IO tagging lacks such labels. Thus, it is possible for mislabeled tokens to result in small address segments that are incomplete. Consequently, we try to apply a maximal scoring subsequence (MSS) to the labeled sequence for full address extraction. An MSS is a contiguous subsequence having the greatest total score among all proper subsequences. The MSS problem arises in

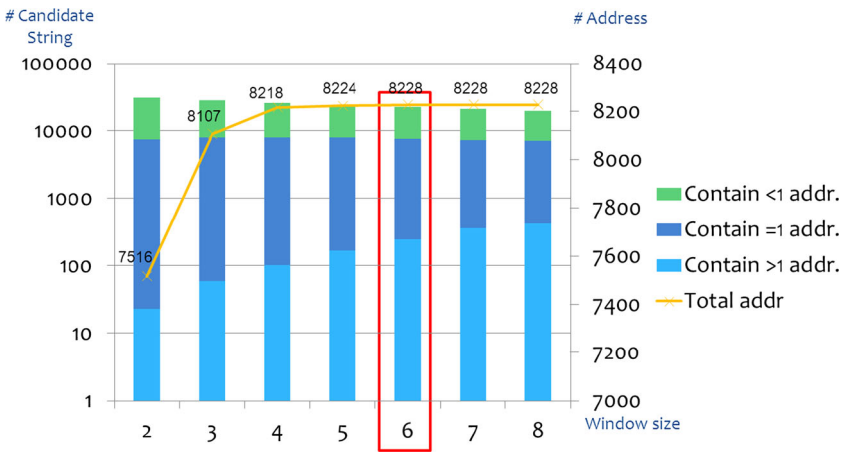

Fig. 8 Number of candidate strings and addresses with various window sizes for ICCS tokenization

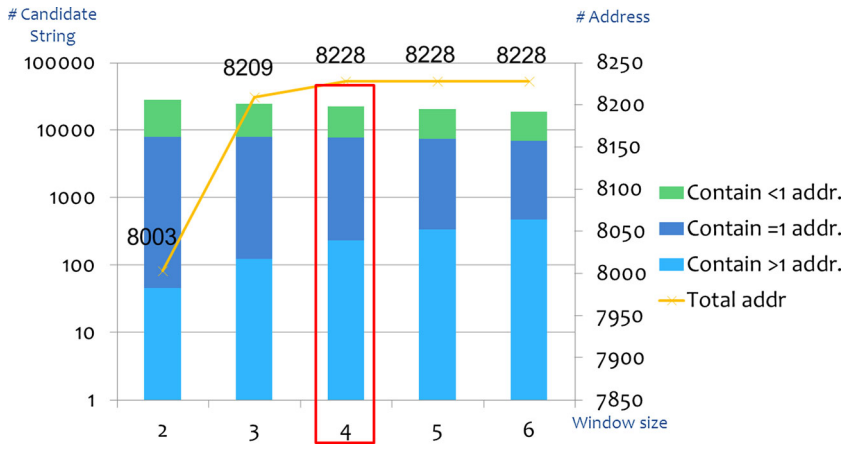

Fig. 9 Number of candidate strings and addresses with various window sizes for YCWS tokenization

biological-sequence analysis, where the high-scoring subsequences correspond to regions of unusual composition in a nucleic acid or protein sequence [27]. It has also been applied to the extraction of text from articles on the Web [25].

Because the MSS algorithm requires the input to be a sequence of positive and negative values, the probability of each token of an I-label is subtracted by 0.5 to ensure that the value falls within an interval between [-0.5, 0.5]. For example, given a sequence of probabilities (0, 0, 0, .99, .99, .99, .40, .99, .99, .99, 0) where each value denotes the probability that a token has an I-label, the transformed number sequence will be (-.5, -.5, -.5, .49, .49, .49, -.10, .49, .49, .49, -.5). The MSS task is to find the maximal scoring subsequence (.49, .49, .49, -.10, .49, .49, .49). Assuming that a candidate segment might contain only one address, the MSS algorithm can be employed to extract the complete address. The detailed algorithm can be found in [25].

3.2 Address associated information segmentation

In many extraction tasks, there is an explicit need to relate the extracted entities to other attribute values, for example in order to associate with a store name to form a POI relation. When a page contains multiple addresses, there is an explicit need to segment the input page such that the addresses are associated with some information to identify them on maps. Thus, the task here focuses on associated information segmentation from webpages that contain multiple addresses.

Chang and Li [13] proposed an unsupervised method for associated information segmentation by growing record patterns from HTML-tag strings. However, record patterns can be interrupted by optional data fields, causing incomplete patterns and uncertain address blocks. In addition, there might be more than one layout for displaying postal addresses as shown in Fig. 6. Because a record pattern for one page can not extract associated information for all addresses, Chang and Li's method is inherently limited. In fact, by ignoring the individual data fields and concentrating exclusively on the general associated information, we can focus on the information block represented by DOM tree paths, rather than detailed record patterns.

DOM is a standard representation of HTML documents. Each element, label, and text in HTML is denoted by some nodes in the DOM tree. To focus on the boundary of the associated information, the following three assumptions are made in this study.

- First, the associated information for an address forms a continuous block encompassing the address.
- Second, the information blocks associated with each address are mutually exclusive.
- Third, we assume that the information block of an address is contained in a single subtree in the DOM, such that information blocks expand the region, provided that they are mutually exclusive.

For example, suppose there are four numbered nodes n_8, n_{15}, n_{23}, and n_{25} in a page, as shown in Fig. 7. According to the above-mentioned assumptions, the numbered nodes n_6, n_{11}, n_{22}, and n_{24} will be taken as the root for the information block of the four addresses, respectively, because these information blocks are mutually exclusive (as enclosed by the dotted line in Fig. 7).

Table 4 Performance of Chinese address extraction

Tagging	Tokenization	AvgP	AvgR	AvgF
BIEO	ICCS	0.9713	0.9689	0.9701
BIEO	YCWS	0.9689	0.9580	0.9634
IO	ICCS	0.9527	0.9464	0.9497
IO	YCWS	0.9492	0.9316	0.9403
Baseline: Regular Expression		0.8805	0.9020	0.8911

Table 5 F-measures of BIEO tagging and IO tagging

BIEO Tagging	B	I	E	O
ICCS	0.9923	0.9934	0.9937	0.9983
YCWS	0.9666	0.9797	0.9780	0.9967
IO Tagging		I		O
ICCS		0.9771		0.9940
YCWS		0.9793		0.9939

Table 6 P-values of the t-test with various pairs

2c Pair of two methods	P-value	
IO+YCWS	Regular expression	3.10e-8
BIEO+ICCS	IO+ICCS	5.73e-6
BIEO+ICCS	BIEO+YWCS	9.64e-3
IO+ICCS	IO+YCWS	3.46e-2

Given an input page with a corresponding DOM tree T and a set of N postal addresses located in the terminal nodes[3] of the DOM tree T, the path p_i of each address/terminal node t_i is an array list of nodes from the root to t_i in order—i.e. $p_i[k]$ denotes the k-th node of p_i from the root, as shown in Table 3.

Definition 1 An ancestor node $a_i = p_i[k]$ is called the farthest distinguishable ancestor (FDA) of its address nodes t_i if it is the farthest node from t_i and if it is distinct from all ancestor nodes of other addresses. In other words, $p_i[k-1]$ is an ancestor of some address nodes t_j, $j \neq i$.

For example, the farthest distinct ancestors for the four address nodes are exactly node n_6, n_{11}, n_{22}, and n_{24}, and these are equivalent to the associated information-block that the process aims to discover (see Table 3). Thus, the problem of associated information-block segmentation can be solved by finding the FDA nodes for each extracted address.

3.2.1 Discover farthest distinguishable ancestors

The algorithm that discovers the FDAs for the given address nodes is shown in Algorithm 1. First, the FDAs of all nodes a_i are initialized to be null for all i (Lines 3-5). Next, we count the number of paths that share the same nodes at each level (Lines 7-9). For nodes near the root, the occurrence count is greater than 1, and this value decreases for nodes that are near leafs. If the occurrence count for $p_i[level]$ is one and the FDA for p_i is null, a_i, will be assigned $p_i[level]$ (Lines 11-13). For example in Fig. 7, the four address paths from the root to the nodes n_8, n_{15}, n_{23}, and n_{25} share node n_0, n_3, n_4, and n_5. Therefore, the occurrence count is 4 for nodes n_0, n_3, n_4, and n_5. At Level 5, nodes n_6 and n_{11} exist in only one path (to n_8 and n_{15}, respectively). As a result, the occurrence count is 1 and the nodes are assigned the FDA for the address nodes n_8 and n_{15}.

[3]If an address is scattered over multiple nodes, these nodes will be combined into one terminal node.

The above algorithm benefits from the following advantages: (1) it is a very simple concept with linear-time complexity; (2) because the conditions of FDAs are mutually exclusive in web-structured trees, the correctness of the other associated information that is extracted will not be affected; (3) the algorithm does not require the address entities to be located under the same parent node, allowing different layouts in the same page; (4) the method can be applied to other languages, because the object analyzed is the DOM tree structure, not the address itself.

Algorithm 1 Discovering the farthest distinguishable Ancestors

```
1:  inputs
1:      P = {p_1, p_2, ..., p_m}
2:  outputs
2:      A = {a_1, a_2, ..., a_m}
3:  for i = 1 to m do
4:      set a_i = null // initialization
5:  end for
6:  for level = 1 to maxLength(P) do
7:      for i=1 to m do
8:          Compute the occurrence count of each node for
            p_i[level] //check all nodes in each level of paths
9:      end for
10:     for i=1 to m do
11:         if (a_i == null & p_i[level] != null & the
            occurrence count of node p_i[level] ==  )
12:             set a_i = p_i[level]
13:         end if
14:     end for
15: end for
```

4 Experiments

The experiments comprised two parts. The first part focused on address extraction, and the second part involved verifying the performance of the associated information segmentation with the help of webpage partitions.

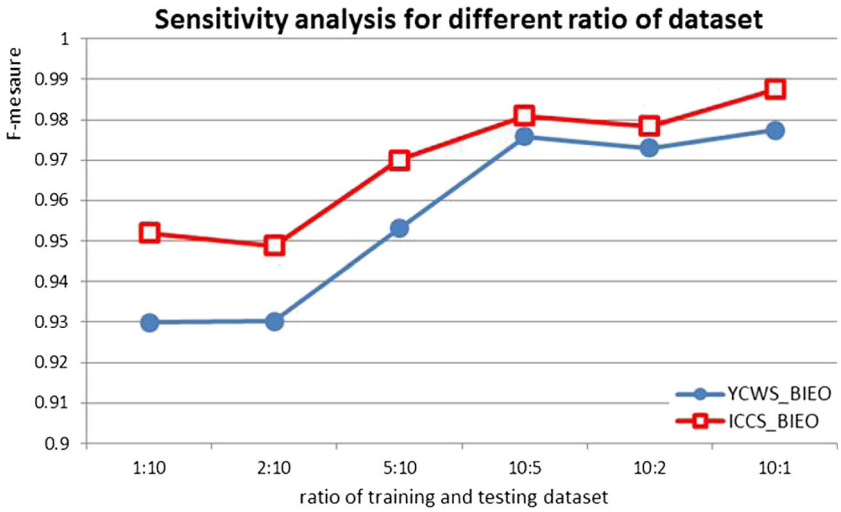

Fig. 10 Sensitivity analysis for different training and testing splits of the dataset

4.1 Data collection

In order to verify the effectiveness of the machine-learning-based approach for English and Chinese address extraction, we utilized an English dataset that was collected by [32] to compare the performance of English-language address extraction. The English dataset was obtained with three topic queries (contact, hotel, and pizza) from Google. The dataset contained a total of 1,740 webpages and 9,724 U.S. addresses. Among them, 1,205 pages contained a single address, and 535 pages contained multiple addresses. The dataset was randomly split into ten equal-sized and mutually exclusive subsets. The learning module was trained and tested ten times, and each time it used all but one of the ten subsets.

Thus far, there is no public dataset for the task of Chinese address extraction. Therefore, we prepared the data by querying Google with 19 administrative district names, such as Taipei (台北), Taichung (台中), Hsinchu (新竹), Taoyuan (桃園), and Kaohsiung (高雄). A total of 1,689 webpages, including 12,124 addresses were obtained. The data was divided into a training set and a testing set. For the training set, there were 1,140 webpages with 8,228 addresses. For the testing set, there were 549 webpages with 3,896 addresses.

For the experiments in associated information segmentation, the data was prepared by querying Yahoo with 477 queries that combined administrative district names (i.e., countries and townships) such as Da'an District, Taipei City (台北市大安區), Lukang Township, Changhua County (彰化縣鹿港鎮), etc. We crawled 84,380 webpages and filtered illegal HTML structures using the Tidy tool [26]; finally, we retained 34,604 webpages that contain more than two addresses. For the evaluation, we randomly sampled 250 webpages and manually labeled their associated-information area for a total of 8,414 addresses as the ground-truth dataset.

4.2 Evaluation Metrics

For address extraction, we adopted precision, recall, and F-measures to evaluate individual labels and complete addresses from each page. Precision is defined as the correct ratio of the extracted addresses, and recall measures the extracted ratio of the correct addresses from each webpage. The values for each page were then averaged for comparison. For the English-language-dataset obtained from [32], all addresses were previously tagged with the ground truth, whereas the Chinese dataset was manually labeled based on the success criterion that an address should contain an administrative division and a number to identify the location.

For associated information segmentation, we manually labeled the text blocks near each address as its associated information based on the visual information and the semantic context relation. We define the precision, recall and

Table 7 Performance of associated information segmentation for English addresses

English Pages		Total	Contact	Hotel	Pizza
Information	# of pages	536	230	173	133
	# of addresses	6243	1317	3175	1751
FDA algorithm	Precision	0.914	0.860	0.945	0.899
	Recall	0.907	0.857	0.915	0.929
	F-measure	0.911	0.858	0.930	0.914
Chang and Li's algorithm	Precision	0.807	0.799	0.886	0.669
	Recall	0.822	0.861	0.894	0.663
	F-measure	0.814	0.829	0.889	0.666

F-measures for the associated information segmentation for each address as follows:

$$Precision = \frac{\#\text{ of correct words identified}}{\#\text{ of words identified}} \quad (5)$$

$$Recall = \frac{\#\text{ of correct words identified}}{\#\text{ of correct words}} \quad (6)$$

The values for each page were then averaged to determine the overall performance.

4.3 Parameter tuning for address extraction

Before the sequence-labeling task, the parameter for the window size of candidate string segments was first determined. Using a small window size for address suffixes produces many incomplete addresses, whereas a large window size generates long candidate strings, resulting in a lengthy labeling time. Therefore, candidate strings generated with various window sizes using the training set were examined. The candidate strings were divided into three groups according to the number of addresses contained in the candidate strings. The first group included candidate strings that contain no address. The second group included candidate strings with only one address, and the third group included candidate strings that contain more than one address.

As shown in Fig. 8, the number of candidate strings dropped from 23,788 to 12,841 when the window size increased from 2 to 8 for ICCS tokenization. More obviously, the number of candidate strings that contain more than one address increased from 23 to 426 with the window size increased from 2 to 8. A log scale for the number of candidate strings was used, because the numbers varied from dozens to tens of thousands. Note that the number of overall candidate strings is ostensibly high for a window size of 2, although many of these addresses were incomplete fragments. The actual number of addresses captured by these address suffixes was 7,516 with a window size of 2. The number increased to 8,228 with window size of 6 and remained at the same level with larger window sizes. Therefore, we decided on a window size of 6 for ICCS.

A similar trend was observed for YCWS tokenization, as shown in Fig. 9. The number of overall candidate strings dropped from 19,889 to 11,755 when the window size increased from 2 to 4, whereas the number of candidate strings that contain more than one address increased from 46 to 470. The actual number of addresses captured by these address suffixes increased from 8,003 with a window size of 2 to 8,228 with a window size of 4, and remained at that same level for larger window sizes. Therefore, we choose a window size of 4 for YCWS.

Table 8 Performance of associated information segmentation for Chinese addresses

Chinese pages	AvgP	AvgR	AvgF
FDA algorithm	0.967	0.962	0.964
Chang and Li's algorithm	0.813	0.809	0.811

4.4 Performance of address extraction

For Chinese address extraction, the CRF learning module was conducted with two tagging methods: BIEO and IO tagging. Afterwards, we compared the complete addresses that were extracted among four combinations of the two tagging methods and two tokenization methods are compared. Regular expressions for Chinese address extraction were manually constructed as a baseline. Because ICCS tokenization with BIEO tagging has a considerably favorable F-measure (greater than 0.99) with all address-related tags (B, I, and E), its performance for complete-address extraction was also the best (0.97 F-measure), as shown in Table 4. For the two combinations with IO tagging, the F-measure for complete-address extraction did not exceed 0.95, even with the help of the maximal scoring subsequence. Some of the errors were caused by incorrect labeling of the I-tag between two addresses. Overall, BIEO performed better than IO, and ICCS performed better than YCWS, owing to some erroneous labeling that occurred from incorrect segmentation.

Table 5 shows the experimental results obtained with BIEO and IO tagging, by examining the respective F-measures for each tag. For BIEO tagging, ICCS performed better performance than YCWS in address-related tags (B, I, E) and with an equivalent F-measure for O-tags. As for IO tagging, the performance between ICCS and YCWS was comparable. However, the F-measure for address-related tags (I) was lower than that for BIEO. Overall, Chinese word segmentation does not perform better, and indeed may deteriorate the performance as a result of incorrect word segmentation.

Table 9 Effectiveness of associated information segmentation with various quantities

Chinese Pages (# of addresses<= 10)	AvgP	AvgR	AvgF
FDA algorithm	0.846	0.915	0.879
Chang and Li's algorithm	0.722	0.766	0.743
Chinese Pages (# of addresses> 10)	AvgP	AvgR	AvgF
FDA algorithm	0.975	0.965	0.970
Chang and Li's algorithm	0.821	0.813	0.817

Fig. 11 *Left*: Search results for an example query, "樂禾音樂." *Right*: PowerPOI app interface showing the extracted address "中壢市中央西路二段223號" and the description of the POI represented by search snippets that contain the address

To compare the performance between different methods, a t-test was employed to examine whether the average F-measure had significant differences. Four pairs were tested individually: IO+YCWS and Regular Expression, BIEO+ICCS and IO+ICCS, BIEO+ICCS and BIEO+YWCS, and IO+ICCS and IO+YCWS. As shown in Table 6, the P-values were 3.10e-8, 5.73e-6, 9.64e-3, and 3.46e-2, respectively. The first two P-values were under 0.0001, demonstrating statistically significant differences at a confidence level of 99.99 %. Thus, BIEO is preferable over IO. Furthermore, the difference between ICCS and YCWS was also statistically significant, with a confidence level of 99 % for BIEO tagging (P-value < 0.01) and 95 % for IO tagging (P-value < 0.05).

In order to verify the effectiveness of the address-extraction approach, we conducted a sensitivity analysis for different ratios of the training and testing datasets. We randomly sampled webpages from the Chinese dataset based on the following ratio:[4] 1:10, 2:10, 5:10, 10:5, 10:2 and 10:1. The approach to address extraction by YCWS and BIEO tagging achieved high F-measures between 0.929 and 0.977. The approach by ICCS and BIEO tagging attained F-measures between 0.948 and 0.987, as shown in Fig. 10. By increasing the training data, the F-measure also increases. This result demonstrates that the approach works well, even at a low splitting ratio.

4.5 Performance of associated information segmentation

In order to verify the performance of associated information segmentation, we conducted two experiments for the associated information segmentation of English addresses and Chinese addresses.

We adopted Chang and Li's algorithm [13] based on tag-token pattern mining as the baseline, and we compared its performance with the FDA, as shown in Table 7. The results show that the FDA outperformed Chang and Li's algorithm, because it can handle multi-layout pages, and because it does not require exact tag patterns. The experiment confirms the feasibility of the associated information segmentation by FDA.

As in Section 4.4, Table 8 shows the average values for the precision, recall and F-measures of the associated information for Chinese addresses. Chang and Li's algorithm performed comparably with an average F-measure

[4] A ratio of 1:10 means that the quantity of testing data is 10 times that of the training data.

Fig. 12 Example query for "地球村美日語," showing the extracted addresses and their descriptions via associated information segmentation

of 0.811, whereas the FDA showed a better result, with a 0.964 F-measure, confirming that the use of a DOM tree structure overcomes the issues caused by optional data in address-associated information segmentation. The large tables used in Chinese webpages may explain why the FDA outperformed Chang and Li's algorithm for the Chinese dataset.

We further divided the above dataset into two groups in order to compare the effectiveness of the extraction of various addresses from webpages: the first group contained 128 webpages that have fewer than ten addresses, and the other group contained 122 pages with more than ten. As shown in Table 9, both algorithms work better with webpages that contain more addresses, because webpages with more information typically present that information with more regularity in order to ease the presentation of the data.

4.6 Online POI search from a smartphone

Finally, we applied the proposed address-extraction method and the associated information segmentation to build a POI search server and a client-side app, named PowerPOI (疾疾店家現身)[5] to accept user queries and provide extracted POIs on maps. Figure 11 shows addresses extracted from search snippets, given a POI query for "music studio" ("樂禾音樂"). The snippets that contain the address were used as a description for this POI. Figure 12 shows another example, for the query "地球村美日語" ("Global Village Organization") which associates each extracted address with an information block via the proposed FDA algorithm. With this POI search server, we can automatically extract POIs from search snippets and related pages to save users from tediously copying and pasting between browsers and map services.

5 Conclusions and future work

Global smartphone shipments reached 1.167 billion units in 2014, making it clear that the industry is moving ahead at full speed. Meanwhile, global mobile-data traffic grew by 69 % in 2014, nearly 30 times the size of the entire global Internet in 2000. We can expect an increase in local/POI search, given the ubiquity of mobile devices.

For POIs that can not be found on Google Maps, there is still a chance that they can be found with Google search engine. Therefore, in this paper, we proposed the idea of querying Google Search engine with user

[5] 疾疾店家現身(PowerPOI), https://play.google.com/store/apps/details?id=com.widmlab.powerpoi

keywords (e.g., the name of the POI) and extracting addresses from the search-result snippets to locate the POIs. In this paper, we applied a machine-learning-based approach for online Chinese postal-address extraction via linear-chained conditional random fields. We compared two tokenization methods (one based on individual characters, and the other based on Chinese word segmentation) and two tagging methods (one based on BIEO and the other based on IO). The results show that BIEO tagging with ICCS tokenization achieved the highest F-measure, at 0.970. In other words, word segmentation does not improve the performance.

Meanwhile, we designed an unsupervised technique for associated information segmentation from pages that contain multiple addresses. This novel idea uses the extracted addresses as landmarks and detects the Farthest Distinct Ancestor node for each landmark in a DOM tree. The average F-measure for associated information segmentation improved from 0.814 to 0.911 for English addresses, and from 0.811 to 0.964 for Chinese addresses. To the best of our knowledge, this information-extraction algorithm with the address landmarks is novel, given the other web-data extraction techniques that have been proposed [28].

We applied these two online modules to enhance POI search on maps by integrating the search results from Google Maps with those from Google search engine. In practice, the online module discovered several addresses with different locations. At present, we can only rank these addresses based on how frequently they occur. However, whether these addresses are relevant to the POI query is a crucial point for users. Meanwhile, the efficiency of the online module is also a concern. Thus, in future research, we shall construct a POI database from the Web. In doing so, the associated information represented by each corresponding address will greatly impact the performance of the retrieval.

Acknowledgments This work is partially sponsored by the Ministry of Science and Technology, Taiwan under grant 103-2221-E-008-.

References

1. Ahlers D, Boll S (2007) Location-based web search, the geospatial web. Springer, pp 55–66
2. Ahlers D, Boll S (2008) Retrieving address-based locations from the web. In: GIR, pp 27-34
3. Ahlers D (2013) Business entity retrieval and data provision for yellow pages by local search. In: ECIR
4. Asadi S, Yang G, Zhou X, Shi Y, Zhai B, Jiang W (2008) Pattern-based extraction of addresses from web page content. In: APWeb, pp 407–418
5. Baum LE, Petrie T (1966) Statistical inference for probabilistic functions of finite state Markov chains. Ann Math Stat 37(6):1554–1563
6. Borges KAV, Laender AHF, Medeiros CB, Davis CA (2007) Discovering geographic locations in web pages using urban addresses. In: GIR, pp 31–36
7. Borkar VR, Deshmukh K, Sarawagi S (2000) Automatically extracting structure from free text addresses. IEEE Data Eng Bull 23(4):27–32
8. Buttler D, Liu L, Pu C (2001) A fully automated object extraction system for the world wide web. In: ICDCS, pp 361–370
9. Cafarella MJ, Madhavan J, Halevy A (2008) Web-scale extraction of structured data. ACM SIGMOD 34(4):55–61
10. Cai W, Wang S, Jiang Q (2005) Address extraction: extraction of location-based information from the web. In: APWeb, pp 925–937
11. Chang C-H, Lui S-C (2001) IEPAD: information extraction based on pattern discovery. In: Proceedings of the 10th international conference on World Wide Web (WWW '01). ACM, New York, NY, USA, pp 681–688
12. Chang C-H, Kayed M (2006) A survey of web information extraction systems. IEEE Trans Knowl Data Eng 18:1411–1428
13. Chang C-H, Li S-Y (2010) MapMarker: extraction of postal addresses and associated information for general web pages. In: WI, pp 105–111
14. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297
15. CRF++ (2005) Yet Another CRF toolkit. Available from: http://crfpp.sourceforge.net/
16. Cunningham H, Maynard D, Bontcheva K, Tablan V (2002) GATE: a framework and graphical development environment for robust NLP tools and applications. In: Proceedings of the 40th anniversary meeting of the association for computational linguistics (ACL)
17. Laender AHF, Ribeiro-Neto BA, da Silva AS, Teixeira JS (2002) A brief survey of web data extraction tools. SIGMOD Record 31(2):84–93
18. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML, pp 282–289
19. Lin C, Zhang Q, Meng X, Liu W (2005) Postal address detection from web documents. In: WIRI, pp 40–45
20. Liu B, Grossman RL, Zhai Y (2003) Mining data records in web pages. In: SIGKDD, pp 601–606
21. McCallum A, Freitag D, Pereira F (2000) Maximum entropy Markov models for information extraction and segmentation. In: ICML, pp 591–598
22. McCallum A (2003) Efficiently inducing features of conditional random fields. In: UAI, pp 403–410
23. Nagabhushan P, Angadi SA, Anami BS (2006) A fuzzy symbolic inference system for postal address component extraction labeling. In: FSKD, pp 937–946
24. Ourioupina O (2002) Extracting geographical knowledge from the Internet. In: ICDMAM, pp 108–113
25. Pasternack J, Roth D (2009) Extracting article text from the web with maximum subsequence segmentation. In: WWW, pp 971–980
26. Raggett D (2008) HTML Tidy Library Project. Available from: http://tidy.sourceforge.net/
27. Ruzzo WL, Tompa M (1999) A linear time algorithm for finding all maximal scoring subsequences
28. Sleiman HA, Corchuelo R (2013) IEEE Trans Knowl Data Eng 25(9):1960–1981
29. Stirling G (2014) Study: 78 percent of local-mobile searches result in offline purchases, Search Engine Land, Apr. 9, 2014

30. Sutton C, McCallum A (2006) An introduction to conditional random fields for relational learning, Introduction to Statistical Relational Learning. MIT Press
31. Uryupina O (2003) Semi-supervised learning of geographical gazetteers from the internet. In: Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References, Alberta, Canada, pp 18–25
32. Yu Z (2007) High accuracy postal address extraction from web pages. Dalhousie University
33. Zhai Y, Liu B (2005) Web data extraction based on partial tree alignment. In: WWW, pp 76–85
34. Zhao H, Meng W, Yu C (2006) Automatic extraction of dynamic record sections from search engine result pages

Chia-Hui Chang is a full professor at the National Central University, Taiwan. She received a B.S. degree in computer science and information engineering from the National Taiwan University in 1993 and Ph.D. degree from the same department in 1999. Her research interests include information extraction, wrapper induction, Web intelligence, data mining, and knowledge discovery from databases. She is a board member of Taiwan Association for Artificial Intelligence (TAAI) and Association for Computational Linguistic and Chinese Language Processing (ACLCLP).

Hsiu-Min Chuang received the master's degree from National Tainan University, Taiwan in 2006. Currently, she is a PhD student in department of computer science and information engineering from National Central University, Taiwan since 2012. Her research interests includes Web mining, information extraction and bioinformatics.