Assigning Location Information to Display Individuals on a Map for Web People Search Results

Harumi Murakami[1], Yuya Takamori[1,⋆], Hiroshi Ueda[2], and Shoji Tatsumi[2]

[1] Graduate School for Creative Cities, Osaka City University,
3-3-138, Sugimoto, Sumiyoshi, Osaka 558-8585 Japan
harumi@media.osaka-cu.ac.jp
http://murakami.media.osaka-cu.ac.jp/
[2] Graduate School of Engineering, Osaka City University,
3-3-138, Sugimoto, Sumiyoshi, Osaka 558-8585 Japan

Abstract. Distinguishing people with identical names is becoming more and more important in Web search. This research aims to display person icons on a map to help users select person clusters that are separated into different people from the result of person searches on the Web. We propose a method to assign person clusters with one piece of location information. Our method is comprised of two processes: (a) extracting location candidates from Web pages and (b) assigning location information using a local search engine. Our main idea exploits search engine rankings and character distance to obtain good location information among location candidates. Experimental results revealed the usefulness of our proposed method. We also show a developed prototype system.

Keywords: location information, Web people search, map interface, character distance, information extraction.

1 Introduction

Finding information about people on the Web is one of the most popular search activities. According to [1], 30% of all queries in Web searches include person names. Person name disambiguation, or distinguishing people with identical names, is becoming more and more important in Web searches. Most research into person name disambiguation concentrates on automatically separating Web pages for different people using clustering algorithms. However, if the list of search results is merely *person 1, person 2, . . . and so on*, users have difficulty determining which person clusters they should select.

This research locates person icons on a map to provide a user-interface to help users select person clusters that are separated into different people from the result of person searches on the Web. We assign one piece of representative

⋆ Currently, Jupiter Telecommunications, Co., Ltd.

G.G. Lee et al. (Eds.): AIRS 2009, LNCS 5839, pp. 26–37, 2009.

location information to an individual. Fig. 1 shows the aim of this research and a prototype interface. When a person name is input as a query, Web pages are classified into person clusters, and individual icons that express appropriate locations are displayed on a map. Users can select icons to display their location information to access searched Web pages.

In this paper, we assign one piece of location information to person clusters.

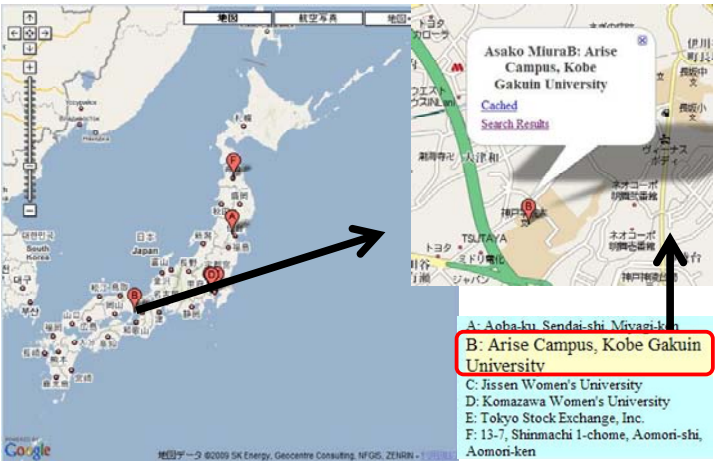

Fig. 1. Research aim

Many services extract location information from Web pages [2]. Most extract *addresses* using pattern-matching algorithms. For example, the following sequence, *number, avenue, city*, is treated as an address. These systems extract all such information from a Web page. However, existing systems have problems satisfying our requirements. First, they don't judge which address is the most suitable for particular people. Second, they don't treat multiple Web pages. Third, they cannot extract location information when addresses are not included in Web pages.

We propose a method to assign one piece of location information that is suitable for an individual using Yahoo! local search API [3] (hereafter Local search). Local search is one geocoding service that returns location coordinates based on an address or landmark query. For example, when a user inputs the following as an address query, *6-10-1, Roppongi, Minato-ku, Tokyo-to*, Local search returns *10-1, Roppongi-6-chome, Minato-ku, Tokyo-to* as an address, *35.65716694* as a latitude, and *139.73245194* as a longitude.

Our main ideas are as follows. First, we utilize landmarks as well as addresses to extract location information even from Web pages without addresses. Second, we use search engine rankings and character distance to assign suitable location information. Third, we use Local search to convert texts included in Web pages

to location information. To obtain one piece of location information from Local search, we design several mechanisms. For the zero hit problem, we introduce two heuristics: a one-character deleting heuristic when the text is an address type, and a formal name inference heuristic when the text is a landmark type. A candidate list is another mechanism to avoid zero hits. For the multiple hits problem, we introduce a calculating context heuristic to select one result.

Below, in Section 2 we explain our method. The experimental results are described in Section 3. We discuss our method's usefulness and related work in Section 4. The examples in this paper were translated from Japanese into English for publication.

2 Our Approach

This research locates person icons on a map to provide user-interfaces to help users select person clusters that are separated into different people from the result of person searches on the Web. We obtain the following set of location information for person clusters: *a location label, a location address, and a latitude and longitude pair.*

Table 1 shows the example results of this research. When a person cluster is input, location information is obtained by Local search.

Table 1. Example results

Person cluster	Location Information		
	Location label	Location address	Lat/Lon
Asako Miura 3 (B)	Arise Campus, Kobe Gakuin University (landmark type)	Nishi-ku, Kobe-shi, Hyogo-ken	34.6.../135.0...
Asako Miura 11 (F)	13-17, Shinmachi-1-chome, Aomori-shi, Aomori-ken (address type)	13-17, Shinmachi-1-chome, Aomori-shi, Aomori-ken	40.8.../140.7...

For example, for the first cluster, Asako Miura 3 (associate professor at Kobe Gakuin University), a location label, *Arise Campus, Kobe Gakuin University*, a location address, *Nishi-ku, Kobe-shi, Hyogo-ken*, and a latitude and longitude set, *34.6.../135.0...* are obtained. This location information is a landmark type. For the second cluster, Asako Miura 11, the location label and the location address are *13-17, Shinmachi-1-chome, Aomori-shi, Aomori-ken*, and the location address is *40.8.../140....* This is an address type.

A method overview is shown in Fig. 2. Our method is comprised of two steps: (a) extracting location candidates from Web pages and (b) assigning location information using Local search.

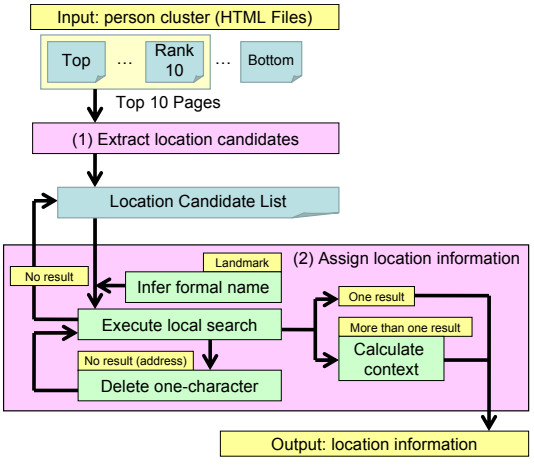

Fig. 2. Overview

2.1 Extracting Location Candidates

When a person cluster (HTML files obtained from person searches) is given, location candidates are extracted and a location candidate list is generated.

Determining web pages using search engine ranking. One difficult problem in this research is identifying which information is the most suitable for the designated person. If we only use one Web page, one very good piece of information may be obtained, but much good information may be overlooked. If we use all Web pages, all information might be obtained, but selecting the best information may become unwieldy. In this research we need to examine which Web pages should be treated. We propose using search engine rankings to determine Web pages to extract location information. We decided to use the top 10 Web pages to extract location information based on the experimental results in Section 3.

Extracting address and landmark candidates. When Web pages are given, after removing HTML tags, new line codes, and spaces from HTML files, we extract address and landmark candidates based on heuristics and morphological analysis. We used MeCab [4] for morphological analysis. Two heuristics are described below:

1. Extracting address candidate heuristic
   When a morphone that meets conditions (a) or (b) appears continuously more than once, we combine these morphones into a location candidate (address type): (a) whose type is judged *location, prefix*, or *number* by MeCab (b) which is included in the predefined term list such as: *hyphen, street, avenue, north, south, east*, or *west*.

For example, *3-3-138, Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka-fu*, is extracted as an address type candidate.

2. Extracting landmark candidate heuristic
   We extract a morphone whose type is judged *organization* as a location candidate (landmark type).
   For example, *City Hall of Kyoto City, Osaka Station, The University of Tokyo* are extracted as landmark types.

Again, one difficult problem in this research is identifying which information is the most suitable for the designated person. Our solution uses the character distance between location candidates and the designated person name. Character distance is number of characters (including spaces) between location candidates and the designated person names. For example, an original Web page contains *Asako Miura (Kobe Gakuin University)* and the distance between a location candidate *Kobe Gakuin University*, and person name *Asako Miura* becomes 2. Address candidates whose character distance is less than 71 and landmark candidates whose character distance is less than 31 are extracted from the top 10 Web pages.

Sorting location candidates. To avoid zero hits from a Local search, we order extracted location candidates by character distance to generate a location candidate list. When the character distance between two candidates is identical, those extracted from higher ranked pages become higher, and those extracted from the upper part of pages become higher.

2.2 Assigning Location Information

When a location candidate list is given, our method obtains one piece of location information using Local search. First, our method gives a location candidate from the top of the candidate list to Local search. When the candidate is a landmark type, a formal name inference heuristic is processed to change the abbreviation into a formal name to hit a Local search beforehand.

When only one result is obtained from Local search, it becomes the answer. When more than one result is obtained, our method calculates the similarity between the candidate and obtained results, and the most similar result becomes the answer. We call this a calculating context heuristic. When the candidate is an address type and no result is obtained, the one-character deleting heuristic is applied to modify the candidate to hit a Local search. When no result is obtained after these processes, the next candidate in the list will be processed. When there is no answer after all candidates in the list are processed, *none* (*no location information*) is the output.

Formal name inference heuristic. When a location candidate is a landmark type, a formal name inference heuristic is applied once to change the abbreviations into formal names to get the Local search results. The formal name

inference heuristic gives the candidate to the Yahoo! Web search API [5] (hereafter *Web search*) and gets the title of the first result. The title is modified to a formal name using a heuristic based on a stop list.

For example, when the candidate is *Todai* (abbreviation of *The University of Tokyo*), it is given to a Web search, and the title of the first result is *The University of Tokyo Homepage*. After *Homepage* is deleted using a stop list based heuristic, the candidate becomes *The University of Tokyo*.

One-character deleting heuristic. When no result is obtained for the address type candidate, the one-character deleting heuristic is repeatedly executed to delete the last character of the candidate and to repeatedly give it to Local search until the following stop condition is reached. The heuristic stops when the string no longer contains *ku* (town), *shi* (city), *to, do, fu*, or *ken* (these four terms denote prefectures).

For example, if address query *6-10-1, Roppongi, Minato-ku, Tokyo-to* obtains no result from a Local search, the next query becomes *6-10, Roppongi, Minato-ku, Tokyo-to*. Then the next query becomes *6, Roppongi, Minato-ku, Tokyo-to*.

Calculating context heuristic. When more than one result is obtained from a Local search, our method calculates context with a calculating context heuristic that calculates the similarity between the location candidate and the obtained results using vector space models. The most similar result becomes the answer.

For example, when the location candidate is *The University of Tokyo*, it is given to a Local search, and such multiple results are obtained as 1) The University of Tokyo, 2) The University of Tokyo Komaba Campus, 3) The University of Tokyo Cultural Development, and so on. We need to select a result. Our method can select an answer based on the Web page context. For example, if a person teaches at the main campus of The University of Tokyo, 1) is selected; if s/he works at the Komaba campus, 2) is selected.

The similarity between the candidate and the results is calculated using a cosine measure, based on a vector space model. Given candidate vector c and results vector r, similarity $sim(c, r_i)$ is defined as follows:

$$sim(c, r_i) = \frac{\sum_{j=1}^{t} w_{cj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{cj})^2} \sqrt{\sum_{j=1}^{t} (w_{ij})^2}} \quad (1)$$

Here t is the number of terms, w_{cj} is the weight of t_j in candidate vector c, and w_{ij} is the weight of t_j in result vector r. The terms are defined as the following morphones: (a) those included in the candidate list (when the candidate is an address type), (b) those included in the original Web page with a location candidate (when the candidate is a landmark type), and (c) those included in the Local search results. The weights of the terms are calculated by tf-idf. The result that is most similar and larger than 0.7 becomes the answer. When no result is larger than 0.7, the top result becomes the answer.

2.3 Example

There are 14 people for query *Asako Miura*. For the five people (Person clusters 2, 3, 6, 8, 11) with location information in Web pages, 100% (5/5) was correct. For nine people without location information in Web pages, no location information was assigned for eight people. In other words, 88.9% (8/9) was correct.

Asako Miura 3 is currently an associate professor at *Kobe Gakuin University*. There are several Kobe Gakuin University campuses, and our method outputs the correct answer, *Arise Campus, Kobe Gakuin University*, where her office is located. This good example shows the usefulness of our method. If we use all (i.e. 100) Web pages, *the Faculty of Human Science, Osaka University* will be displayed. If we use the frequency method (see Section 3.1), *the Faculty of Human Science, Osaka University* will also become the answer. Osaka University is her previous affiliation where she worked for many years. *Kyushu University*, which she occasionally visits for conferences, was the output from 15 pages. In this case, Kyushu University had the shortest character distance. No answer was provided when using the address method (see Section 3.1) because the Web pages had no address information.

Fig. 1 is an example interface. Icons indicating each person on a map and a list of location labels are displayed. Asako Miura 3 becomes Asako Miura B because person clusters with no location information assigned are not displayed on a map. When a user selects an icon or a list, information about the person is displayed with the location label: *Arise Campus, Kobe Gakuin University*. The user can display the original Web page that includes Kobe Gakuin University (cached) and the search results.

3 Evaluation

3.1 Method

Dataset. The twenty person names used in related work [6] were selected as queries. 100 HTML files were obtained for all 20 queries from Web searches [5]. Two subjects manually classified these Web pages into different people. 151 people were found in all 100 Web pages. For 13 person names, different people existed, and for seven person names, only one person existed.

The subjects extracted the location information from the person clusters by checking all Web pages. For 79 out of 151 people (52.3%), location information existed, and there was no appropriate location information for 72 people. The average number of Web pages for the former 79 people was 21.3 (SD=35.8) and 4.1 for the latter 72 (SD=14.6).

The evaluation measurement was as follows. First, current places (offices or homes) were judged as appropriate location information. When there was no current place, past places (offices or homes) were treated as appropriate location information. Places the person temporarily visited do not become location information. An address type and a landmark type are equally satisfactory.

Comparative methods. To evaluate the usefulness of search engine rankings and to determine the number of Web pages, we examined the number of Web pages (the top, top five, 10, 15, and 100). We call these methods: (a) top page, (b) top five pages, (c) top 10 pages (our method), (d) top 15 pages, and (e) top 100 pages.

To evaluate the usefulness of a combination of search engine rankings, landmarks, and character distance, we examined a method that only extracts addresses and a method that uses frequency to calculate location information. We used the top 10 pages because the former evaluation revealed that 10 pages was the best. We call the former (f) the address method (top 10 pages and only extracting addressess) and the latter (g) the frequency method (top 10 pages and using frequency).

Evaluation. Two subjects evaluated the assigned location information by the above methods. Precision, recall, F-measure, and total were calculated as follows:

$$precision = \frac{\text{number of people whose assigned LI is correct}}{\text{number of people whose LI is assigned}}$$
$$recall = \frac{\text{number of people whose assigned LI is correct}}{\text{number of people whose LI is included in Web pages}}$$
$$F - measure = \frac{2 \times precision \times recall}{precision + recall}$$
$$total = \frac{\text{number of correct answers}}{\text{number of people}}$$

With Local search, an assigned address and an address included in an original Web page sometimes differ. For example, when *3-3-138, Sugimoto-cho, Sumiyoshi-ku, Osaka-shi* is included in a Web page and Local search returns *Sugimoto 3-Chome, Sumiyoshi-ku, Osaka-shi*, we judged this correct. Another such example is *Fukuoka City*, which is included in a Web page; Local search returns an address of *city hall of Fukuoka City*.

3.2 Results and Discussion

Effect of search engine ranking. Fig. 3 shows precision, recall, F-measure, and total when the number of Web pages varies. Top 10 pages (our method) performed best in recall, F-measure, and total. The top page result was particularly inferior in recall due to the lack of information from only one page. The result of top 100 pages was also inferior on average for the following reasons: (a) location candidates included in lower ranked Web pages tend to be unrelated to designated people, and (b) the possibility of incorrect answers increases as the number of pages increases. The above results show the usefulness of our method to use top 10 Web pages as sources to extract location information.

Effect of landmarks and character distance. Fig. 4 compares other methods when the Web pages are in the top 10. Our method greatly outperforms the other methods in all evaluations. Comparison with the address method shows our method's usefulness using landmarks to extract useful information to assign location information. Comparison with the frequency method shows our method's usefulness to use character distance for person names rather than frequency.

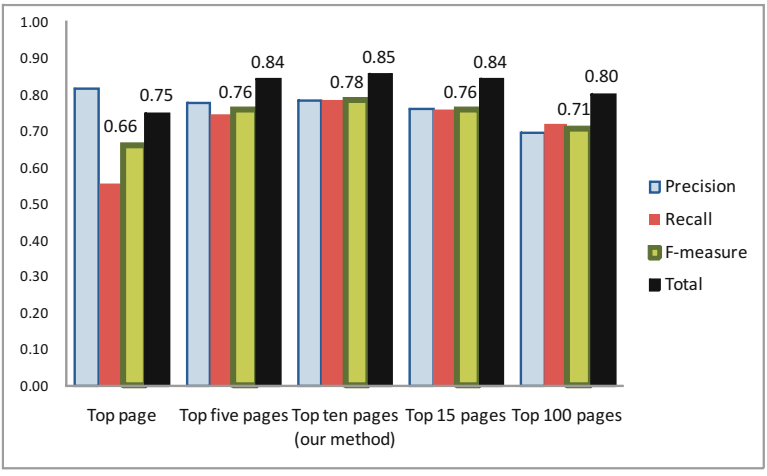

Fig. 3. Effect of search engine rankings

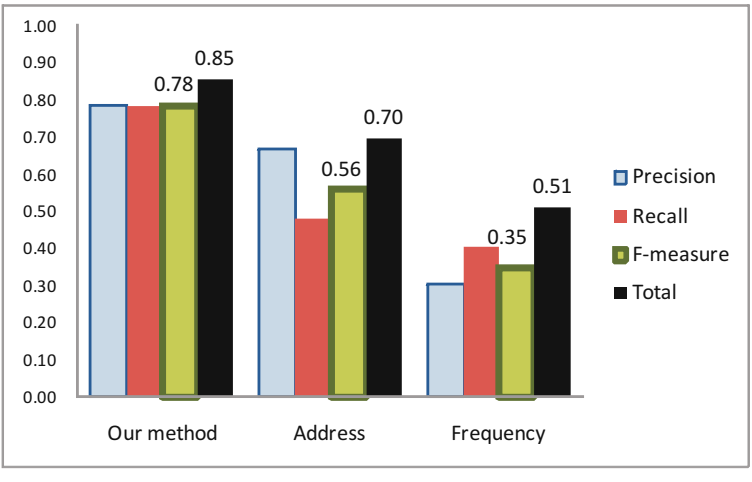

Fig. 4. Comparison with other methods

Answer analysis. We analyze the answers in detail. Out of 62 correct answers for assigned location information, 40 (64.5%) were landmark types and 22 (35.5%) were address types. This result suggests the usefulness of using landmarks. For 40 landmarks, 19 (47.5%) were universities, 6 (15.0%) were prefectural governments, 6 (15.0%) were train stations, 3 (7.5%) were high schools, and 6 (15.0%) were other. Most were public institutions. For people with answers, 12 were incorrect for assigned location information. 91.2% (11/12) chose incorrect candidates by character distance, 8.3% (1/12) chose incorrect results from Local search by calculating similarity. Inside the former 91.2% error, target candidates

were extracted, but other candidates were chosen by character distance (91.0%, 10/11) and target candidates could not be extracted (9.0%, 1/11). A typical extraction error was that publishers were assigned because people with many pages often wrote books or were discussed in books. Overall, the main reason for errors included problems of (a) character distance and (b) extracting landmarks.

4 Related Work and Discussion

Wan et al. [7] also separated Web people search results and assigned titles to person clusters. We assign location information to display person icons on a map. Even though much work (e.g., [8], [9], [10]) separates Web pages into person clusters, it seldom assigns labels to person clusters.

The WWW9 WePS-2 workshop [11] evaluated a technique to extract attribute information. This task extracts 18 kinds of attribute values for target individuals whose names appear on each of the provided Web pages. However location is not included in the evaluation due to its ambiguity.

Our work is related to such clustering search engines as Clusty [12] which usually assigns keywords or phrases to a Web page cluster to help users select a cluster based on such information as term frequency and URLs.

Google's alternative to search results [13] displays multiple location information included in Web search results as an experimental service. It does not display one piece of representative location information. It mainly extracts city names and sometimes college names. We extract addresses (including city names) and landmarks (including college names).

Most existing systems and research extract multiple addresses contained in Web pages. Few systems extract landmarks. [14] extracted addresses, postal codes, and phone numbers contained in Web pages and converts them into coordinates. It does not extract landmarks, and we do not extract phone numbers. The discrepancy reflects different system aims. [14] gathered as much location information as possible, but we aim to output one good piece of location information for many people.

Morimoto et al. [15] extracted address pairs and descriptive text from Web pages and displays them on a map using a HTML structure. Much research utilizes HTML structure to limit the range of documents to judge the relationship between an object and an address. Instead of using HTML structure, we exploit character distance to judge the strength of the relationships. Our heuristics for extracting addresses are related to [16] because they use a Japanese morphological analysis and utilize location attached to nouns. [16] does not extract landmarks.

Many Web sites provide a facility that converts addresses or landmarks into coordinates with geocoding services including a Local search using simple pattern-matching. They return the first result (coordinates) based on an exact match. For example, when a user inputs *Hanshidai* (abbreviation of Osaka City University) it may fail because no Hanshidai exists in the geocoding databases. Our method copes with pattern-matching problems in three steps. First, a formal

name inference heuristic (landmark type) and a one-character deleting heuristic (address type) are applied. Second, we use a candidate list to try another location candidate to get a result. Third, we apply a calculating context heuristic to identify one piece of location information from multiple results.

One advantage of our method is that no special location dictionary is needed because such existing tools and services as morphological analysis (MeCab), Yahoo! local search, and web search APIs are combined. A formal name inference heuristic, a one-character deleting heuristic, and a calculating context heuristic are powerful enough to utilize Local search to get one result. Our algorithm is simple and easy to implement. Although our algorithms are heuristic based, we do not need to modify or add new heuristics; therefore systems are easy to maintain.

The experimental results revealed that our method has the best recall, F-measure, and total performance among comparative methods. The above results suggest our method's usefulness. In addition, we successfully built a prototype interface to select a person cluster on a map. Our method is only inferior in precision to the top page method. If we need to build an interface with fewer incorrect icons (regardless how few), the top page method can be selected.

The limitations of our evaluation include the following. Since our evaluation is limited to Japanese, the proposed heuristics should be adjusted to other languages. We extracted addresses and landmarks, except for those addresses and landmarks that are not contained in the MeCab dictionary and the Local search database.

Future work includes the following. First, we need to improve the extraction of location candidates. For example, we should remove publishers by using a stop list. Second, new algorithms should be investigated to cope with errors caused by character distances. One possible solution may be to combine character distances with term-frequency, HTML structure, and/or syntactic analysis.

5 Conclusions

We proposed a method to assign person clusters with one piece of *representative* location information to display person icons on a map. The following are the main ideas of our method: (1) using landmarks as well as addresses to extract location information, (2) using search engine rankings and character distance to assign suitable location information, and (3) using Local search to convert texts included in Web pages to location information.

The experimental results revealed that our method has the best recall, F-measure, and total performance among comparative methods. The above results suggest our method's usefulness. In addition, we successfully built a prototype interface to select a person cluster on a map.

References

1. Guha, R., Garg, A.: Disambiguating people in search. Stanford University (2004)
2. Okilab.jp projects, `http://okilab.jp/project/location/`

3. Yahoo! Local search API, http://developer.yahoo.co.jp/webapi/map/localsearch/v1/localsearch.html
4. MeCab, http://mecab.sourceforge.net/
5. Yahoo! Web search API, http://developer.yahoo.co.jp/webapi/search/websearch/v1/websearch.html
6. Sato, S., Kazama, K., Fukuda, K.: Distinguishing between People on the Web with the Same First and Last Name by Real-world Oriented Web Mining. IPSJ Transactions on Databases 46(8), 26–36 (2005)
7. Wan, X., Gao, J., Li, M., Ding, G.: Person Resolution in Person Search Results: WebHawk. In: CIKM 2005. Proceedings of the Fourteenth ACM Conference on Information and Knowledge Management, pp. 163–170 (2005)
8. Bekkerman, R., McCallum, A.: Disambiguating Web Appearances of People in a Social Network. In: WWW 2005, Proceedings of the Fourteenth World Wide Web Conference, pp. 463–470 (2005)
9. Kozareva, Z., Moraliyski, R., Dias, G.: Web People Search with Domain Ranking, Text, Speech, and Dialogue. In: LNCS, pp. 133–140. Springer, Heidelberg (2008)
10. Artiles, J., Gonzalo, J., Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. In: Proceedings of the Fourth International Workshop on Semantic Evaluations, pp. 64–69 (2007)
11. Task Definition of Attribute Extraction Subtask for WePS-2, http://nlp.uned.es/weps/weps2/WePS2_Attribute_Extraction.pdf
12. Clusty the clustering search engine, http://clusty.com/
13. Google Experimental Search, http://www.google.com/experimental/
14. McCurley, K.S.: Geospatial Mapping and Navigation on the Web. In: WWW 2001, pp. 221–229 (2001)
15. Morimoto, H., Fujimoto, N., Nagaya, T., Idehara, H., Hagihara, K.: A System for Web Retrieval of Address-Related Information. IEICE Trans. D J90-D(2), 245–256 (2007)
16. Arai, I., Kawaguchi, Y., Fujikawa, K., Sunahara, H.: Geocrawler; Web Indexer for Store Search based on Geographical Information and Evaluation Information on Personal Web Sites. IPSJ Journal 48(7), 2319–2327 (2007)