Conference Information Management System : Towards a Personal Assistant System

MINE, Tsunenori[1], AMAMIYA, Makoto[1] and MITAMURA, Teruko[2]

[1] Kyushu University, 6-1 Kasugakouen, Kasuga 816-8580, JAPAN,
E-mail: mine@is.kyushu-u.ac.jp,
WWW home page: http://www-al.is.kyushu-u.ac.jp/~mine
[2] Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
E-mail: teruko@cs.cmu.edu,
WWW home page: http://www.cs.cmu.edu/~teruko

Abstract. We are aiming to develop a personal assistant system which handles its user's files stored in his/her computers and his/her interesting information that can be accessed on the Web. The system categorizes the files and Web pages gathered and extracts information specified by him/her and stores it into a structured database for use as the knowledge of its dialogue module. This paper presents a prototype of the system that handles information about conferences of interest to the user. The system extracts conference names, submission deadlines, dates, URIs and locations from e-mail messages the user has received, and web pages gathered by both crawling and meta-search. Extracted information is stored into a database so that the user can interactively search conference information via a user interface with natural language queries.

1 Introduction

After the concept of the WWW was first proposed in 1989, the number of Web pages grew to exceed 800 million pages by Feb, 1999[1]. They provide manifold services and are one of the most important resources accessed easily in the world. Among their useful services, the Internet search engines are playing the important role to find the information users want. Although they try to provide useful and relevant search results to their users, they still return a lot of non-relevant results because of the lack of information to convey user's intention, part of which coming from the isolation between search and evaluation process[5]. Therefore, it is important to make use of user's responses as feedback to his/her own search process. In addition to retrieving relevant web pages, extracting and storing frequently accessed information in the database is useful for reducing the search time, and enabling the information to be used as new knowledge for both human and agents.

In this paper, we present a prototype of our personal assistant system which handles the information about conferences of interest. It consists of the following modules: Information Retrieval & Web page gathering(**IRWG**), Information Filtering(**IF**), Information Extraction(**IE**), DataBase Manager(**DBM**), and User

Interface(**UI**) that analyzes user's natural language query and learns user's preference and interests. The **IE** module extracts conference names, deadlines, dates, URIs, and locations from e-mail messages the user received and web pages gathered by both crawling and meta-search. Extracted information is stored into a database so that the user can search it interactively via the **UI** with a natural language query.

2 Conference Information Management System

2.1 Overview of Our Prototype System

The left side of figure 1 shows the overview of a prototype system. **IRWG** performs meta-search and polling of some interesting sites that are often visited by a user, and the sites included in the conference announcement messages the user received by e-mail. It traverses the relevant pages linked from those sites as a base-point. **IF** divides the relevant information and non-relevant information to the user with SVM^{light}[2]. **IE** extracts a set of named entities specified at each category. The extraction rules consist of pattern matching rules and detection rules. The pattern matching rules are hand-coded as regular expressions and select the candidates for named entities. The detection rules are constructed from the N-gram statistics of negative data, which are results extracted and detected incorrectly. The detection rules score each candidate named entity to be extracted and detect its boundary words. The detection rules are learned with negative feedback data. **DBM** stores the set of information extracted by the **IE** module into a structured database and retrieves information specified by the user's query from the database. In the interactive mode, **DBM** accepts only the information confirmed by a user. **UI** is the interface between a user and other modules. **UI** analyzes the user's natural language query and transforms it into an S expression format. **UI** talks to other modules using a sentence in S expression format. The right side of figure 1 shows the data flow between modules in the case where a user performs a Web search when s/he isn't satisfied with the answer returned from the **DBM**. **UI** learns user's preference related to concepts specified for each category. In the case of the conference category, these are topics, conference names, locations, dates and so on. Such preference cues are recorded in the user profile(**UP**). The **UP** is shared by not only **UI** but also **IRWG** so that it searches Web pages about conferences that the user likes, provided that it is maintained only by the **UI**. In the current system, **UI** is implemented with Java except for morphological analyzer, and other modules are written in perl. The rest of this section focuses on the **UI** and **IE** module.

2.2 User Interface(UI)

The **UI** is one of the most important parts of this system. **UI** performs some simple natural language processing: morphological analysis, word expansion with a dictionary and pattern matching with rules predefined for each category. **UI** transforms an analyzed query into an S expression format, which has the following format: **(category (attribute value) .. (attribute value))**

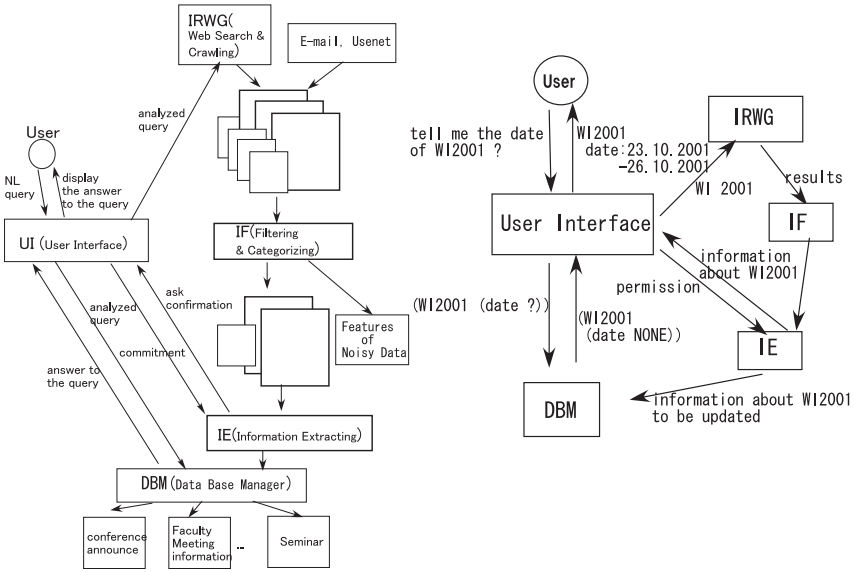

Fig. 1. The overview of the system and the data flow between modules in the Web searching process

When a query asks the value of an attribute, its value slot is replaced with a question mark **?**. If a query asks the name of a category or attribute X, X has a mark **:?**. If a query asks whether or not the value X is correct, X has a mark **:OK?** and is expressed as X:OK?. The following shows examples that transform natural language queries into S expressions.

Where is the location of the conference ? → (conference (location ?))
When is the conference held ? → (conference (date ?))
What is the topic of the conference ? → (conference (topic ?))
What are the conferences whose topic includes NLP ? → (conference:? (topic NLP))
Is the location of the conference Maebashi ? → (WI2001 (location Maebashi:OK?))

User Profile(UP): UP consists of 3 fields: category, attribute, and example. The category 'conference' includes 9 attributes: conference name, topic, location, date, URI, deadline, speaker, organizer and organization. The example field of topic attribute includes a set of ('words', value) pairs such as: (Information Retrieval,10) (Information Extraction,10) (Natural Language Processing,18) (Machine Translation,12). Those examples of 'words' are extracted from user's queries and manually input by the user. The value of 'words' is the number of occurrences that the 'words' have appeared in queries. Those examples are used to rank the results in reply to a user's query. The relevance value of a retrieved document is calculated by summing all weights of 'words', which are included in both the document and in the example field of **UP**. The weight of 'words' is calculated by dividing their value by the total of the values of all words in the example filed of the attribute.

Interactions between User and UI, and between Modules: The resulting replies to a user's query first comes from **DBM** via **UI**. When s/he is not satisfied with them, s/he can ask **UI** to perform re-searching of relative pages on the Web. In that case, **UI** passes his/her query to **IRWG**. After searching Web pages relevant to the query, **IRWG** sends the results to **IF**. **IF** filters and categorizes them, and sends the results to **IE**. After extracting named entities specified in the category of the results, **IE** sends them to **UI** so that **UI** will get the user's permission to store the results in the database. If s/he is satisfied with the results from **IE**, s/he gives the permission to **IE** via **UI**. After receiving the permission, **IE** sends them to **DBM**(See the right side of figure 1). In case s/he is not satisfied with the results extracted by **IE**, s/he can ask **UI** to show the pages filtered by **IF** or retrieved by **IRWG**.

Displaying Results Returned from DBM: The **UI** module displays the information with two formats: calendar format and simple string format. The calendar format is used for listing up many results retrieved from the database, and the string format is for returning a simple answer. For example, when **DBM** extracts all information about conferences held in 1999 and 2000 in order to reply the query (conference (date *.*.1999) (date *.*.2000)), which are transformed from a natural language query such as *Show conferences to be held in 1999 and 2000*, **UI** shows the results with a calendar format (see figure 2).

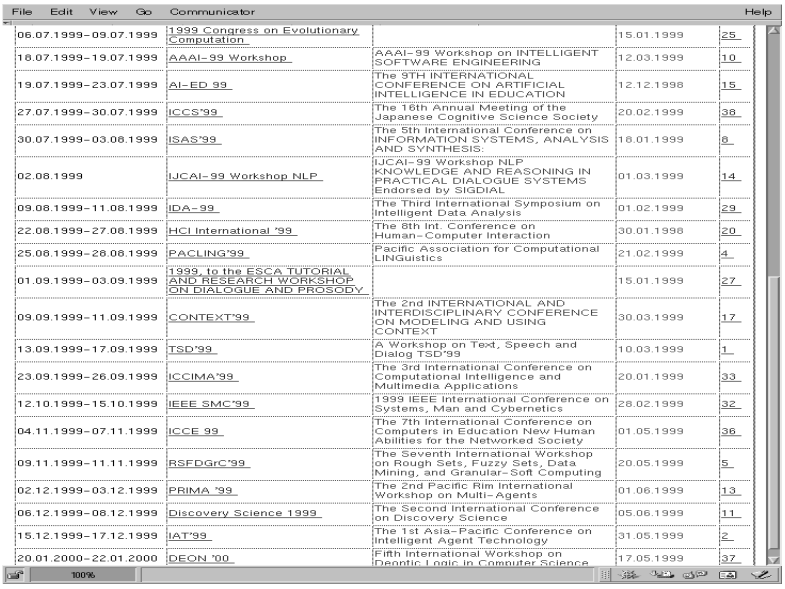

Fig. 2. An example displayed with a calendar format

2.3 Extraction Module

The extraction rules for matching named entities, such as conference names, are hand-coded with perl's regular expressions. The detection rules for named enti-

ties are constructed from the N-gram statistics of negative data. The number of extraction rules for conference names is currently 29. Before applying extraction rules, the extraction module searches for areas that are candidates for a named entity. That search is performed with heuristic rules constructed from 50 conference announcement articles. The extraction procedure for named entities is performed as follows:

1. apply pattern matching rules that consist of regular expressions.
2. apply detection rules that consist of the difference between positive and negative N-gram data. Positive data are constructed from the labeled data of conference name and negative data are results returned incorrectly.
3. rank the detected candidates.

Preliminary Experiments of Conference Name Extraction To evaluate the current extraction performance with pattern matching rules, we first used web page data including the keywords 'CALL for Paper' or 'CFP'. The number of raw data pages is 482. After manually checking the data, the number of conference announcements is 141. Among them, the extraction module extracted and detected 85 conference names correctly or partly correctly(60.3%). This result was gotten without any heuristic rules to account for the structure of HTML tags. Next, we used 250 conference announcement e-mail articles to show the effectness of detection rules. Detection rules were constructed from the experimental results without detection rule.

	correct	partly correct	wrong	correctness(w. partly correct)
No detection rule	123	106	21	49.2 % (91.6 %)
detection rule	144	75	31	57.6 % (87.6 %)

These results indicate that 1) we need to add rules that handle the HTML tag structure to get a higher extraction rate for Web pages, 2) detection rules constructed from negative data are effective to detect boundaries of conference names, but also cause more errors. We need to investigate the relationships between positive data and negative data to improve detection rules.

3 Related Works

The related works are user adaptive information retrieval, information extraction, information filtering and recommendation system.

The DeadLiner[6] is a specialized search engine that handles conference announcement. It monitors specified web pages, news groups and E-mail articles widely distributed on the Internet. The DeadLiner has been been constructed for academic researchers, while we aim to construct our system as a personal assistant system that helps the user search, recommend and digest the information the user wants. WebKB(e.g. [4]) project has been trying to construct machine readable knowledge bases from the WWW. It is really challenging research. However, the WebKB system only focuses on building knowledge bases specified by users without any interaction between the users and the system, although the knowledge should be interactively updated. A lot of user adaptive search engines have been proposed (e.g. [8],[3],[7],[5]). They infer users' preference and interests from their retrieval history and browsing behaviors to search the next web page

to be recommended with making use of them. Most of the systems unfortunately only focus on retrieval and not the extraction of named entities the users find interesting. In order to realize a personal assistant system, it is indispensable to integrate retrieval and extraction via an interactive interface.

4 Conclusion

In this paper, we presented a conference information management system that learns a user's interests and preferences from his/her queries via a **UI** module. The system has also the function of managing conference announcement messages distributed by e-mail to the user and gathered from web pages the user wants. The system extracts conference name, topic, location, paper submission deadline, URL and date, and stores them into a structured database. If the extraction is performed in the interactive retrieving process between the user and **IRWG** module, the extracted results are passed to the user to get his/her confirmation before passing them to **DBM**. We are currently working to realize practical extraction performance on conference names, to merge more than one page that indicates the same conference announcement, and to handle different formats like Web page and e-mail articles. The latter challenge is not easy but crucial to realize a useful personal assistant system. After that, we plan to extend our system so that it handles other announcement types such as local meetings, seminars and lectures. In the future, this system will be extended to be a more robust question and answering system for conference information management.

References

1. S. Lawrence and C. L. Giles. Accessibility of Information on the Web, Nature, 400(6740), pp. 107–109, 1999.
2. T. Joachims. SVM^{light} version 3.50 `http://ais.gmd.de/\~{}thorsten/svm_light/` 2000.
3. M. Balabanovic. An adaptive web page recommendation service. In W. Lewis Johnson and B. Hayes-Roth, editors, *Proceedings of the First International Conference on Autonomous Agents (Agents'97)*, pp. 378–385, 1997.
4. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigan, and S. Slattery. Learning to construct knowledge bases from the world wide web. *Artificial Intelligence*, 118(1–2):69–113, 2000.
5. T. Helmy, T. Mine, and M. Amamiya. Adaptive exploiting user profile and interpretation policy for searching and browsing the web on kodama system. In *Proceedings of the 2nd International Workshop on Natural Language and Information Systems(NLIS 2000)*, pp. 120–124, 2000.
6. A. Kruger, C. L. Giles, F. Coetzee, E. Glover, G. Flake, S. Lawrence, and C. Omlin. DEADLINER: Building a new niche search engine. In *Ninth International Conference on Information and Knowledge Management, CIKM 2000*, pp. 272–281, 2000.
7. H. Lieberman. Letizia: An agent that assists web browsing. In Chris S. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pp. 924–929. 1995.
8. M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & webert: Identifying interesting web sites. In *Proc. of 13th National Conference on Artificial Intelligence(AAAI-96)*, 1996.