Personal Navigating Agent

H. L. Wang[1], W. K. Shih[1],
C. N. Hsu[2], Y. S. Chen[2], Y. L. Wang[2] and W. L. Hsu[2,3]

Abstract

The World Wide Web provides a huge distributed web database. However, information in the web database is free formatted and unorganized. Traditional keyword-based retrieval approaches are no longer appropriate. In this paper, we consider a framework for constructing agents that can simulate the behavior of human browsing on the Internet. Given a specific target, such an agent will make use of existing search engines to navigate through the web to locate the sites containing the target information and extract them into a database. We refer to these types of agents as *Personal Navigating Agents* (*PNA*). Since the information service is domain specific, we shall first focus on those *PNA* that can retrieve people's information on the web in this paper. In this particular experiment, given the name of a university, we shall extract the following information about its faculty: name, telephone number, fax number, email address and *URL*. We explore web page knowledge in two ways: First, we develop a tagging system for each web page to facilitate information extraction. Our tagging system employs an *HTML* parser together with a natural language semantic tagger. These semantic tags are more general than part-of-speech tags used in linguistics. Second, we equip our *PNA* with a navigation map. A navigation map will guide our *PNA* to traverse through related pages and to arrive at pages containing the target information. In our experiments, our prototype agents have successfully explored a university web site and extracted target information with a very high accuracy.

1. Introduction

The World Wide Web provides a new concept for information distribution. Web servers around the world connected through the Internet become a distributed database, which we refer to as *web database*. Using a browser, anyone can quickly access this huge database. Nowadays, virtually any ordinary keyword will extract a large number of web pages. Unfortunately, information in web database is free

[1] Department of Computer Science, National Tsing Hua University, Hsin-Chu, Taiwan, R.O.C.
[2] Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C.
[3] hsu@iis.sinica.edu.tw

formatted and unorganized since people are very creative in designing their home pages. Traditional approaches such as database retrieval and keyword search are no longer appropriate. Semantics-based *IR* becomes increasingly important.

In this paper, we consider a framework for constructing agents that can simulate the behavior of human browsing on the Internet. Given a specific target, such an agent will make use of existing search engines to navigate through the web to locate the sites containing the target information and extract them into a database. We refer to these types of agents as *Personal Navigating Agents* (*PNA*). Since information service is domain specific, we shall first focus on those *PNA* that can retrieve people's information on the web in this paper. In this particular experiment, given the name of a university, we shall extract the following information about its faculty: name, telephone number, fax number, email address and *URL*. The extracted information will soon be expanded to include title, affiliation and mailing address. The techniques developed are general enough for constructing *PNA* that can perform many other tasks.

Many "spider agents" have been deployed to collect *URLs* or E-mail addresses. But due to the inability to understand and reason about web page contents, it is still difficult for them to constrain their exploration on the web, nor can they assemble collected pieces of raw data into useful information. Research in information extraction from semi-structured web pages [5][9][10] and free text [1][3] only concentrate on extract information from a single Web page. They do not address the problem of selectively retrieving and assembling relevant pieces of data from relevant web pages. Recently, the "World Wide Knowledge Base" project at Carnegie Mellon University [2][4] attempts to solve this problem by learning relational rules from training web pages in a target domain. Our approach differs from theirs in that the knowledge used by our agent is obtained from analyzing a large domain-independent natural language corpus. The knowledge is expressed in terms of probabilistic semantic templates and navigation maps, which guide our agent to recognize which hyperlink should be explored next, and which string should be extracted.

Since information in *HTML* documents is presented in many different forms, information extraction (*IE*) from web pages requires many different techniques. We propose to develop a tagging system (*TS*) for web pages to facilitate *IE*. Our *TS* employs an *HTML* parser together with an *NL* semantic tagger. These semantic tags are more general than part-of-speech tags used in linguistics. The semantic tags give the underlying semantic meaning of part of texts (usually a phrase) by its WH features such as who, what, when, where, why and how. The kernel of our *NL* understanding system is designed based on that of *GOING* [7], which will be described in more detail in Section 3. Currently, our *NL* tools can process both Chinese and English.

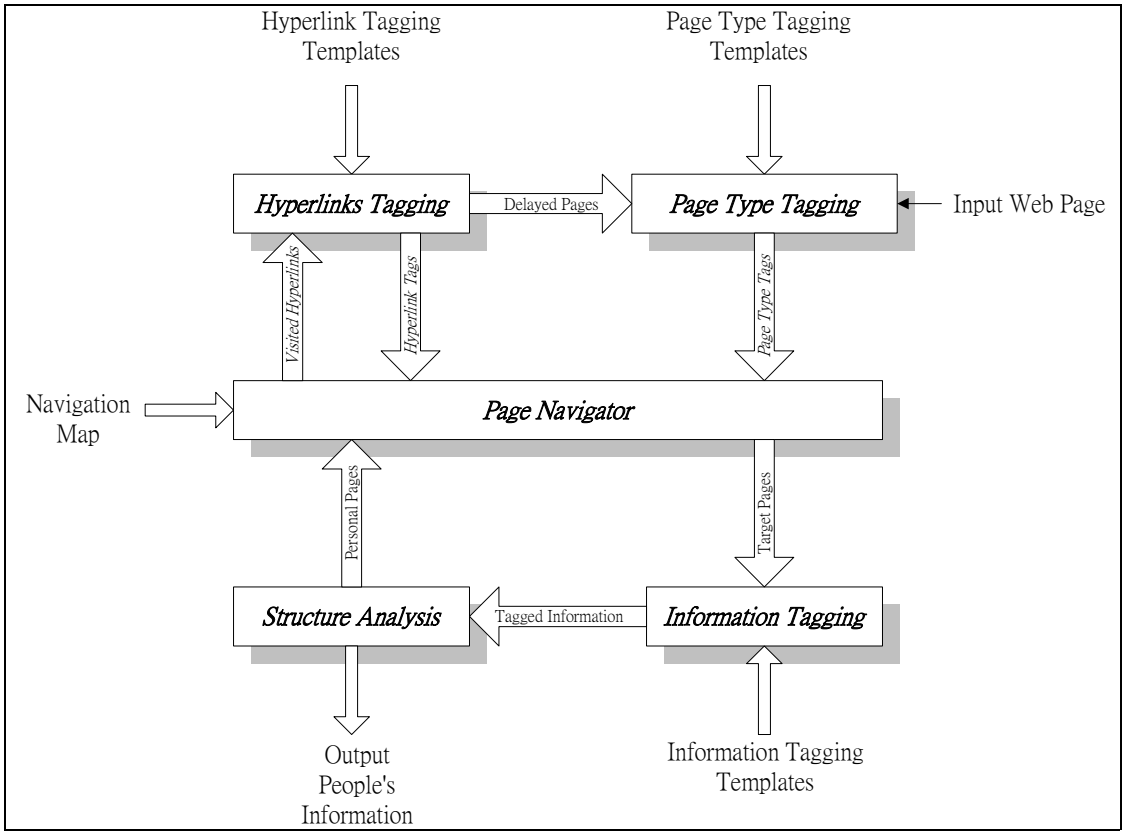
However, our semantic tagger focuses more on Chinese. Tags and semantic templates for English will be ready in a few months.

Besides these useful tags, we will also equip our *PNA* with a navigation map. The navigation map will guide our *PNA* to traverse through related pages and to arrive at pages containing the target information.

This paper is arranged as follows. In Section 2, we will describe various semantic tags used for our *PNA*. The underlying semantic templates for the *TS* are discussed in Section 3. Section 4 discusses the navigation map and how it guides our *PNA* to navigate the web. Experimental results are presented in Section 5. We discuss related works in Section 6 and a brief discussion of the future research in Section 7.

2. The Tagging System

As much as the navigation map guides our *PNA* through different web pages semantic tags guide our *PNA* to read through a web page and to collect relevant information. Our *TS* also leaves important marks on the navigation map to facilitate the traversal of the *PNA*. There are three kinds of tags in our *TS*: page type tags (*PTT*), hyperlink tags (*HT*) and information tags (*IT*). A page type tag is used to determine the type of this web page. A hyperlink tag predicts the type of the page referred to by a hyperlink. Information tags give the underlying semantic meaning of part of texts. Figure 1 gives an overview of the *PNA* system in terms of semantic tags and navigation maps.



Page Navigator will predict the target pages from pages accepted by hyperlinks tags and page type tags, or generate hyperlinks for those that are not the target pages.

Figure 1. *PNA* system overview

2.1. Page Type Tags

PTT is a useful information for the web navigation. The types of pages used are usually application dependent. In our experiment, possible types of page type are University, Education-Units, College, Department, Education-Resource, Research-Groups, Faculty, and Person (cf. Table 1). For each web page, we inspect the following three areas to determine *PTT*: page title, page headline and page body. Sometimes, the *URL* would also provide important information for the *PTT*.

Page type Tags	University, Education-Units, College, Department, Education-Resource, Research-Groups, Faculty, and Person
Hyperlink Tags	L-University, L-Education-Units, L-College, L-Department, L-Chinese, L-Education-Resource, L-Research-Groups, L-Faculty, and L-Person

The *HT* L-Chinese is a link that leads to Chinese home pages

Table 1. Types of hyperlinks and page type tags

2.2. Hyperlink Tags

A user browsing through a web page will often select a hyperlink of interest and enter into its associated page. The relevance of such a hyperlink usually depends on the text description on the hyperlink. For our *PNA* to simulate such behavior, we shall classify the type of *HT* in a similar way as we classify *PTT*. The possible types of *HT* for our experiment are listed in Table 1.

There are many hyperlinks in a web page, but not all of them are useful for our navigation. Sometimes, the description on the hyperlink is unclear and we need to take contextual information into consideration. The *PNA*'s task is to find (or sometimes guess) the most probable type of *HT* and to decide whether it is useful for further navigation. An *HT* predicts the type of the page referred to by a hyperlink. Note that a more accurate *HT* can be obtained if our *PNA* does navigate into the corresponding page and analyzes its *PTT*. However, this beats its own purpose.

Hyperlink tags are applied to the anchored string, the *ALT* names of images or maps, and *URL* names.

2.3. Information Tags

Information tags are applied to the page body to identify useful semantics of parts of the text. *IT* can be used to classify the *PTT* type. In our experiment, useful *IT* include people's names, telephone numbers, fax numbers, e-mail addresses, and homepage *URLs*.

After tagging, higher level semantic templates are used to characterize, connect and organize tagged personal information. The *PNA* will analyze the text structure to determine the relationships of the tagged information. At this step we associate all relevant information to the right person.

The details are discussed below. Information within a list item belongs to the same person. This is true for both ordered and unordered lists. If there is no owner information in a list item, we may assume that the list belongs to the *last appearing* (*LA*) owner of previous list items that is inside an <ol> or <ul> scope. For table structures in *HTML*, we apply a general table analysis. Usually, personal information listed in a table follows the left and top adjacent principle. Therefore, we assume that information in a table cell with no owner is bound to the *LA* owner in the same row or to the owner of the top cell in the same column.

The algorithm for finding the owner of the tagged information in a table cell is described as follows:

```
For Each Row <tr> … [</tr>] in Table
    Last_Owner := nil
    For Each Cell <td>..</td> or <th> .. </th> in Row
        Determine the Cell Location (X, Y) with rowspan and colspan
            Information
        Apply Information Tagging to Document in Cell
        If a Name tagged then
            Cell_Owner := The first tagged Name
            Last_Owner := Cell_Owner
            Create an Information Set for Cell_Owner, and insert it into the
                pool of Information Sets
        Else
            If Last_Owner != nil then
                Cell_Owner := Last_Owner
            Else
                Cell_Owner := LA Cell_Owner from (X, Y-1) to (X, 0),
                    unless Y is 0
Bind All Information Tags to a Cell_Owner's Information Set
```

Figure 2. An algorithm for table analysis

For the case of free or semi-structured text, information in a line can usually be bound to either the owner found at this line or the last owner found at the previous line.

3. Semantic Templates

The tagging system for our *PNA* is supported by an *NL* understanding system whose kernel is designed based on that of *GOING*[7]. *GOING* is an *NLP* system constructed in 1995 for deciphering Chinese homophones based on the context. *GOING* is more or less equivalent to a word sense tagging system for English. But, it is more versatile in that it also gives the underlying semantic meaning of phrases. The main thrust of *GOING* is a hierarchical semantic pattern-matching algorithm. Besides a lexicon of 50,000 words, *GOING* contains 254 semantic categories arranged in a semantic category tree. Furthermore, *GOING* uses over 30,000 semantic templates. These templates give certain relationships among the words. Some templates will form noun phrases and verb phrases. Each item in a template is associated with a weight so that, when the template is successfully matched, these items will gain their respective weights. Semantic templates are divided into different hierarchies such that templates in the same hierarchy are processed in parallel. The collection of templates is obtained through intensive corpus analysis and human mediation.

The template system for *PNA* uses the same philosophy and semantic category tree. Depending on the information domain we sometimes have to add a few more

semantic categories. Some templates will give the semantic tag for a word or a phrase. These templates are constructed in a semi-automatic fashion, namely, based on statistical corpus analysis combined with human supervision. Besides our *NL* corpus, we also download a large collection of *HTML* documents from the web to form a web corpus.

A semantic template consists of a sequence of tokens. There are several attributes on the templates as discussed below.

- length(T, [Num_{From}, Num_{To}]): The number of tokens in the range from Num_{From} to Num_{To} matched by template T.
- is(T, *Feat*, *Value*): The sequence of tokens matched by T must satisfy that its feature *Feat* has the value *Value* (e.g., is(T, *First-Name*, *true*) requires that tokens matched by template T can be used as a first name). The set of features includes semantic category in a semantic tree and a *key_fragment* feature used for matching a specific text fragment.
- position(T, *From*, [Num_{From}, Num_{To}]): The template T matches tokens whose positions are between the Num_{From}-*th* token and the Num_{To}-*th* token measured from the *first* or *last* token as specified in the variable *From*.
- relpos(T_1, T_2, [Num_{From}, Num_{To}]): The two sequences S_1 and S_2 of tokens matched by two templates T_1 and T_2, respectively, satisfy that S_1 is in front of S_2 and the number of tokens between S_1 and S_2 ranges from Num_{From} to Num_{To}.
- set(T, *Setop*, T_1, T_2): The template T's property must be the *union* or the *difference* of two other distinct sub-templates T_1 and T_2.
- compose(T, T_1, T_2): The sequence of tokens in T must start from the first token in T_1 and end at the last token in T_2.
- repeat(T, [Num_{From}, Num_{To}], T_r): Define the number of times (from Num_{From} to Num_{To}) template T can be applied and let template T_r be the result of repeating.
- info(T, *Tag*, *Degree*): The sequence of tokens matched by template T will be assigned the *Tag* with weight *Degree*. This attribute only appears in the head of semantic rule. When the strings matched by several rules overlap each other, the matched string with highest degree is accepted.

In the following we illustrate a semantic template for composing a Chinese name. Chinese words in a sentence do not have delimiters. This makes the unknown word identification a little cumbersome. A Chinese name consists of two contiguous parts: Last name, First name. Most last names have one character, but a few have two. Most first names have two characters, but single character first name is becoming popular in Mainland China. There are many characters that can be used as last name or first name with different probabilities. The category [ma] includes all characters that are

very likely to be part of a male's first name. The category [mb] includes all characters that are likely to be part of a male's first name. This template will compose one type of probable Chinese names.

4. Navigation Tour of *PNA*

Assume we are given the name of a university and try to find its faculty's information. We shall provide our *PNA* with a navigation map to guide it to the target web site. Each node of a navigation map represents a page type. Each edge of the map is directed from one page type to another. Each edge represents a state transition. The transition includes the current page type, the next page type and a specified hyperlink type. We shall assign a priority for each edge emanating from the same node based on their relevance. We can define possible actions at each node. Such actions can be one of the following: collect hyperlinks, determine whether the navigation should continue, stop or roll back. The format of navigation map is defined below:

Act [Page-Type Tag**] {**Actions in order**}**
(Page-Type Tag**,** Hyperlink Tag**) → (**Next Page-Type Tag, Order**)**

In this format, we specify the actions that should be triggered when the *PNA* encounters a particular page type and define the next possible page types from the current one. During the transition, we need to do a hyperlink type checking to see if there exists any relevant hyperlink. In case there are more than one possible transitions, we order their next page types according to their priority of navigation

The navigation map is constructed by analyzing a corpus of related web pages. To start the navigation, the *PNA* sends the name of the university to an existing search engine. Then it applies a page type tagging to the retrieved pages until we get a page with the desired page type (e.g., a University type). In case there is no description on a hyperlink, we apply page type tagging to the page referred to by the hyperlink and ignore the *HT* in the transition. Figure 1 gives an example of a navigation map.

Act [Faculty**] {**Extract-All-Faculty-Info-and-Personal-Links**}**
Act [University, Departments, Education-Resource**]{**Collect-All-Hyperlinks**}**
(University**,** L-Chinese**)→(**University**,** 2**)**
(University**,** L-Departments**)→(**Departments**,** 1**)**
(Departments**,** L-Department**)→(**Department**,** 1**)**
(Department**,** L-Chinese**)→(**Department**,** 3**)**
(Department**,** L-Education-Resource**)→(**Education-Resource**,** 2**)**
(Department**,** L-Faculty**)→(**Faculty**,** 1**)**
(Education-Resource**,** L-Faculty**)→(**Faculty**,** 1**)**

Figure 3: An example of a simple navigation map

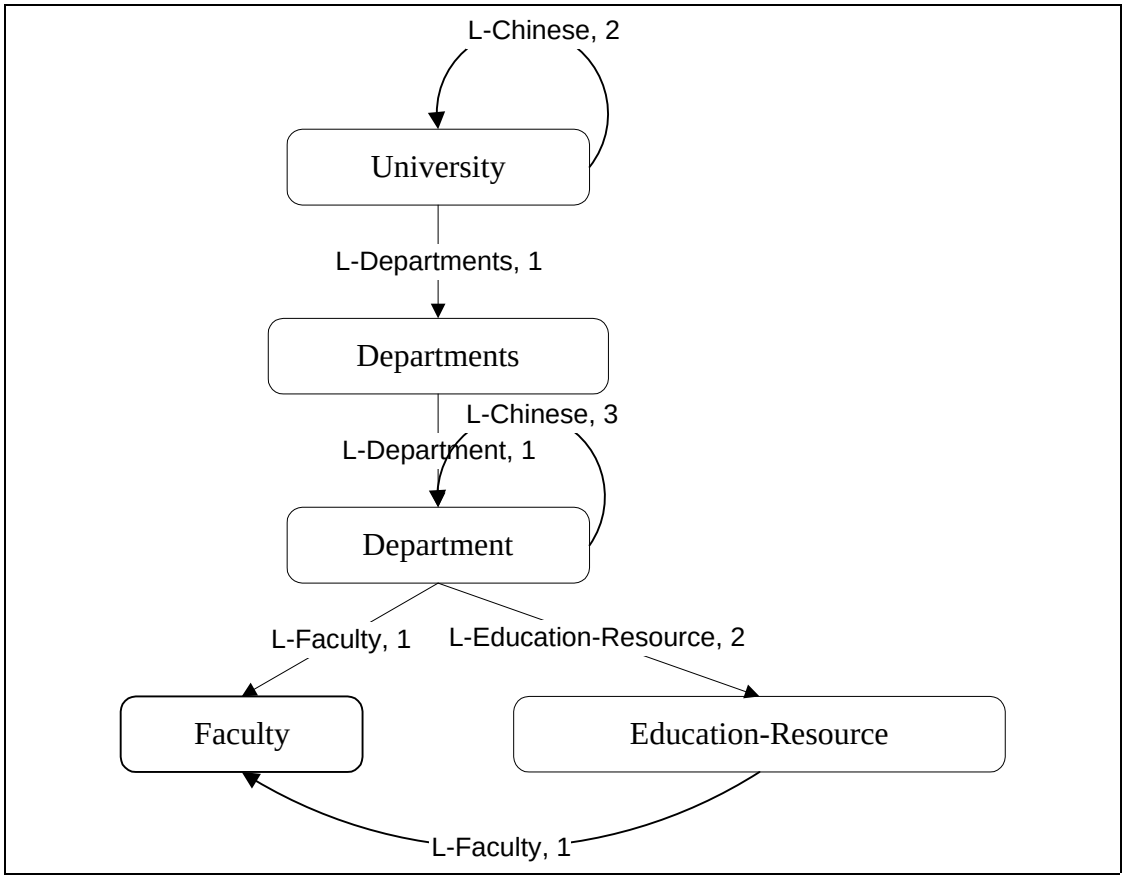


Figure 4. A simple navigation map from begin University to target Faculty

The navigation map allows us to build a tree of traversed pages. For each incomplete information such as a stand-alone telephone extension number, we can trace back to ancestor pages for an organization's telephone number in the cache, and merge them together to form a complete telephone number.

5. Experimental Results

Our *PNA* is implemented in C++. We choose National Tsing Hua University in Taiwan (http://www.nthu.edu.tw) as our test site. There are over one hundred semantic templates in our system. We use 8 page types to construct the navigation map.

The *PNA* is designed with two main windows: the top window, the navigation tree window displaying the tree of navigated pages, and the lower window displaying the information collected at the selected page in the navigation tree window. Users can put the input *URL* in the Combo Box above the navigation tree window and press RETURN to execute the program.

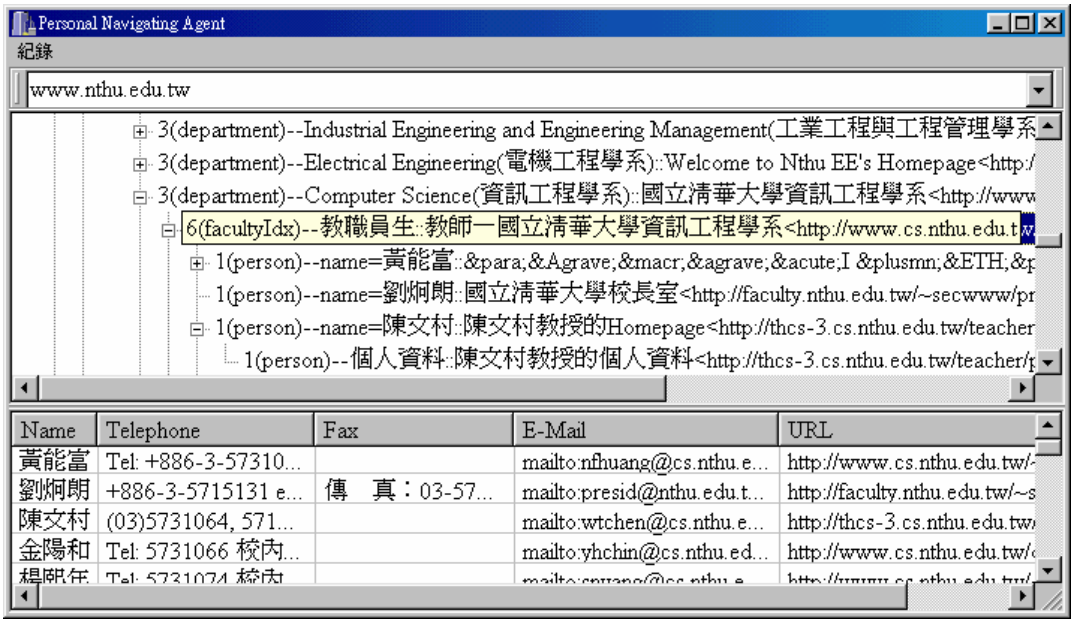


Figure 5. A screenshot of our *PNA* system

Name	Telephone	Fax	Email	URL
陳理定	(03)5715131 轉 3756	(03)5722840	nil	www.pme.nthu.edu.tw/teacher/陳理定/陳理定.html
劉通敏	(03)5715131 轉 3752	(03)5722840	tmliou@pme.nthu.edu.tw	http://www.pme.nthu.edu.tw/teacher/劉通敏/劉通敏.html
洪英輝	(03)5715131 轉 3740	(03)5722840	yhhung@pme.nthu.edu.tw	http://www.pme.nthu.edu.tw/teacher/洪英輝/洪英輝.html
李雄略	(03)5728230、5715131 轉 3732	(03)5728230	nil	http://www.pme.nthu.edu.tw/teacher/李雄略/李雄略.html
楊鏡堂	(03)5715131 轉 3732	(03)5724242	jtyang@pme.nthu.edu.tw	http://www.pme.nthu.edu.tw/teacher/楊鏡堂/楊鏡堂.html

Figure 6. An example of collected information

Table 2 shows the experimental result of information tagging. Three attributes, Name, Telephone, and Fax, are tagged using semantic templates. Attributes of Email and URL are tagged from the *HTML* anchor tags directly.

	Name	Telephone	Fax	Email	URL
#tagged	734	298	75	379	1048
#correct	717	296	75	379	1032
Accuracy(%)	97.68	99.32	100.00	100.00	98.47

Table 2. Accuracy of Information Tagging

Table 3 shows the rate of irrelevant pages filtered by applying *HT*, *PTT*, and the navigation map. The *PNA* encountered 2605 web pages and rejected 1826 pages during hyperlink tagging and rejected another 346 pages during page type tagging. The total filter rate is the number of total rejected *URLs* over the number of total visited *URLs*. The filter rate of *HT* with the navigation map is the number of rejected *URLs* over the total number of visited *URLs*. The filter rate of *PTT* with navigation map is the number of rejected *URLs* over the number of delayed *URLs*. The processing of hyperlinks without tagging result and anchored strings are delayed during page content tagging.

	Total Visited	Hyperlink Tags			Personal	Page Content Tags	
		Accepted	Rejected	Delayed		Accepted	Rejected
#URL	2896	124	2072	404	296	50	354
Filter Rate(%)	83.77	71.55			N/A	87.62	

Table 3. Filter Rate of Tagging with Navigation Map

Table 4 shows the result of information assignment for names assigned to faculty and attributes assigned to correct faculty names.

Faculty names are tagged and collected at each department's faculty page.

#total correct : total number of correct faculty names

#collected: total number of collected faculty name

#correct: total number of correct faculty names collected

The other four attributes are tagged and collected at each department's faculty page, each faculty's personal page listed at each department, and each personal home page listed elsewhere. Our *PNA* tries to assign these attributes to the correct faculty name (or owner).

Faculty	#total correct	#collected	#correct	Recall(%)	Precision(%)
Name	401	370	367	91.52	99.18
Owner assignment					
Faculty	#total correct	#assigned	#correct	Recall(%)	Precision(%)
Telephone	187	198	170	90.91	85.86
Fax	83	71	70	84.34	98.59
Email	236	260	233	98.73	89.62
URL	274	286	249	90.88	87.06

Table 4. Result of structural analysis from information tagging

6. Related Work

The "World Wide Knowledge Base" project [2][4] at Carnegie Mellon University is the most closely related work. In order to build a knowledge base from the Web,

they developed a “crawler” to explore the Web and extract interesting classes and relations using machine-learned relational rules. Their work is different from ours in several aspects. First, the expressive power of the rules is different. Our rules may include set operators such as union and difference (which to some extent implements negation), while their rules can apply web-dependent predicates such as whether two pages are linked to each other. Second, their rules are learned from web pages in a target domain, while our rules are obtained from analyzing a large domain-independent natural language corpus. Third, our navigation is configurable. Users can define their navigation map for a new information domain. According to [2], the “crawler” does not reason whether a hyperlink is relevant and may explore a large number of irrelevant web pages.

Many components of our agent are related to previous work. Tagging is an important problem in natural language processing. [1] presented a tagging system that recognizes personal names, organization names, E-mail addresses and *URLs*. The key idea of their approach can be summarized as follows. To tag personal names, if the relative frequency that a given token is used as a personal name exceed some threshold, then the token is tagged as a personal name. They performed an experiment on 703 web pages drawn from a university web site in Taiwan, a similar domain as in our experiment. However, the accuracy (precision) of personal name tagging is 46.46%, significantly lower than our result. It is evident that their statistical approach is simple and does not perform as well. Expressive semantic knowledge plus guided exploration to relevant web fragments and pages might be the reasons why *PNA* can achieve superior results in our experiment.

Document structural analysis is also an active research problem in these years. [8] describes an approach to automatic analysis of the structure of a tabular document. [6] proposes to build a template for each general structure of *HTML* documents in order to extract information from the Web. Our structural analysis is similar to theirs but our analysis also takes tagged information into account and thus can provide more accurate association of relevant data (e.g., associating personal names with their corresponding E-mail addresses).

7. Future Research

In our current experiment, we extract all faculty information in a university. While this might not be a typical activity for the *PNA* (usually one is only interested in a specific person in some organization), the experiment is conducted for the sake of statistical analysis. The results give us an indication how the *PNA* will perform in the average case.

Our experiments show that semantic tagging together with navigation maps result in an impressive performance for an automatic web-browsing agent. Although the information our *PNA* currently extract is very restricted, we believe the methodology is domain independent. It is interesting to see how this semantic tagging approach can be applied to structural analysis in general. Currently, we have only used around one hundred semantic templates in our *PNA*. Compared to our *GOING* system (which works for a wide variety of domains in the newspaper) that adopted 30,000 templates, there is still a lot of room for expansion. In the future, we shall focus on automatic template learning from *HTML* corpus.

The idea of *PNA* has wide applications. A *PNA* seems to be a handy assistant for anyone on the web. We shall embed *PNA* in other on-going agent systems such as travel agents and information service agents.

References

[1] H. H. Chen, G. W. Bian 1998. White Page Construction from Web Pages for finding People on the Internet. In International Journal of Computational Linguistics and Chinese Language Processing vol.3 no.1 Feb.

[2] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery 1998. Learning to Extract Symbolic Knowledge from the World Wide Web. In Proceeding of 15th National Conference on Artifical Intelligence (AAAI-98).

[3] Defense Advanced Research Projects Agency. 1995. Proceedings of the sixth message understanding conference (MUC-6), Morgan Kaufmann Publishers, Inc.

[4] D. Freitag 1998. Information Extraction from HTML: Application of a General Machine Learning Approach. AAAI98.

[5] C. N. Hsu, 1998, Initial Results on Wrapping Semi-structured Web Pages with Finite-States Transducers and Contextual Rules. In Proceedings og AAAI-98 Workshop on AI and Information Integration, Technical Report WS-98-14, AAAI Press, Men Park, CA.

[6] J. Y. J. Hsu and W. T. Yih 1997. Template-Based Information Mining from HTML Documents. AAAI97.

[7] W. L. Hsu, 1995. Chinese parsing in a phoneme-to-character conversion system based on semantic pattern matching. International Journal on Computer Processing of Chinese and Oriental Languages 40, (1995),227-236.

[8] M. Hurst and S. Douglas, 1997. Layout and Language: Preliminary investigations in recognizing the structure of tables. In Proceedings of ICDAR'97, August 18-20.

[9] Kushmerick, N.1997. Wrapper Induction for Information Extraction. Ph.D. Dissertation, Department of Computer Science and Engineering, University of Washington, Seatle, WA.

[10] I. Muslea, S. Minton, C. Knoblobk. 1998. STALKER: Learning Extraction Rules for Semi-structured Web-based Information Source. In Proceedings of AAAI-98 Workshop on AI and Information Integration, Technical Report WS-98-14, AAAI Press, Men Park, CA.