A Natural Language Processing Based Internet Agent

Ming-Hsuan Yang
Beckman Institute
University of Illinois
Urbana, IL 61801
Email: myang1@uiuc.edu

Christopher C. Yang
Department of Computer Science
The University of Hong Kong
Hong Kong
Email: yang@cs.hku.hk

Yi-Ming Chung
NCSA
University of Illinois
Urbana, IL 61801
Email: ychung@ncsa.uiuc.edu

Abstract

Searching for useful information is a difficult job by the virtue of information overloading problem. With the technological advances, notably World-Wide Web (WWW), it allows every ordinary information owner to offer information on line for others to access and retrieve. However, it also makes up a global information system that is extremely large-scale, diverse and dynamic. Internet agents and Internet search engines have been used to deal with such problems. But the search results are usually not quite relevant to what a user wants since most of them use simple keyword matching.

In this paper, we propose a natural language processing based agent (NIAGENT) that understands a user's natural query. NIAGENT not only cooperates with a meta Internet search engine in order to increase *recall* of web pages but also analyzes the contents of the referenced documents to increase *precision*. Moreover, the proposed agent is autonomous, light-weighted, and multithreaded. The architectural design also represents an interesting application of distributed and cooperative computing paradigm. A prototype of NIAGENT, implemented in Java, shows its promise to find useful information than keyword based searching.

1 Introduction

With the phenomenal growth of Internet and World Wide Web, the information overload problem is more serious and inevitable. Although a lot of information is available on the Internet, it is usually difficult to find particular pieces of information efficiently. To address this problem, several tools have been developed to help search relevant information more effectively by either assisted browsing or keyword/phrase based searching. Assisted browsing, such as WebWathcer [6] and Syskill & Webert [10], guides/suggests a user along an appropriate path through the web based on its knowledge of the user's interests, of the location and relevance of various items in the collection, and the way in which others have interacted with the collection in the past. Internet search engines, such as AltaVista and Lycos, sends out spiders or robots to index any visited web pages and allow keyword or phrase based search. One characteristic of these approaches is they all rely on central sever to solve the problem. Also, these approaches are either time consuming or the search results are often not quite relevant to what a user wants.

The agent concept can be used to simplify the solution of large problems by distributing them to some collaborating problem solving units. This distributed problem solving paradigm is particularly suitable for information retrieval on the web. In this paper, we focus on developing an intelligent and efficient search agents. All the major search engines use different schemes to index web pages by using keyword or phrase, and support Boolean operations in keyword or phrase search. The major problem with these search engines is that many irrelevant pieces of information are also returned (i.e. low precision of extracted information) since they use unordered keyword and phrase as indices. In other words, a document is deemed as relevant to a query if all the phrases are matched. But since different phrases can appear in the same text with any relationship between them, the recalled web pages by keyword based matching are usually uninteresting or irrelevant to a user's query. Also, each search engine usually returns different documents for the same query because they use different ranking algorithm in indexing, different indexing cycles, and different resources. It has been shown in [12] that users could miss 77% of the references they would find most relevant by relying on a single search engine. MetaCrawler [13] is a meta search engine designed to address these problems by aggregating Internet search services. In order to achieve high *recall* in

0-7803-4053-1/97/$10.00 © 1997 IEEE

retrieved documents, it is necessary to cooperate with as many Internet search engines as possible. However, the search results of MetaCrawler are usually not relevant to the query because of the problem with keyword matching. Therefore, it is important to increase *precision* in the retrieved information.

We propose a Natural language processing based Internet AGENT, NIAGENT, that understands natural queries. It cooperates with an NLP agent at MIT Media Lab to understand the input query. Meaningful noun phrases are extracted out of the natural sentence and formated as appropriate query to search engines such as MetaCrawler or other search engines. NIAGENT fetches back the web pages, based on the recalled references from search engines, and then cooperates with PARAGENT (PARAgraph Analysis AGENT) for analyzing their text contents to sift out irrelevant documents. By cooperating with MetaCrawler and PARAGENT, NIAGENT increases not only recall but also precision in retrieved documents. Our experiments show that NIAGENT is not only more user-friendly but more effective in searching for relevant and useful information.

2 NIAGENT

We agree with the arguments of Lewis and Jones in [7] that "All the evidence suggests that for end-user searching, the indexing language should be natural language, rather than controlled language oriented ... For interactive searching, the indexing language should be directly accessible by the user for request formulation; users should not be required to express their needs in a heavily controlled and highly artificial intelligence." and "Evidence also suggests that combining single terms into *compound terms*, representing relatively fixed complex concepts, may be useful ..." In light of this trend, we propose an intelligent agent that cooperates with other agents to understand the natural query, extract meaningful noun phrases, and analyze the search results. Natural query can alleviate users from the pains and efforts to learn strict formats in different Internet search engines. Analysis of the text, based on noun phrases, helps in extracting relevant information with high precision. With recent research results in artificial intelligence and natural language processing, mature technologies are ready to help in designing intelligent information filtering systems.

Figure 5 shows the architectural design of NIAGENT. The design represents an example of distributed and cooperative computing in that NIAGENT cooperates with other agents to understand the user's interests and to search for relevant references. A user makes a natural query without the need to learn different formats in various Internet search engines. NIAGENT cooperates with Chopper, a natural language understanding agent, to figure out the interests of the user by extracting meaningful phrases. Appropriate queries to Internet search engines or spiders are then made by NIAGENT. Based on the returned hyperlinks from search engines, NIAGENT fetches back the referred documents and ask PARAGENT to sift out irrelevant documents. Finally, the relevant documents are returned to user.

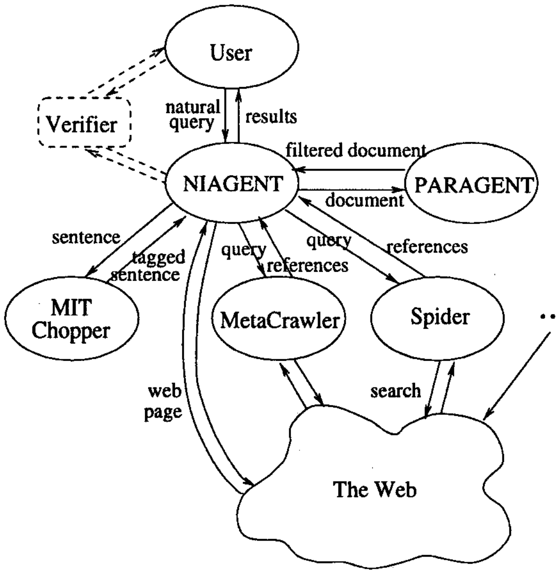
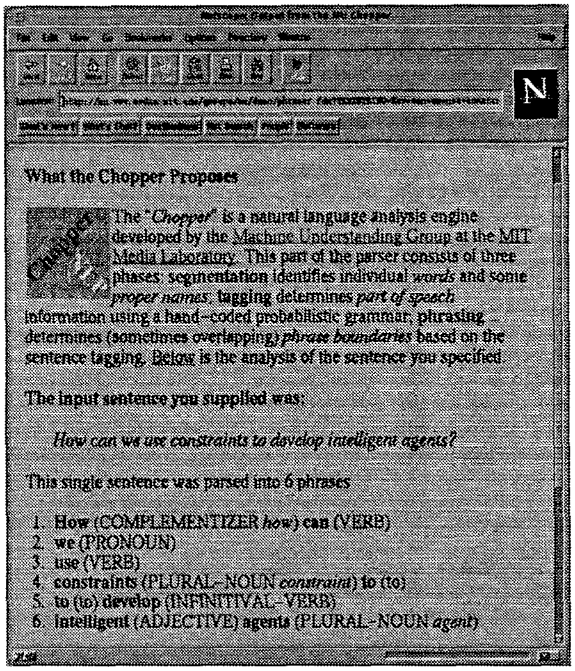


Figure 1: NIAGENT architecture

2.1 Understanding a Natural Query

In order to build an intelligent system as a society of interacting agents, each having their own specific competence, to "do the right thing" [8], NIAGENT cooperates with Chopper to understand a user's natural query. Chopper, developed by the Machine Understanding Group at MIT, is a natural language analysis engine that generates an analysis of the phrase structure and parts of speech in the input sentence. This parser consists of three parts: segmentation identifies individual words and some proper names, tagging determines part of speech information using a hand-coded probabilistic grammar; phrasing determines (sometimes overlapping) phrase boundaries based on sentence tagging [4].

Instead of learning to form a query based on Boolean operations and phrases for a specific Internet search engine, a user interested in how we can use search techniques of constraint satisfaction to develop an intelligent agent can ask NIAGENT a question, in natural English, like "How can we use constraint to develop intelligent agents?" NIAGENT cooperates with Chopper to understand the interests of the user and then extract the meaningful nouns phrases to form appropriate queries for other agents such as MetaCrawler or various Internet spiders [1]. Figure 2 shows the analysis result of the natural query discussed above.



What the Chopper Proposes

The "*Chopper*" is a natural language analysis engine developed by the Machine Understanding Group at the MIT Media Laboratory. This part of the parser consists of three phases: **segmentation** identifies individual *words* and some *proper names*, **tagging** determines *part of speech* information using a hand-coded probabilistic grammar, **phrasing** determines (sometimes overlapping) *phrase boundaries* based on the sentence tagging. Below is the analysis of the sentence you specified.

The input sentence you supplied was:

How can we use constraints to develop intelligent agents?

This single sentence was parsed into 6 phrases

1. **How** (COMPLEMENTIZER *how*) **can** (VERB)
2. **we** (PRONOUN)
3. **use** (VERB)
4. **constraints** (PLURAL-NOUN *constraint*) **to** (to)
5. **to** (to) **develop** (INFINITIVAL-VERB)
6. **intelligent** (ADJECTIVE) **agents** (PLURAL-NOUN *agent*)

Figure 2: Analysis results of MIT Chopper

It has been shown in [2] that an occurrence of the noun phrase in a document usually provides much more evidence than other phrases for the concept of the text. For the query described just described, only the noun phrase, "constraint" and "agent", capture the concept of the natural query. Other phrases such as "how", "can", "we", "use", etc, do not carry useful information in the query. Based on this analysis, NIAGENT extract meaningful noun phrases from the analyzed results suggested by Chopper and understands the user's natural query.

2.2 Cooperating with Internet Search Engines

The World Wide Web can be viewed as an information food chain where the maze of web pages and hyperlinks are at the very bottom of the chain [3]. In this analogy, the Internet search engines such as AltaVista or Lycos are information herbivores that they graze on web pages and regurgitate them as searchable indices. MetaCrawler is developed to be one of the information carnivores that hunt and feast on herbivores. NIAGENT is also on the top of the information food chain that it works with MetaCrawler in order to intelligently hunt for useful information. Concept-contained noun phrases, generated by NIAGENT and Chopper, are passed to MetaCrawler as directives to hunt for information herbivores. Finally, the caught preys are then forwarded back to NIAGENT.

2.3 Analyzing Web Pages

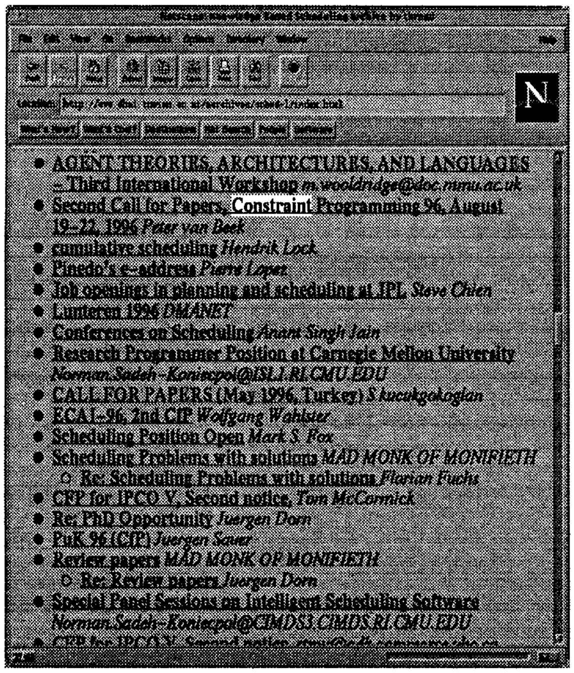
Most search engines return a user with hyperlinks that contain the queried phrases. However, a recalled web page that has keywords in the text is not necessarily relevant to the query. It has been shown that basic compound phrases would not typically be further combined into frames, templates, or other structured units unless there is a syntactic or semantic relationship between them [7]. For example, a web page might have "constraint satisfaction problem" and "agent" in a list of research interests or different paragraphs about problem solving techniques. Most Internet services would think, based on keyword matching, this web page is relevant to the user's query discussed previously since the key phases "constraint" and "agent" are matched. But the contents of the web page are actually not related to the interests of the user. Figure 3 shows a paragraph of a web page that consists of a list of conferences including the queried phrases.

- AGENT THEORIES, ARCHITECTURES, AND LANGUAGES - Third International Workshop *m.wooldridge@doc.mmu.ac.uk*

- Second Call for Papers, Constraint Programming 96, August 19-22, 1996, *Peter van Beek*

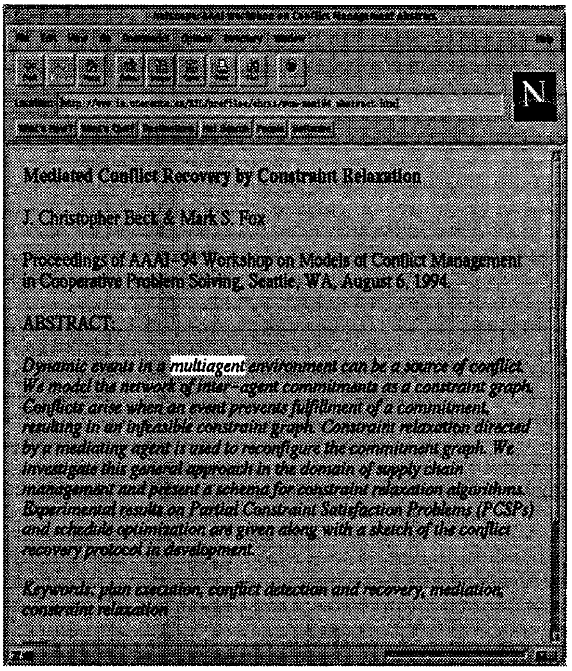
However, the description of this document is just an unordered set of phrases and individual words. Therefore, this web page should not be considered relevant to the user's interest.

The relevant information to be extracted is the relationships between individual phrases. It is not enough



- AGENT THEORIES, ARCHITECTURES, AND LANGUAGES – Third International Workshop *m.wooldridge@doc.mmu.ac.uk*
- Second Call for Papers, Constraint Programming 96, August 19–22, 1996 *Peter van Beek*
- cumulative scheduling *Hendrik Lock*
- Pinedo's e-address *Pierre Lopez*
- Job openings in planning and scheduling at JPL *Steve Chien*
- Lunteren 1996 *DMANET*
- Conferences on Scheduling *Anant Singh Jain*
- Research Programmer Position at Carnegie Mellon University *Norman.Sadeh-Koniecpol@ISL1.RI.CMU.EDU*
- CALL FOR PAPERS (May 1996, Turkey) *S kucukgokoglan*
- ECAI–96, 2nd CfP *Wolfgang Wahlster*
- Scheduling Position Open *Mark S. Fox*
- Scheduling Problems with solutions *MAD MONK OF MONIFIETH*
  - Re: Scheduling Problems with solutions *Florian Fuchs*
- CFP for IPCO V, Second notice, *Tom McCormick*
- Re: PhD Opportunity *Juergen Dorn*
- PuK 96 (CfP) *Juergen Sauer*
- Review papers *MAD MONK OF MONIFIETH*
  - Re: Review papers *Juergen Dorn*
- Special Panel Sessions on Intelligent Scheduling Software *Norman.Sadeh-Koniecpol@CIMDS3.CIMDS.RI.CMU.EDU*

Figure 3: A web page with matched keywords (http:// www. dbai. tuwien.ac.at/ marchives/ sched-l/ index.html)



Mediated Conflict Recovery by Constraint Relaxation

J. Christopher Beck & Mark S. Fox

Proceedings of AAAI–94 Workshop on Models of Conflict Management in Cooperative Problem Solving, Seattle, WA, August 6, 1994.

ABSTRACT:

Dynamic events in a multiagent environment can be a source of conflict. We model the network of inter–agent commitments as a constraint graph. Conflicts arise when an event prevents fulfillment of a commitment, resulting in an infeasible constraint graph. Constraint relaxation directed by a mediating agent is used to reconfigure the commitment graph. We investigate this general approach in the domain of supply chain management and present a schema for constraint relaxation algorithms. Experimental results on Partial Constraint Satisfaction Problems (PCSPs) and schedule optimization are given along with a sketch of the conflict recovery protocol in development.

Keywords: plan execution, conflict detection and recovery, mediation, constraint relaxation

Figure 4: A web page with matched keywords (http://www.ie.utoronto.ca/ EIL/profiles/ chris/ cwdai.abstract.html)

to identify isolated phrases, which can be done by a simple keyword search. In order to intelligently extract relevant information from the World Wide Web, NIAGENT first fetches back the web pages recalled by the search engines and pass them to PARAGENT for analyzing the syntactic relation of phrases. While a web page contains all the queried phrases is not necessarily relevant to a user's interests, a web page is usually directly relevant to the query if those phrases appear in the same logical segment. The rationale here is that more complex structures for deeper understanding of text is computationally expensive and difficult whereas simple keyword match is not likely to provide good precision in information retrieval.

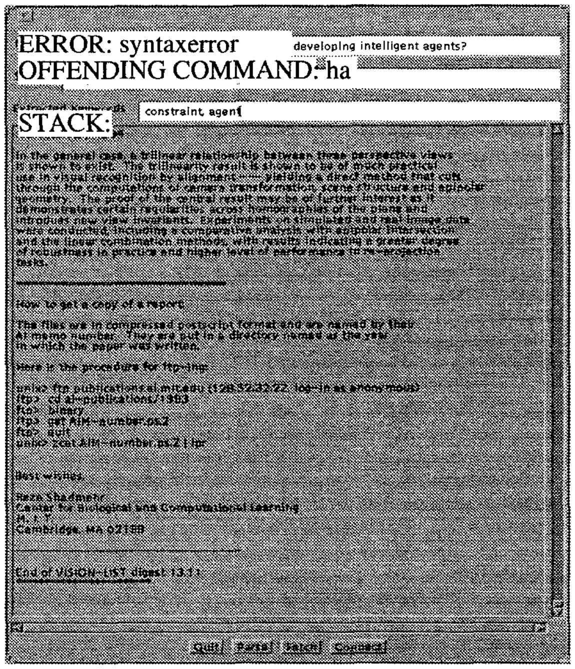
PARAGENT uses the page layout cues to divide a web page into coherent segments (usually paragraphs) as the first step in analyzing the contents. Then the relationships between noun phrases are analyzed by PARAGENT to determine the relevance of the web page. Figure 4 shows an Web document that contains information interesting to the user. Note that all the meaningful noun phrases extracted by NIAGENT and Chopper appear in the same paragraph.

Figure 5 shows a prototype of NIAGENT based on our architectural design. A user can key in a natural query and select a Internet search engine (such as MetaCrawler, AltaVista, etc). NIAGENT will show the concept-contained keywords and send them as queries. The analyzed web pages that are relevant to the input query are displayed in the text field. Some features of the current implementation (in Java) are multithreaded, light-weighted, and portable.

2.4 Experimental Results

In order to compare the performance of NIAGENT with other Internet search engines in terms of precision, we conduct several experiments. For each example in the test set MetaCrawler returns 20 references (same amount of *recall*) and NIAGENT determines the precision based on the contents of the recalled web pages. These web pages are then evaluated by two fellow colleagues to determine the relevance. The results in Table 1 summarizes the performance results of NIAGENT versus MetaCrawler.

For the first test query, only 7 out of 20 web pages recalled by MetaCrawler are relevant to the user's interest. On the other hand, NIAGENT achieves high precision (90%) in this experiment. The result is not surprising since NIAGENT takes one more step to analyze the contents of the referenced web pages rather than use simple keyword search. From our prelimi-



ERROR: syntaxerror
OFFENDING COMMAND: ha
STACK:

developing intelligent agents?

Extracted Keywords: constraint, agent

In the general case, a trilinear relationship between three perspective views is shown to exist. The trilinearity result is shown to be of much practical use in visual recognition by alignment --- yielding a direct method that cuts through the computations of camera transformation, scene structure and epipolar geometry. The proof of the central result may be of further interest as it demonstrates certain regularities across homographies of the plane and introduces new view invariants. Experiments on simulated and real image data were conducted, including a comparative analysis with epipolar intersection and the linear combination methods, with results indicating a greater degree of robustness in practice and higher level of performance in re-projection tasks.

How to get a copy of a report:

The files are in compressed postscript format and are named by their AI memo number. They are put in a directory named as the year in which the paper was written.

Here is the procedure for ftp-ing:

```
unix> ftp publications.ai.mit.edu (128.52.32.22, log-in as anonymous)
ftp> cd ai-publications/1993
ftp> binary
ftp> get AIM-number.ps.Z
ftp> quit
unix> zcat AIM-number.ps.Z | lpr
```

Best wishes,

Reza Shadmehr
Center for Biological and Computational Learning
M.I.T.
Cambridge, MA 02139

End of VISION-LIST digest 13.11

Figure 5: A prototype of NIAGENT

nary experiments, NIAGENT achieves high precision in recalled web pages.

Table 1: Performance results

Natural Query	MetaCrawler Precision	NIAGENT Precision
How can we use *constraints* to develop *intelligent agents?*	35%	90%
How do *agents* reduce *information overload?*	65%	85%
How can we use *genetic algorithms* in solving *CSP?*	10%	95%

3 Related Work

In recent years, there has been a growing interest in applying NLP techniques to information retrieval on text-based documents. Soderland [11] develops Webfoot and CRYSTAL to create a formal representation of the web pages that is equivalent to relational database entries. Webfoot uses page layout cues to divide a web page into sentence-length segments of text that are passed to CRYSTAL to learn domain-specific text extraction rules from examples. The current experiments are limited to a specific domain of weather forecast web pages.

On the other hand, Air Travel Information System (ATIS) [5], developed at SRI International, combines speech recognition and natural language understanding technology that is capable of answering spoken queries in the domain of air travel.

4 Summary and Future Research

NIAGENT presents users with a intelligent and friendly interface that understands a user's natural query, translates the user's interests into appropriate queries for each search engine, analyzes the returned references and returns the relevant references. This agent also represents an example of cooperative computing in that it cooperates with other agents and servers to do search and filter web pages.

Our experience with NIAGENT suggests a number of topics for future research:

- More advanced NLP parser. Currently NIAGENT cooperates with MIT Chopper to understand the user's natural query. However, NIAGENT could cooperate with other NLP agents such as MIT NLP Parser to better understand a natural query by inferring the relations between the frames of the sentence and interpreting their actions. NIAGENT could search for relevant references more correctly by cooperating with such agents.

- Learning a user's actions. NIAGENT could learn a user's interests by the user's query. It could also learn the response of the user to the filtered web pages and see whether the user follows the hyperlinks of these pages or simply backtracks from them. In other words, NIAGENT could learn whether the user is interested in the contents of the filtered web pages or not and adjusts its behavior.

- Learning to understand the contents of the visited pages. NIAGENT could keep a user profile about the term frequency, document frequency of the filtered web pages. It could also use several learning algorithms such as Perkowitz and Etzinoi's ILA [9] and Soderland's CRYSTAL [11] to understand the information that the user is interested.

- Concept-based searching. NIAGENT could find more references by sending keywords, that the same concept of the input query, to keyword based Internet search engines in order to broaden the scope of the search space. Such search results would be not only relevant (because of NIAGENT's analysis) and broad (because of concept-based search).

Acknowledgments

We thank Dr. HsinChun Chen at University of Arizona for his valuable ideas and suggestions.

References

[1] H. Chen, Y. Chung, C. Yang, and M. Ramsey, "A Smart Itsy Bitsy Spider for the Web," *Journal of the American Society for Information Science*, forthcoming, 1997.

[2] W. B. Croft, H. R. Turtle and D. D. Lewis, "The Use of Phrases and Structured Queries in Information Retrieval," In *Proceedings of the Fourteenth Annual International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 32-45, Chicago, Oct. 1991.

[3] O. Etzinoi, "Moving Up the Information Food Chain: Deploying Softbots on the World Wide Web," In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1322-1326, 1996.

[4] `http://mu.www.media.mit.edu/groups/mu/phraser.html`, MIT Machine Understanding Group.

[5] L. Julia, A. Cheyer, L. Neumeyer, J. Dowding and M. Charefeddine, "HTTP://WWW.SPEECH.SRI.COM/DEMOS/ ATIS.HTML," In *Proceedings AAAI'97: Stanford*, pp. 72-76, July. 1997.

[6] T. Joachims, D. Freitag and T. Mitchell, "WebWatcher: A Tour Guide for the World Wide Web," In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, 1997.

[7] D. D. Lewis and K. S. Jones, "Natural Language Processing for Information Retrieval," *Communications of the ACM*, vol. 39, no. 1, pp. 92-101, Jan. 1996.

[8] P. Maes, "How to Do the Right Thing," *Connection Science Journal*, vol. 1, no. 1, pp. 291-323, Dec. 1989.

[9] M. Perkowitz and O. Etzioni, "Category Translation: Learning to Understand Information on the Internet," In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp., 1995.

[10] M. Pazzani, J. Muramatsu and D. Billsus, "Syskill & Webert: Identifying interesting Web Sites," In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 54-61, Portland, 1996.

[11] S. Soderland, "Learning to Extract Text-based Information from the World Wide Web," In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[12] E. Selberg and O. Etzioni, "Multi-Service Search and Comparison using the MetaCrawler," In *Proceedings of the fourth World Wide Web Conference*, pp. 21-70, Boston, Dec. 1995.

[13] E. Selberg and O. Etzioni, "The MetaCrawler Architecture for Resource Aggregation on the Web," *IEEE Expert*, pp. 11-14, Jan.-Feb. 1997.