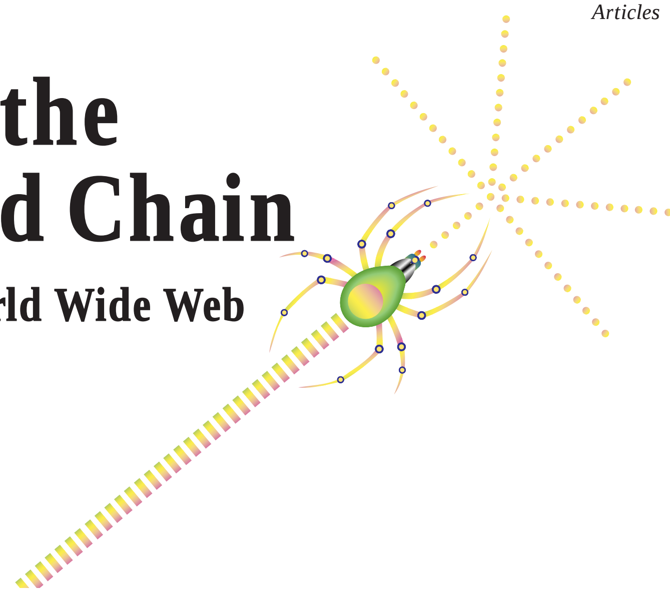
AI Magazine Volume 18 Number 2 (1997) (© AAAI)

Moving Up the Information Food Chain

Deploying Softbots on the World Wide Web

Oren Etzioni



■ I view the World Wide Web as an information food chain. The maze of pages and hyperlinks that comprise the Web are at the very bottom of the chain. The WEBCRAWLERS and ALTAVISTAS of the world are information herbivores; they graze on Web pages and regurgitate them as searchable indices. Today, most Web users feed near the bottom of the information food chain, but the time is ripe to move up. Since 1991, we have been building information carnivores, which intelligently hunt and feast on herbivores in UNIX, on the Internet, and on the Web. Information carnivores will become increasingly critical as the Web continues to grow and as more naive users are exposed to its chaotic jumble.

I view the World Wide Web as an information food chain (figure 1). The maze of pages and hyperlinks that comprise the Web are at the very bottom of the chain. The WEBCRAWLERS and ALTAVISTAS of the world are *information herbivores*; they graze on Web pages and regurgitate them as searchable indices. Today, most Web users feed near the bottom of the information food chain, but the time is ripe to move up. Since 1991, we have been building *information carnivores*, which intelligently hunt and feast on herbivores in UNIX (Etzioni, Lesh, and Segal 1993), on the Internet (Etzioni and Weld 1994), and on the Web (Doorenbos, Etzioni, and Weld 1997; Selberg and Etzioni 1997; Shakes, Langheinrich, and Etzioni 1997). I predict that information carnivores will become increasingly critical as the Web continues to grow and as more naive users are exposed to its chaotic jumble.

Motivation

Today's Web is populated by a panoply of primitive but popular information services. Consider, for example, an information cow such as ALTAVISTA. ALTAVISTA requires massive memory resources (to store an index of the Web) and tremendous network bandwidth (to create and continually refresh the index). The cost of these resources is amortized over millions of queries per day. As a result, the CPU cycles devoted to satisfying each individual query are sharply curtailed. There is no time for intelligence. Furthermore, each query is independent of the previous one. No attempt is made to customize ALTAVISTA's responses to a particular individual. The result is homogenized, least-common-denominator service.

In contrast, visionaries such as Alan Kay and Nicholas Negroponte have been advocating *agents*—personal assistants that act on your behalf in cyberspace. While the notion of agents has been popular for more than a decade, we have yet to build agents that are both widely used and intelligent. The Web presents a golden opportunity and an implicit challenge for the AI community. As the old adage goes "If not us, then who? And if not now, when?"

The challenge of deploying web agents will help revitalize AI and forge closer links with other areas of computer science. But be warned, the Web community is hungry, impatient, and skeptical. They expect *robustness*, a working system, accessible seven days a week, twenty-four hours a day; *speed*, virtually all widely-used Web resources begin transmitting useful (or at least entertaining) information within seconds; and *added value*,

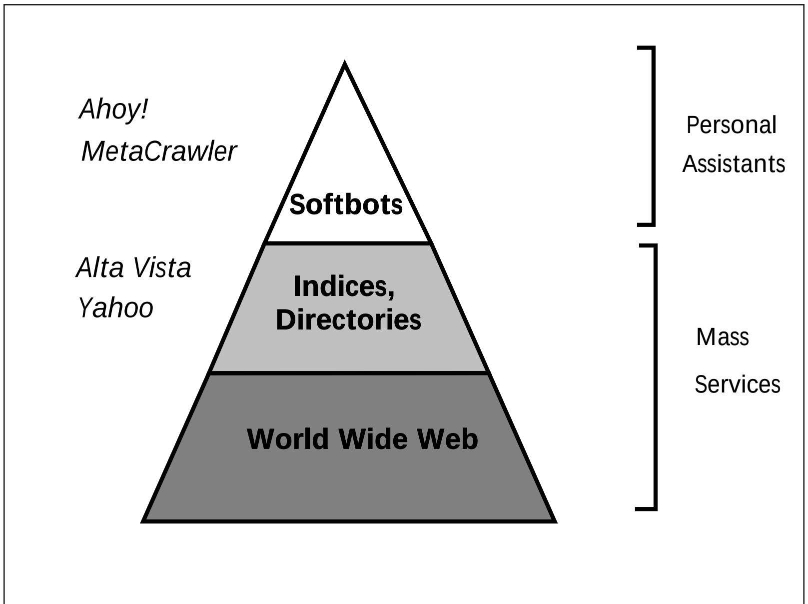


Figure 1. The Information Food Chain.

any increase in sophistication had better yield a tangible benefit to users.

Is the Web challenge a distraction from our long-term goal of understanding intelligence and building intelligent agents? I believe that the field benefits from a mixture of long-term and short-term goals and from both empirical and theoretical work. Work toward the goal of deploying intelligent agents on the Web is a valuable addition to the current mix for two reasons. First, the Web suggests new problems and new constraints on existing techniques. Second, intelligent Web agents will provide tangible evidence of the power and utility of AI techniques. Next time you encounter AI bashing, wouldn't it be satisfying to counter with a few well-chosen URLs?

Personally, I find the Web irresistible. To borrow Herb Simon's phrase, it is today's "Main Chance." Simon describes his move from the "academic backwater" of public administration to AI and cognitive psychology as "gravitating toward the sun" (Simon 1991, pp. 113–114). Although AI is not an academic backwater, the Web is today's sun. Turning towards the sun and responding to the Web challenge, my collaborators and I have begun to deploy a species of information carnivores (called softbots) on the Web.

Softbots

Softbots (software robots) are intelligent agents that use software tools and services on a person's behalf (figure 2). Tool use is one of the hallmarks of intelligence. In many cases, softbots rely on the same tools and utilities available to human computer users—tools for sending mail, printing files, and so on. Mobile robots have yet to achieve the physical analog—using vacuum cleaners, lawn mowers, etc. Softbots are an attractive substrate for intelligent-agent research for the following reasons (Etzioni 1994, 1993). First, the cost, effort, and expertise necessary to develop and systematically experiment with software artifacts are relatively low. Second, software environments circumvent many of the thorny but peripheral problems that are inescapable in physical environments. Finally, in contrast to simulated physical worlds, software environments are readily available (sophisticated simulations can take years to perfect), intrinsically interesting, and real. However, Softbots are not intended to replace robots; Robots and softbots are complimentary.

Much of our work has focused on the Internet softbot (also known as RODNEY) (Etzioni and Weld 1994). RODNEY enables a person to state what he or she wants accomplished. RODNEY disambiguates the request and dynamically determines how and where to satisfy it, utilizing a wide range of Internet services and UNIX commands. RODNEY relies on a declarative representation of the different software tools at its disposal, enabling it to chain together multiple tools in response to a user's request. RODNEY uses automatic planning technology to dynamically generate the appropriate action sequence. The Internet softbots project has led to a steady stream of technical results (Etzioni, Golden, and Weld 1997, 1994; Kwok and Weld 1996; Perkowitz and Etzioni 1995; Golden, Etzioni, and Weld 1994; Etzioni et al. 1992). Closely related projects include Kirk et al. (1995) and Arens et al. (1993).

Unfortunately, we have yet to produce a planner-based softbot that meets the stringent demands of the Web community. While continuing our ambitious long-term project to develop planner-based softbots, we have embraced a new strategy for the creation of intelligent agents which I call "useful first." Instead of starting with grand ideas about intelligence and issuing a promissory note that they will eventually yield useful intelligent agents, we take the opposite tack; we begin with useful softbots deployed on the Web, and issue a promissory note that they will evolve into more intelligent agents. We are still committed to the goal of producing agents that are both intelligent and useful. However, I submit that we are more likely to achieve this conjunctive goal if we reverse the

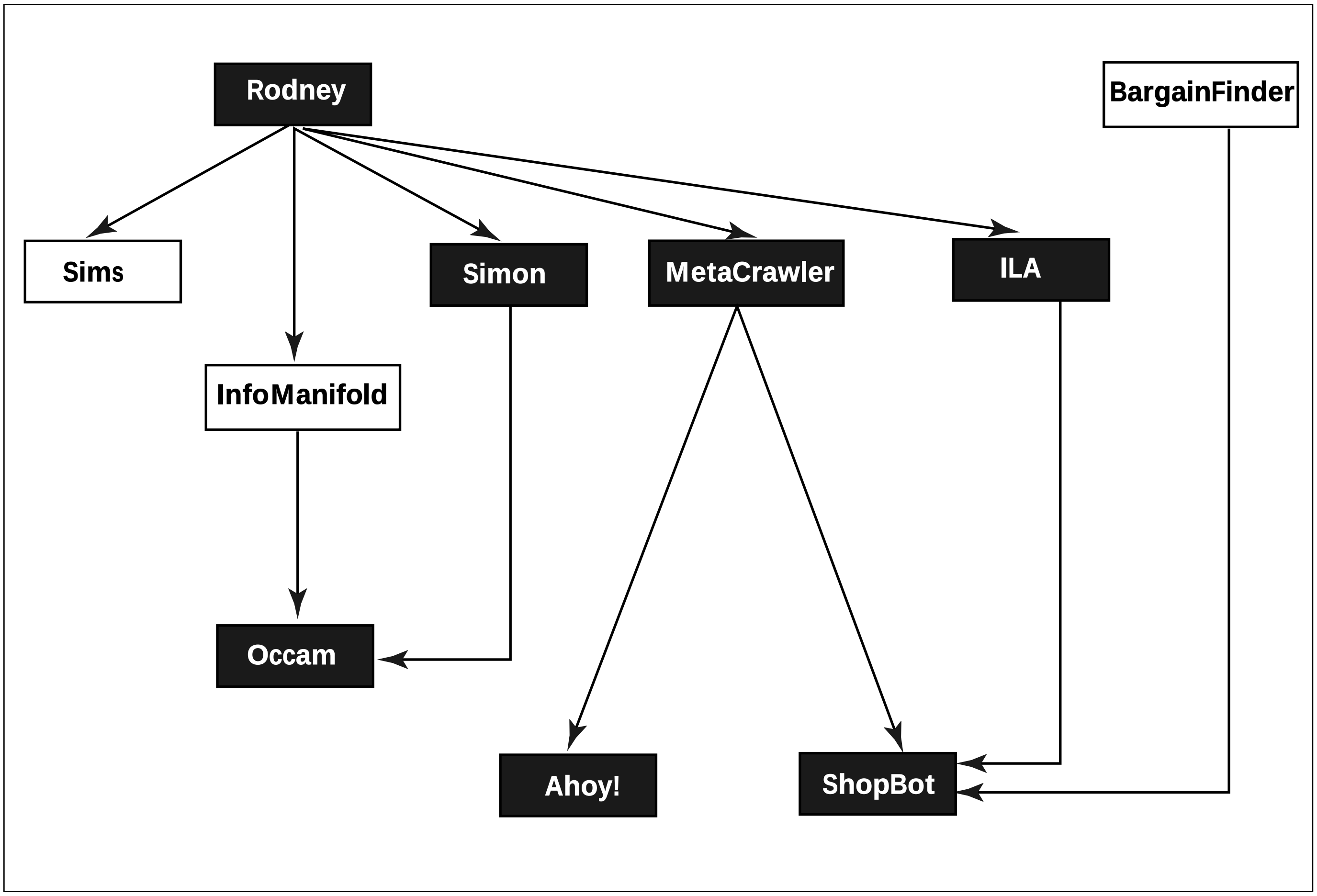


Figure 2. The Softbot Family Tree.

The black boxes represent work at the University of Washington; the white boxes represent softbots currently deployed on the Web.

traditional sub-goal ordering and focus on building useful systems first.

The argument for "useful first" is analogous to the argument made by Rod Brooks (1991) and others (Etzioni 1993; Mitchell et al. 1990) for building complete agents and testing them in a real world. As Brooks put it, "with a simplified world…it is very easy to accidentally build a submodule of the systems which happens to rely on some of those simplified properties…the disease spreads and the complete system depends in a subtle way on the simplified world" (p. 150). This argument applies equally well to user demands and real-time constraints on Web agents.

There is a huge gulf between an AI prototype and an agent ready for deployment on the Web. One might argue that this gulf is of no interest to AI researchers. However, the demands of the Web community constrain the AI techniques we use, and lead us to new AI problems. We need to recognize that intelligent agents are ninety-nine percent computer science and one percent AI. The AI is critical but we cannot ignore the context into which it is embedded. Patrick Winston has called this the "raisin bread" model of AI. If we want to bake raisin bread, we cannot focus exclusively on the raisins.[1]

Operating on a shoestring budget, we have been able to deploy several softbots on the Web. I review our fielded softbots and then consider both the benefits and pitfalls of the "useful first" approach.

The METACRAWLER softbot (see www.cs.washington.edu/research/metacrawler) provides a single, unified interface for Web document searching (Selberg and Etzioni 1997, 1995). METACRAWLER supports an expressive query language that allows searching for documents that contain certain phrases and excluding documents containing other phrases. METACRAWLER queries five of the most popular information herbivores in parallel. Thus, METACRAWLER eliminates the need for users to try and re-try queries across different herbi-

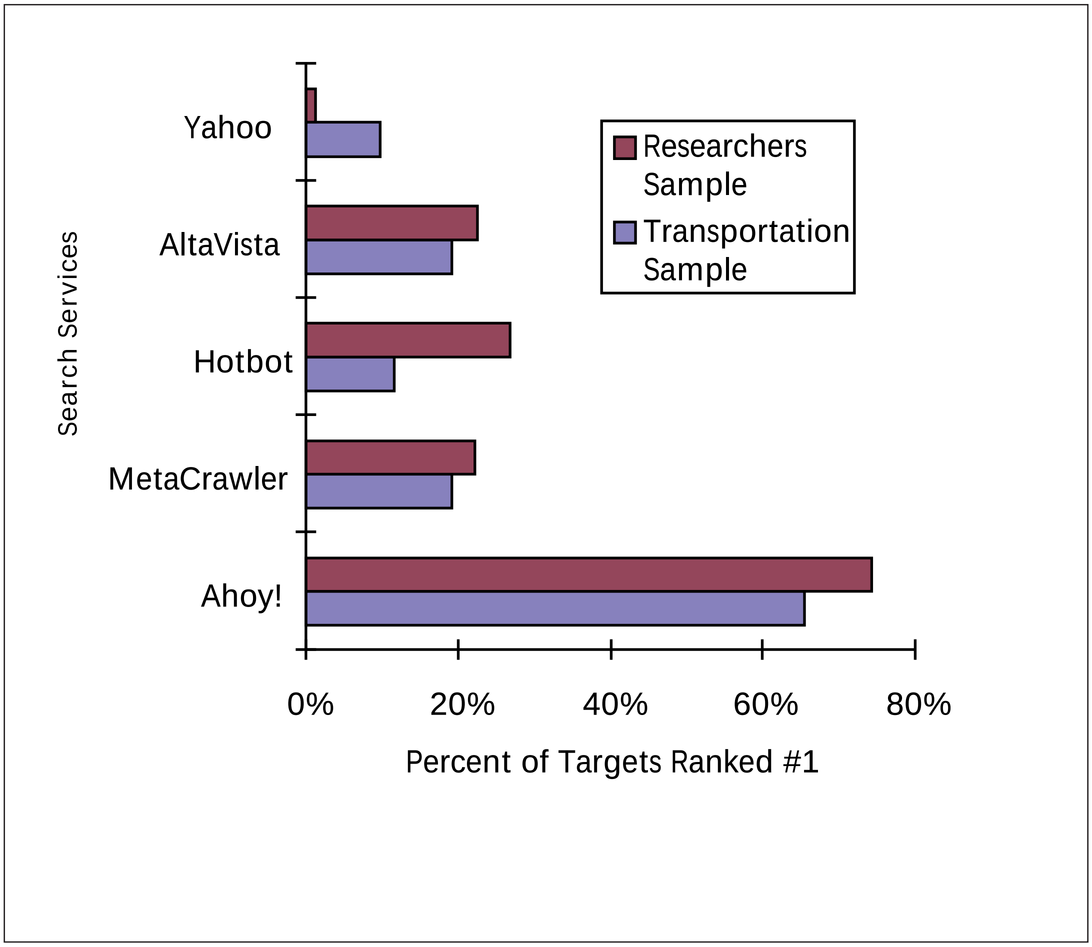


Figure 3. A Comparison of AHOY*!'s Accuracy or "Precision" with General-Purpose Search Engines on Two Test Samples.*

See Shakes, Langheinrich, and Etzioni (1997) for the details.

vores. Furthermore, users need not remember the address, interface and capabilities of each one. Consider searching for documents containing the phrase "four score and seven years ago." Some herbivores support phrase searching whereas others do not. METACRAWLER frees the user from having to remember such details. If a herbivore supports phrase searching, METACRAWLER automatically invokes this feature. If a herbivore does not support phrase searching, METACRAWLER automatically downloads the pages returned by that herbivore and performs its own phrase search locally.[2]

In a recent article, *Forbes* Magazine asked Lycos's Michael Maudlin "why aren't the other spiders as smart as METACRAWLER?" Maudlin replied "with our volume I have to turn down the smarts...METACRAWLER will too if it gets much bigger." Maudlin's reply misses an important point: because METACRAWLER relies on information herbivores to do the resource-intensive grazing of the Web, it is sufficiently lightweight to run on an average PC and serve as a personal assistant. Indeed, METACRAWLER-inspired PC applications such as WEBCOMPASS and Internet FASTFIND are now on the market.

METACRAWLER demonstrates that Web services and their interfaces may be de-coupled. METACRAWLER is a meta-interface with three main benefits. First, the same interface can be used to access multiple services simultaneously. Second, since the meta-interface has relatively modest resource requirements it can reside on an individual user's machine, which facilitates customization to that individual. Finally, if a meta-interface resides on the user's machine, there is no need to "turn down the smarts." In a Web-mediated client/server architecture, where intelligence resides in the client, "volume" is no longer a limiting factor on the "smarts" of the overall system.

While METACRAWLER does not currently use AI techniques, it is evolving rapidly. For example, we have developed a novel document clustering algorithm that enables users to rapidly focus on relevant subsets of the references returned by METACRAWLER (Zamir et al. 1997). In addition, we are investigating mixed-initiative dialog to help users focus their search. Most important, METACRAWLER is an enabling technology for softbots that are perched above it in the information food chain.

AHOY! The Home Page Finder

Robot-generated Web indices such as ALTAVISTA are comprehensive but imprecise; manually generated directories such as YAHOO! are precise but cannot keep up with large, rapidly growing categories such as personal homepages or news stories on the American economy. Thus, if a person is searching for a particular page that is not cataloged in a directory, she is forced to query a web index and manually sift through a large number of responses. Furthermore, if the page is not yet indexed, then the user is stymied. In response we have developed *Dynamic Reference Sifting*—a novel architecture that attempts to provide both maximally comprehensive coverage and highly precise responses in real time, for specific page categories (Shakes, Langheinrich, and Etzioni 1997).

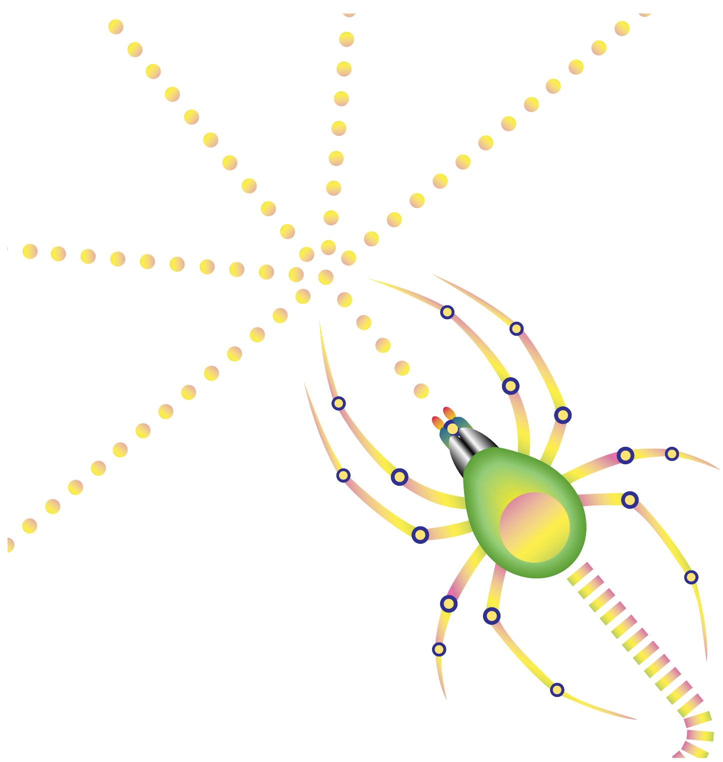
The AHOY! softbot (see www.cs.washington.edu/research/ahoy) embodies dynamic reference sifting for the task of locating people's home pages on the Web. AHOY! takes as input a person's name and affiliation, and attempts to find the person's home page. AHOY! queries METACRAWLER and uses knowledge of Web geography (e.g., the URLs of home pages at the University of Washington end with washington.edu) and home page appearance to filter METACRAWLER's output. Typically, AHOY! is able to cut the number of references returned by a factor of forty but still maintain very high accuracy (see Figure 3).

To improve its coverage, AHOY! learns from experience. It rapidly collects a set of home pages to use as training data for an unsupervised learning algorithm that attempts to discover the conventions underlying home page placement at different institutions. For example, AHOY! learns that home pages at the University of Washington's Computer Science Department typically have the form www.cs.washington.edu/homes/<lastname>. After learning, AHOY! is able to locate home pages of individuals even if they are not indexed by METACRAWLER's herd of information herbivores. In our experiments, 9% of the home pages located by AHOY! were found using learned knowledge of this sort.

In the context of AHOY! the "useful first" constraint led us to tackle an important impediment to the use of machine learning on the Web. Data is abundant on the Web, but it is unlabeled. Most concept learning techniques require training data labeled as positive (or negative) examples of some concept. Techniques such as uncertainty sampling (Lewis and Gale 1994) reduce the amount of labeled data needed, but do not eliminate the problem. Instead, AHOY! attempts to harness the Web's interactive nature to circumvent the labeling problem. AHOY! relies on its initial power to draw numerous users to it; AHOY! uses its experience with these users' queries to make generalizations about the Web, and improve its performance.[3] Note that by relying on experience with multiple users, AHOY! rapidly collects the data it needs to learn; systems that are focused on learning an individual user's taste do not have this luxury. AHOY!'s bootstrapping architecture is not restricted to learning about home pages; the architecture can be used in a variety of Web domains.

SHOPBOT

SHOPBOT is a softbot that carries out comparison shopping at Web vendors on a person's behalf (Doorenbos, Etzioni, and Weld 1997). Whereas virtually all previous Web agents rely on hard-coded interfaces to the Web sites they access, SHOPBOT autonomously learns to extract product information from Web vendors given



their URL and general information about their product domain (e.g., software). Specifically, SHOPBOT learns how to query a store's searchable product catalog, learns the format in which product descriptions are presented, and learns to extract product attributes such as price from these descriptions.

SHOPBOT's learning algorithm is based in part on that of the Internet Learning Agent (ILA) (Perkowitz and Etzioni 1995). ILA learns to extract information from unfamiliar sites by querying with familiar objects and analyzing the relationship of output tokens to the query object. SHOPBOT borrows this idea from ILA; SHOPBOT learns by querying stores for information on popular products, and analyzing the stores' responses. Whereas ILA learns to understand the meaning of tokens returned by unfamiliar web sites, SHOPBOT learns the format in which items are presented.

In the software shopping domain, SHOPBOT has been given the home pages for 12 on-line software vendors. After its learning is complete, SHOPBOT is able to speedily visit the vendors, extract product information such as availability and price, and summarize the results for the user. In a preliminary user study, SHOPBOT users were able to shop four times faster (and find better prices!) than users relying only on a Web browser (Doorenbos, Etzioni, and Weld 1997).

NETBOT JANGO

NETBOT JANGO is a softbot (see www.jango.com) that fuses and extends ideas and technologies from METACRAWLER, AHOY!, and SHOPBOT to create a single, unified entry point for on-line shopping in over twenty product categories. JANGO searches the Web for information to help a person decide what to buy and where to buy it; in response to a query, JANGO will visit manufacturer sites for product descriptions, magazine sites for product reviews, and vendor sites for price quotes. Unlike most softbots, JANGO goes beyond information gathering and acts on a person's behalf. When visiting a vendor, JANGO will automatically create an account and a password (saving these for future visits!). Furthermore, JANGO can be instructed to buy a product at the click of a mouse.

Consider the following example. A person queries JANGO with "Adobe Photoshop on CD-ROM." The softbot deduces that this is a request about computer software whose title is "Photoshop" whose manufacturer is "Adobe" and whose medium is "CD-ROM". In response, JANGO visits Adobe's home page, Ziff-Davis Publications, and numerous online merchants selling computer software (e.g. CDW, Internet Shopping Network, etc.). JANGO collates the information it receives, divides it into reviews and price quotes, eliminates irrelevant information, and presents the result to the user. At this point JANGO can be instructed to buy the product using the billing and shipping information in its user's personal profile.

JANGO consists of four key components: a natural language front-end, a query router, an aggregation engine, and a filter. The front-end takes a natural language request such as "I want that book by Bill Gates" or "find me a Woody Allen movie starring Jody Foster on VHS" and parses it into a logical description of the desired product's attributes, which is passed to the query router. The router examines this logical description and determines the set of web resources it is familiar with that contain information relevant to the query. To make this decision, the router relies on efficient taxonomic reasoning and a massive knowledge base to identify the precise category of the product—is it a cigar, a movie title, or a classical music CD? The router associates each product category and subcategory with a set of Web sites that are likely to contain information relevant to the category. Once the product's exact category is identified the router returns the set of relevant sites it is familiar with.

JANGO's aggregation engine is very similar to METACRAWLER. Based on the router's output the aggregation engine queries the relevant sites in parallel, and collates their responses. JANGO extends the METACRAWLER architecture by introducing declarative "wrappers" that enable it to communicate with hundreds of web sites. The wrappers specify how to query a site and how to parse the information returned in response. Finally, the filter module is invoked to perform the kind of stringent analysis done by AHOY! to strip and discard any information irrelevant to the query before a report is generated for the user.

In the future, planning techniques of the sort pioneered in the Internet Softbot will be used to generate multi-step plans that compose the results of queries to multiple web sites in order to obtain a precise response to a query. Finally, JANGO's wrappers can be generated automatically using the machine learning techniques introduced in ILA and SHOPBOT (see also Kushmerick [1997]).

Discussion

Every methodology has both benefits and pitfalls; the softbot paradigm is no exception.

Perhaps the most important benefit has been the discovery of new research challenges, the imposition of tractability constraints on AI algorithms, and the resulting innovations. In recent years, planner-based softbots have led us to the challenge of incorporating information goals, sensory actions, and closed world reasoning into planners in a tractable manner. Our focus on tractability led us to formulate UWL (Etzioni et al. 1992) and Local Closed World Reasoning (Etzioni, Golden, and Weld 1997, 1994). We expect "useful first" to be equally productive over the next few years.

I acknowledge that our approach has numerous pitfalls. Here are a couple, phrased as questions: will we fail to incorporate substantial intelligence into our softbots? Does the cost of deploying softbots on the Web outweigh the benefit? Our preliminary success in incorporating AI techniques into our deployed softbots makes me optimistic, but time will tell.

Conclusion

Each of the softbots described above uses multiple Web tools or services on a person's behalf. Each softbot enforces a powerful abstraction: a person is able to state what they want, the softbot is responsible for deciding which Web services to invoke in response and how to do so. Each softbot has been deployed on the Web, meeting the requirements of robustness, speed, and added value. Currently, METACRAWLER receives over 220,000 queries a day and has been licensed to an Internet Startup (Go2net, Inc.). Due to its narrower scope, AHOY! receives only several thousand queries per day. However, a recent *New York Times* article called it "the most reliable single tool for tracking down home pages." Finally, NETBOT JANGO can be downloaded from the Web, but no usage statistics are available as this article goes to press.

Having satisfied the "useful first" constraint, our challenge is to make our current softbots more intelligent, inventing new AI techniques and extending familiar ones. We are committed to doing so while keeping our softbots both usable and useful. If we succeed, we will help to rid AI of the stereotype "if it works, it ain't AI." To check on our progress, visit the URLs mentioned earlier. Softbots are standing by...

Acknowledgments

This article is loosely based on an invited talk presented at AAAI-96. I would like to thank Dan Weld, my close collaborator on many of the softbots described above, for his numerous contributions to the softbots project and its vision; he cannot be held responsible for the polemic tone and subversive methodological ideas of this piece. I would also like to thank members of the University of Washington (UW) Softbots Group and the folks at Netbot, Inc. for making softbots real. Special thanks are due to Erik Selberg who created METACRAWLER as part of his masters dissertation. Thanks are due to Steve Hanks and other members of the UW AI group for helpful discussions and collaboration. This research was funded in part by Office of Naval Research grant 92-J-1946, by ARPA / Rome Labs grant F30602-95-1-0024, by a gift from Rockwell International Palo Alto Research, and by National Science Foundation grant IRI-9357772.

Notes

1. See Brachman (1992) for an account of the massive re-engineering necessary to transform an "intelligent first" knowledge representation system into a usable one.

2. Like most systems deployed on the web, METACRAWLER is a moving target; it has been licensed to Go2net, Inc. and they have been adding and removing features at their discretion.

3. An early version of AHOY! solicited explicit user feedback on whether AHOY! returned the correct answer to a user query. Currently, AHOY! relies on an unsupervised learning algorithm in which it automatically labels its own training examples as success or failure.

References

Arens, Y.; Chee, C. Y.; Hsu, C.-N.; and Knoblock, C. A. 1993. Retrieving and Integrating Data from Multiple Information Sources. *International Journal of Intelligent and Cooperative Information Systems* 2(2): 127–158.

Brachman, R. 1992. "Reducing" CLASSIC to Practice: Knowledge Representation Theory Meets Reality. In Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning. San Francisco, Calif.: Morgan Kaufmann.

Brooks, R. 1991. Intelligence without Representation. *Artificial Intelligence* 47(1–3): 139–159.

Doorenbos, R.; Etzioni, O.; and Weld, D. 1997. A Scalable Comparison-Shopping Agent for the World-Wide Web. In Proceedings of Autonomous Agents, 39–48. New York: Association of Computing Machinery.

Etzioni, O. 1994. Etzioni Replies. *AI Magazine* 15(2): 13–14.

Etzioni, O. 1993. Intelligence without Robots: A Reply to Brooks. *AI Magazine* 14(4): 7–13.

Etzioni, O., and Weld, D. 1994. A Softbot-Based Interface to the Internet. *Communications of the ACM* 37(7): 72–76.

Etzioni, O.; Golden, K.; and Weld, D. 1997. Sound and Efficient Closed-World Reasoning for Planning. *Artificial Intelligence* 89(1–2): 113–148.

Etzioni, O.; Golden, K.; and Weld, D. 1994. Tractable Closed-World Reasoning with Updates. In Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning, 178–189. San Francisco, Calif.: Morgan Kaufmann.

Etzioni, O.; Lesh, N.; and Segal, R. 1993. Building Softbots for UNIX (Preliminary Report), Technical Report, 93-09-01, University of Washington.

Etzioni, O.; Hanks, S.; Weld, D.; Draper, D.; Lesh, N.; and Williamson, M. 1992. An Approach to Planning with Incomplete Information. In Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, 115–125. San Francisco, Calif.: Morgan Kaufmann.

Golden, K.; Etzioni, O.; and Weld, D. 1994. Omnipotence without Omniscience: Sensor Management in Planning. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 1048–1054. Menlo Park, Calif.: American Association for Artificial Intelligence.

Kirk, T.; Levy, A.; Sagiv, Y.; and Srivastava, D. 1995. The Information Manifold. Presented at the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, 27–29 March, Stanford, California.

Kushmerick, N. 1997. Wrapper Construction for Information Extraction. Ph.D. thesis, Dept. of Computer Science, Univ. of Washington.

Kwok, C., and Weld, D. 1996. Planning to Gather Information. Technical Report, 96-01-04, Department of Computer Science and Engineering, University of Washington.

Lewis, D., and Gale, W. 1994. Training Text Classifiers by Uncertainty Sampling. In Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: Association of Computing Machinery.

Mitchell, T. M.; Allen, J.; Chalasani, P.; Cheng, J.; Etzioni, O.; Ringuette, M.; and Schlimmer, J. C. 1990. THEO: A Framework for Self-Improving Systems. In *Architectures for Intelligence*, ed. K. VanLehn, 323–356. Hillsdale, N.J.: Lawrence Erlbaum.

Perkowitz, M., and Etzioni, O. 1995. Category Translation: Learning to Understand Information on the Internet. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, 930–936. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Selberg, E., and Etzioni, O. 1997. The METACRAWLER Architecture for Resource Aggregation on the Web. *IEEE Expert* 12(1): 8–14.

Selberg, E., and Etzioni, O. 1995. Multi-Service Search and Comparison Using the MetaCrawler. In Proceedings of the Fourth International World Wide Web Conference, 11–14 December, Boston, Massachusetts.

Shakes, J.; Langheinrich, M.; and Etzioni, O. 1997. AHOY! The Home Page Finder. In Proceedings of the Sixth International World Wide Web Conference, 7–11 April, Santa Clara, California.

Simon, H. 1991. *Models of My Life.* New York: Basic Books.

Zamir, O.; Etzioni, O.; Madani, O.; and Karp, R. M. 1997. Fast and Intuitive Clustering of Web Documents. Submitted to the Third International Conference on Knowledge Discovery and Databases (KDD-97), 14–17 August, Newport Beach, California. Forthcoming.

Oren Etzioni received his bachelor's degree in computer science from Harvard University in June 1986, and his Ph.D. from Carnegie Mellon University in January 1991. He joined the University of Washington as assistant professor of computer science and engineering in February 1991, and was promoted to an associate professor in 1996. In 1996 he founded Netbot Inc. with Dan Weld.

In the fall of 1991, he launched the Internet Softbots project. In 1993, Etzioni received an NSF Young Investigator Award. In 1995, the Internet Softbot was chosen as one of 5 finalists in the National DISCOVER Awards for Technological Innovation in Computer Software. He has published more than 50 technical papers on intelligent software agents, machine learning, planning, etc. His group's work on software agents has been featured in *The Economist, Business Week, Discover* Magazine, *Forbes* Magazine, *CyberTimes*, and *The New Scientist.* His e-mail address is etzioni@cs.washington.edu.