Appears in *Comm. of ACM*, Nov.'96

The World Wide Web: quagmire or gold mine?

Oren Etzioni
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195
etzioni@cs.washington.edu
http://www.cs.washington.edu/homes/etzioni/

1 MOTIVATION

Web mining is the use of data mining techniques to automatically discover and extract information from World Wide Web documents and services (*e.g.*, on-line travel agents, job listings, electronic malls, etc.). This article considers the question: is effective Web mining possible?

Skeptics believe that the Web is too unstructured for Web mining to succeed. Indeed, data mining has been applied to databases traditionally, yet much of the information on the Web lies buried in documents designed for human consumption such as home pages or product catalogs. Furthermore, much of the information on the Web is presented in natural language text with no machine-readable semantics; HTML annotations structure the *display* of Web pages, but provide little insight into their *content.*

Some have advocated transforming the Web into a massive layered database to facilitate data mining[12], but the Web is too dynamic and chaotic to be tamed in this manner. Others have attempted to hand code site-specific "wrappers" that facilitate the extraction of information from individual Web resources (*e.g.*, [8]). Hand coding is convenient but cannot keep up with the explosive growth of the Web. As an alternative, this article argues for the:

Structured Web Hypothesis: *Information on the Web is sufficiently structured to facilitate effective Web mining.*

Examples of Web structure include linguistic and typographic conventions, HTML annotations (*e.g.*,<title>), classes of semi-structured documents (*e.g.*, product catalogs), Web indices and directories, and much more. To support the Structured Web Hypothesis, this article will survey preliminary Web mining successes and suggest directions for future work.

Web mining may be decomposed into the following subtasks:

1. **Resource Discovery:** locating unfamiliar documents and services on the Web.
2. **Information Extraction:** automatically extracting specific information from newly discovered Web resources.
3. **Generalization:** uncovering general patterns at individual Web sites and across multiple sites.

I now consider each task in turn.

2 RESOURCE DISCOVERY

Web resources fall into two classes: documents and services. The bulk of the work on resource discovery focuses on the automatic creation of searchable indices of Web documents. The most popular indices have been created by Web robots such as WebCrawler and Alta Vista, which scan millions of Web documents and store an index of the words in the documents. A person can then ask for all the indexed documents that contain certain keywords. There are over a dozen different indices currently in active use, each with a unique interface and a database covering a different fraction of the Web. As a result, people are forced to repeatedly try and retry their queries across different indices. Furthermore, the indices return many responses that are irrelevant, outdated, or unavailable, forcing the person to manually sift through the responses searching for useful information.

MetaCrawler (`http://www.metacrawler.com`) represents the next level up in the information "food chain" by providing a single, unified interface for Web document searching.[1] MetaCrawler's expressive query language allows searching for phrases and restricting the search by geographic region or by Internet domain (*e.g.*, `.gov`). MetaCrawler posts keyword queries to nine searchable indices in parallel; it then collates and prunes the responses returned, aiming to provide users with a manageable amount of high-quality information. Thus, instead of tackling the Web directly, MetaCrawler "mines" robot-created searchable indices.

Future resource discovery systems will make use of automatic text categorization technology to classify Web documents into categories. This technology could facilitate the automatic construction of Web directories such as Yahoo by discovering documents that fit Yahoo categories. Alternatively, the technology could be used to filter the results of queries to searchable indices. For example, in response to a query such as "Find me product reviews of Encarta", a discovery system could take documents containing the word "Encarta" found by querying searchable indices, and identify the subset that corresponds to product reviews.

[1]The information food chain metaphor is elaborated in [4]; Web robots are viewed as herbivores, and MetaCrawler as an information carnivore.

3 INFORMATION EXTRACTION

Once a Web resource has been discovered, the challenge is to automatically extract information from it. The bulk of today's information-extraction systems identify a fixed set of Web resources and rely on hand coded "wrappers" to access the resource and parse its response. To scale with the growth of the Web, Web miners need to dynamically extract information from *unfamiliar* resources, thereby eliminating or reducing the need for hand coding. We survey several such systems below.

The Harvest system relies on models of semi-structured documents to improve its ability to extract information [1]. For example, it knows how to find author and title information in Latex documents and how to strip position information from Postscript files. In one demonstration, Harvest created a directory of toll-free numbers by extracting them from a large set of web documents (see `http://harvest.cs.colorado.edu/harvest/demobrokers.html`). Harvest neither discovers new documents nor learns new models of document structure. However, Harvest easily handles new documents of a familiar type.

FAQ-Finders extract answers to frequently asked questions (FAQs) from FAQ files available on the Web [6, 11]. Like Harvest, they rely on a model of document structure. People pose their question in natural language and the text of their question is used to search the FAQ files for a matching question; FAQ-Finder then returns the answer associated with the matching question. Because of the semi-structured nature of the files, and because the number of files is much smaller than the number of documents on the World Wide Web, FAQ-Finders have the potential to return higher quality information than general-purpose searchable indices.

Both Harvest and FAQ-Finder have two key limitations. First, both systems focus exclusively on Web documents and ignore services (the same holds Web indices as well). Second, both Harvest and FAQ-Finder rely on a pre-specified description of certain fixed classes of Web documents. In contrast, The Internet Learning Agent (ILA) and ShopBot are two Web miners (described below) that rely on a combination of test queries and domain-specific knowledge to automatically learn descriptions of Web *services* (*e.g.*, searchable product catalogs, personnel directories, and more). The learned descriptions can be used to enable automatic information extraction by intelligent agents such as the Internet Softbot [5].

ILA learns to extract information from unfamiliar resources by querying them with familiar objects and matching the output returned against knowledge about the query objects [10]. For example, ILA queries the University of Washington personnel directory with `Etzioni` and recognizes the third output token `685-3035` as his phone number. Based on this observation, ILA might hypothesize that the third token output by the directory is the phone number of the person mentioned in the query. This learning process has a number of subtleties. For example, the output token `oren` could be either Etzioni's userid or first name. To discriminate between these two competing hypotheses, ILA will attempt to query with someone whose userid is different from her first name. In the experiments reported in [10], ILA successfully learned to extract information such as phone numbers and e-mail addresses from the Internet server `Whois` and from the personnel directories of a dozen universities.

ShopBot learns to extract product information from Web vendors [3]. ShopBot borrows from ILA the idea of learning by querying with familiar objects. However, ShopBot tackles a more ambitious task. ShopBot takes as input the address of a store's home page as well as knowledge about a product domain (*e.g.*, software), and learns how to shop at the store. Specifically, ShopBot searches the store's Web to find the 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 learns by querying the store for information on popular products, and analyzing the store's responses. In the software shopping domain, ShopBot was given the home pages for 12 on-line software vendors. ShopBot learned to extract product information from each of the stores, including the product's operating system (Mac or Windows), and more. 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 [3]. Current work on ShopBot explores the problem of autonomously discovering vendor home pages.

4 GENERALIZATION

Once we have automated the discovery and extraction of information from Web sites, the natural next step is to attempt to generalize from our experience. Yet, virtually all machine learning systems deployed on the Web (see [7] for some examples) learn about their user's interests, instead of learning about the Web itself. A major obstacle to learning about the Web is the *labeling problem*: data is abundant on the Web, but it is unlabeled. Many data mining techniques require inputs labeled as positive (or negative) examples of some concept. For example, it is relatively straight forward to take a large set of Web pages labeled as positive and negative examples of the concept "home page" and derive a classifier that predicts whether any given Web page is a home page or not; unfortunately, Web pages are unlabeled.

Techniques such as uncertainty sampling [9] reduce the amount of labeled data needed, but do not eliminate the labeling problem. Clustering techniques do not require labeled inputs, and have been applied successfully to large collections of documents (e.g, [2]). Indeed, the Web offers fertile ground for document clustering research. However, because clustering techniques take weaker (unlabeled) inputs than other data mining techniques, they produce weaker (unlabeled) output. Below we consider an approach to solving the labeling problem that relies on the observation that the Web is much more than a collection of linked documents.

The Web is an interactive medium visited by millions of people each day. Ahoy! (`http://www.cs.wash` represents an attempt to harness this source of power to solve the labeling problem. Ahoy! takes as input a person's name and affiliation, and attempts to locate the person's home page. Ahoy! queries MetaCrawler and uses knowledge of institutions and home pages to filter MetaCrawler's output. Since Ahoy!'s filtering algorithm is heuristic, it asks its users to label its answers as correct or not. Ahoy! relies on its initial power to draw numerous users to it and to solicit their feedback; it then uses this feedback to solve the labeling problem, make

generalizations about the Web, and improve its performance. By relying on feedback from *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. Finally, note that Ahoy!'s boot-strapping architecture is not restricted to learning about home pages; user feedback may be harnessed to provide training data in a variety of Web domains.

5 CONCLUSION

In theory, the potential of Web mining to help people navigate, search, and visualize the contents of the Web is enormous. This brief and selective survey explored the question of whether effective Web mining is feasible in practice. We reviewed several promising prototypes and outlined directions for future work. In essence, we have gathered preliminary evidence for the Structured Web Hypothesis; although the Web is less structured than we might hope, it is less random than we might fear.

Acknowledgments

I would like to thank my close collaborator, Dan Weld, for his numerous contributions to the softbots project and its vision. I would also like to thank my co-softbotists David Christianson, Bob Doorenbos, Marc Friedman, Keith Golden, Nick Kushmerick, Cody Kwok, Neal Lesh, Mark Langheinrich, Sujay Parekh, Mike Perkowitz, Erik Selberg, Richard Segal, and Jonathan Shakes. 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.

References

[1] C. Mic Brown, Peter B. Danzig, Darren Hardy, Udi Manber, and Michael F. Schwartz. The Harvest information discovery and access system. In *Proceedings of the Second International World Wide Web Conference*, pages 763–771, 1994. available from ftp://ftp.cs.colorado.edu/pub/cs/techreports/schwartz/Harvest.Conf.ps.Z.

[2] D. Cutting, D. Karger, J. Pedersen, and J. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *15th Annual Int'l SIGIR92*, 1992.

[3] R. B. Doorenbos, O. Etzioni, and D. S. Weld. A scalable comparison-shopping agent for the world-wide web. Technical Report 96-01-03, University of Washington, Department of Computer Science and Engineering, January 1996. Available via FTP from pub/ai/ at ftp.cs.washington.edu.

[4] O. Etzioni. Moving up the information food chain: deploying softbots on the Web. In *Proc. 14th Nat. Conf. on AI*, 1996.

[5] O. Etzioni and D. Weld. A softbot-based interface to the Internet. *CACM*, 37(7):72–76, 1994. See `http://www.cs.washington.edu/research/softbots`.

[6] Kristen Hammond, Robin Burke, Charles Martin, and Steven Lytinen. FAQ finder: A case-based approach to knowledge navigation. In *Working Notes of the AAAI Spring Symposium: Information Gathering from Heterogeneous, Distributed Environments*, pages 69–73, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[7] Craig Knoblock and Alon Levy, editors. *Working Notes of the AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford University, 1995. AAAI Press. To order a copy, contact sss@aaai.org.

[8] B. Krulwich. The bargainfinder agent: Comparison price shopping on the internet. In J. Williams, editor, *Bots and Other Internet Beasties*. SAMS.NET, 1996. `http://bf.cstar.ac.com/bf/`.

[9] D. Lewis and W. Gale. Training text classifiers by uncertainty sampling. In *17th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[10] Mike Perkowitz and Oren Etzioni. Category translation: Learning to understand information on the Internet. In *Proc. 15th Int. Joint Conf. on AI*, pages 930–936, 1995.

[11] Steven D. Whitehead. Auto-faq: An experiment in cyberspace leveraging. In *Proceedings of the Second International WWW Conference*, volume 1, pages 25–38, Chicago, IL, 1994. (See also: `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/whitehead/whitehead.html`).

[12] O. R. Zaiane and Han Jiawei. Resource and knowledge discovery in global information systems: a preliminary design and experiment. In *KDD-95 Proceedings*, pages 331–336, 1995.