Text-Learning and Related Intelligent Agents: A Survey

Dunja Mladenic, J. Stefan Institute

IN SURVEYING CURRENT RESEARCH IN THE DEVELOPMENT OF TEXT-LEARNING INTELLIGENT AGENTS, THE AUTHOR FOCUSES ON THREE KEY CRITERIA: WHAT REPRESENTATION THE PARTICULAR APPLICATION USES FOR DOCUMENTS, HOW IT SELECTS FEATURES, AND WHAT LEARNING ALGORITHM IT USES. SHE THEN DESCRIBES PERSONAL WEBWATCHER, A CONTENT-BASED INTELLIGENT AGENT THAT USES TEXT-LEARNING FOR USER-CUSTOMIZED WEB BROWSING.

THERE SIMPLY AREN'T ENOUGH hours in the day anymore, are there? Everybody seems to have a stack of things waiting that just had to be done yesterday, and the problem only gets worse, not better. Information overload: it's the bane of our times. This impression seems so realistic that any help in handling at least some simple tasks is usually appreciated.

The Internet's recent ascendancy—not only in the research community but also in many areas of everyday life—is a major culprit. No longer confined to providing researchers access to data, the Internet is often the source of choice for information about everything from what's happening around the world, to where to get the best airline tickets, to how to cook a particular dish, to where to find the best hiking trails. It's wonderful, but it also causes headaches.

The impact on computer systems is particularly pronounced. When different people come together without centralized rules and guidance, many creative and pleasant, as well as some less pleasant, aspects emerge. So, when putting information on the World Wide Web, each of us can decide what to put there and how to organize it. The result is a distributed, world-wide-accessible information source that contains nonhomogeneous data organized according to different human association models and value schemes. While this personal touch and opportunity for creativity can be extremely useful for humans when providing and obtaining information, most computer systems have a hard time coping with the complexity, especially because people want information sooner rather than later.

The large amounts of information available in electronic form on the Web challenges the research community with its distributed organization; its arbitrary mixture of text, speech, image, and video in the same document; and the dynamic nature of the provided information. A number of systems have emerged for helping users browse the Web, some based on content analysis using mostly text from Web documents, and others relying on other information about document relevancy, such as user ratings.

Recent developments at the intersection of information retrieval and machine learning—as well as work in intelligent agents and intelligent user interfaces—offer novel solutions for helping users quickly select the information they want. A wide range of researchers are involved in the intensive development of methods for using machine-learning techniques on text databases, called *text learning*, which combine research in machine learning with information retrieval. This article surveys a part of text learning where supervised learning methods are used for text

1094-7167/99/$10.00 © 1999 IEEE

Table 1. Content-based approaches that use machine-learning techniques.

Agent	Where developed	Goal	Publication
Antagonomy	NEC	Personalized newspaper	T. Kamba, H. Sakagami, and Y. Koseki, "Anatagonomy: A Personalized Newspaper on the World Wide Web," *Int'l J. Human-Computer Studies*, Vol. 46, No. 6, June 1997, pp. 789–803.
Calendar Apprentice	CMU	Meeting scheduling	T. Mitchell et al., "Experience with a Learning Personal Assistant," *Comm. ACM,* Vol. 37, No. 7, July 1994, pp. 81–91.
CiteSeer	TX, NEC UMIACS	Finding papers on WWW	K. Bollacker, S. Lawrence, and L. Giles, "CiteSeer: An Autonomous System for Processing and Organizing Scientific Literature on the Web," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98*), Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml
ContactFinder	Andersen Consulting	Finding experts	B. Krulwich and C. Burkey, "The ContactFinder Agent: Answering Bulletin Board Questions with Referrals," *Proc. 13th Nat'l Conf. AI (AAAI 96)*, AAAI Press, Menlo Park, Calif., 1996, pp. 10–15.
FAQFinder	Chicago Univ.	Answering questions	R. Burke, K. Hammond, and J. Kozlovsky, "Knowledge-Based Information Retrieval for Semi-Structured Text," *Working Notes from AAAI Fall Symp. AI Applications in Knowledge Navigation and Retrieval*, AAAI Press, Menlo Park, Calif. 1995, pp. 19–24. R. Burke et al., "Question Answering from Frequently Asked Question Files," *AI Magazine*, Vol. 18, No. 2, Summer 1997, pp. 57–66.
Internet Fish	MIT	Find info on Internet	B.A. LaMacchia, "Internet Fish, A Revised Version of a Thesis Proposal," MIT, AI Lab and Dept. of Electrical Eng. and Computer Science, Cambridge, Mass., 1996.
Letizia	MIT	Browsing WWW	H. Lieberman, "Letizia: An Agent that Assists Web Browsing," *Proc. 14th Int'l Joint Conf. AI (IJCAI95)*, AAAI Press, Menlo Park, Calif., 1995, pp. 924–929.
Lira	Stanford	Browsing WWW	M. Balabanovic and Y. Shoham, "Learning Information Retrieval Agents: Experiments with Automated Web Browsing," *AAAI 1995 Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, Menlo Park, Calif., 1995.
Musag	Hebrew Univ.	Browsing WWW	C.V. Goldman, A. Langer, and J.S. Rosenschein, "Musag: An Agent That Learns What You Mean," *Applied AI*, Vol. 11, No. 5, 1997, pp. 413–435.
NewsWeeder	CMU	Usenet news filtering	K. Lang, "News Weeder: Learning to Filter Netnews," *Proc. 12th Int'l Conf. Machine Learning,* Morgan Kaufmann, San Francisco, 1995, pp. 331–339.
Personal WebWatcher	CMU, IJS	Browsing WWW	D. Mladenic, *Personal WebWatcher: Implementation and Design*, Tech. Report IJS-DP-7472, Dept. of Computer Science, J. Stefan Inst., 1996; http://cs.cmu.edu/~TextLearning/pww.
Syskill & Webert	UCI	Browsing WWW	M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying Interesting Web Sites," *Proc. 13th Nat'l Conf. AI AAAI 96*, AAAI Press, Menlo Park, Calif., 1996, pp. 54–61. M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning 27*, Kluwer Academic Publishers, Dorrdrecht, The Netherlands, 1997, pp. 313–331.
WAWA	Wisconsin	Browsing WWW	J. Shavlik and T. Eliassi-Rad, "Building Intelligent Agents for Web-based Tasks: A Theory-Refinement Approach," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98),* Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.
WebWatcher	CMU	Browsing WWW	R. Armstrong et al., "WebWatcher: A Learning Apprentice for the World Wide Web," *AAAI 1995 Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, Menlo Park, Calif., 1995.

classification, after which I discuss the Personal WebWatcher intelligent agent in more detail.

Machine learning for intelligent agents

Although there are various definitions of the term *intelligent agent*, I will focus on systems such as user assistants and recommendation systems that employ machine learning[1] or data-mining techniques.[2] These systems assist users by finding information or performing some simpler tasks on their behalf. For instance, such a system might help in Web browsing by retrieving documents similar to already-requested documents.[3]

Two frequently used methods for developing intelligent agents based on machine-learning techniques are *content-based* and *collaborative* approaches.[4] Both can help users find and retrieve relevant information from the Web.

Table 1 summarizes the systems I discuss. For each system, the table lists the system's name, the organization that developed it, a summary of its functionality, and a reference to the paper describing it. Many other intelligent agents not mentioned here have been developed, but my intent is to indicate general current trends.

The content-based approach. In this approach to text classification, the system searches for items similar to those the user prefers based on a comparison of content. The approach has its roots in information retrieval. This approach has difficulty, however, in capturing different aspects of the content—music, movies, and images, for example. Even for text domains, most representations capture only certain aspects of the content, which results in poor system performance. In addition to the representational problems, content-based systems tend to learn in a way that they recommend items similar to the already-seen items.

Applying the content-based approach on text

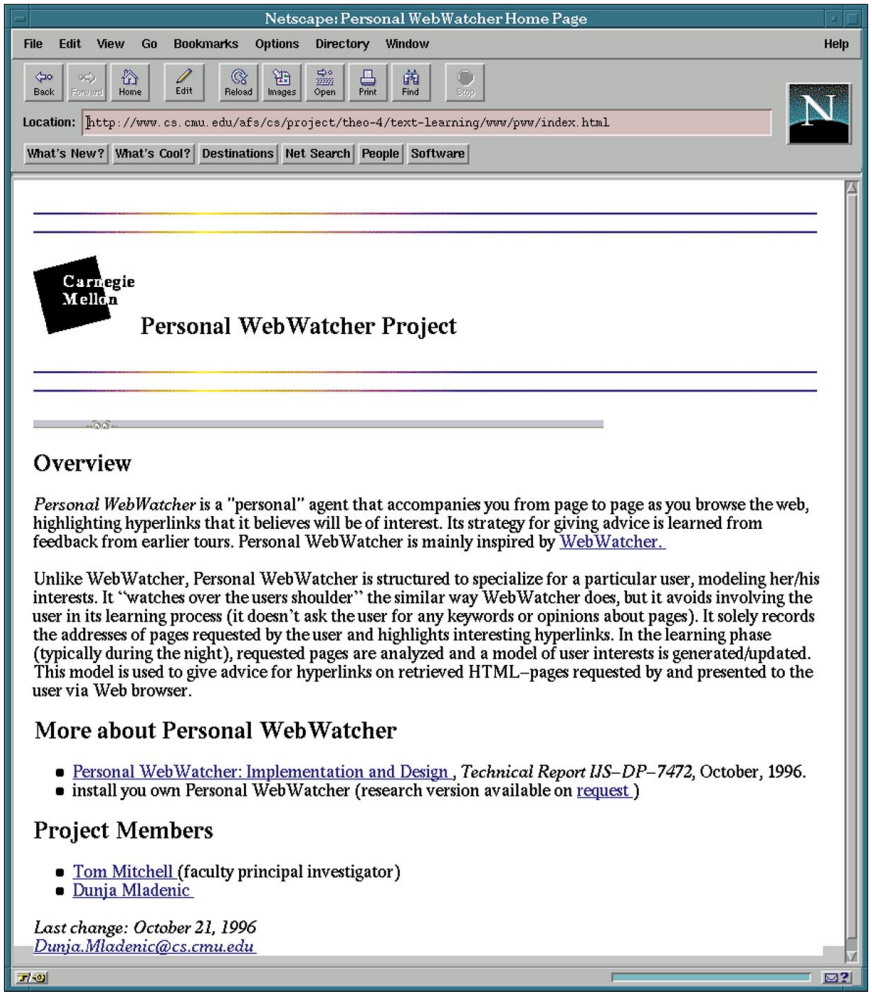

Figure 1. Example of the Web pages that a user found interesting.

data lets us use different text-learning methods. For instance, take the problem where we want to recommend Web pages to the user, based on their content. If the user is interested in the Web page shown in Figure 1, for instance, the Web page in Figure 2a will be recommended by the content-based approach because it has a similar content to the page in Figure 1. The Web page in Figure 2b would be recognized as different and thus would not be recommended (unless the user also found some other pages interesting that are similar to it).

The content-based approach is popular for systems that work on text data—for example, on Web documents or news. The WebWatcher system helps users locate information on the Web by taking keywords from users, suggesting hyperlinks, and receiving evaluation. It also lets users get additional similar documents. The data the system saves contains information about the search keywords typed by different users, hyperlinks that were followed, and the evaluations users gave at the end of each search.

The Lira system learns to browse the Internet on a user's behalf. It searches the Web by taking a bounded amount of time, selecting the best pages and receiving an evaluation from the user. Lira uses the evaluation to update the search and selection heuristics. The Musag system takes keywords from the user and searches the Web for relevant documents. The system generates a kind of thesaurus that relates concepts that are semantically similar to each other. Musag uses the generated thesaurus in document retrieval to extend a set of given keywords.

Letizia, a user-interface agent for assisting Web browsing, does not require any keywords or rating from the user because it infers the user's interests from browsing behavior. Web-browser users usually perform depth-first searches. While the user is reading a Web document, Letizia performs a breadth-first search from the current document. The system suggests potentially interesting hyperlinks found during the search, presented to the user in a separate browser window.

Personal WebWatcher, a personal assistant for Web browsing that accompanies the user from page to page and highlights interesting hyperlinks, generates a user profile based on the content analysis of the requested pages without requesting any keywords or ratings from the user. Syskill & Webert, a system that collects ratings of the explored Web pages from the user and learns a user profile from them, separates pages according to their topics, and learns a separate profile for each topic. The system uses the generated user profile to form queries for the existing search engines to get more potentially interesting documents. WAWA, an intelligent agent for Web-based tasks, lets users input personal interests and preferences, stores them in a neural network, and uses theory revision to refine the obtained knowledge.

CiteSeer helps users find relevant Web-based research publications by getting keywords from the user and calling search engines to find relevant papers (relevant PostScript files on the Web). It then extracts headers, abstracts, and citations from the papers. The system also finds similar papers based on the common citations in the papers. NewsWeeder, a system for electronic news filtering, uses text classification to generate a model of a user's interests. The system uses a Web interface to give the user access to the news in the usual way and to enable the system to collect the user's ratings as feedback. NewsWeeder also assigns predicted ratings to each article and generates a personalized list of the top articles (such as 50 articles predicted as the most interesting) found among all articles.

The proposed ContactFinder agent reads and responds to bulletin board messages, assists users by referring them to other people who can help them, and categorizes messages and extracts their topic areas. The system operates in two phases. It first searches the bulletin board looking for contact people and their topic areas, storing the found information in its database. Second, it searches new messages looking for questions, extracting the topic area of the found question and finding a contact person in its database for that topic area. To extract the topic area, ContactFinder uses some heuristics. For instance, it extracts semantically significant phrases (such as, fully capitalized words, short phrases of one to five words, or words in a different format from the surrounding text).

FAQFinder uses a natural-language question-based interface to access distributed text information sources and helps users find answers to their questions in databases such as FAQ files. It matches questions from relevant FAQ files against user questions and

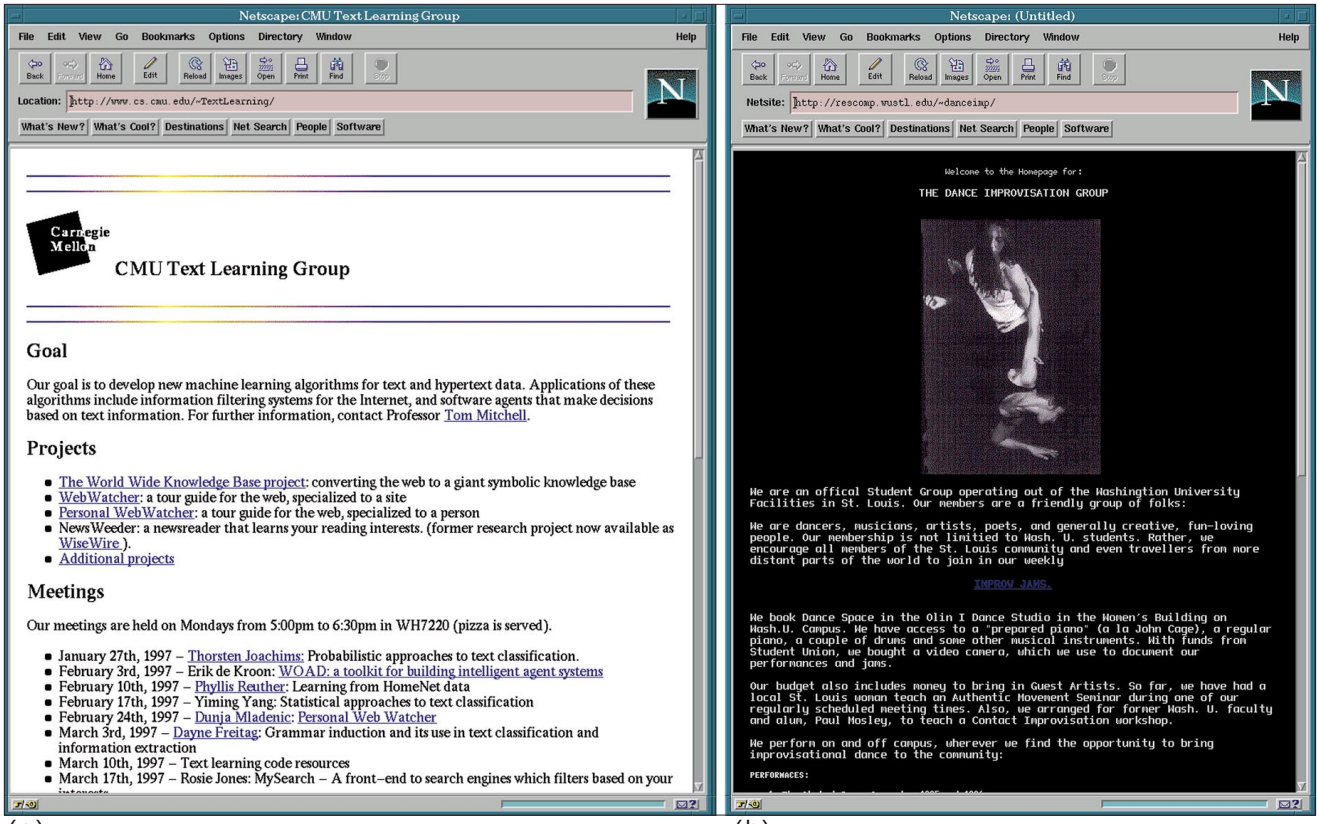

Figure 2. Example of two Web pages having different content and thus recognized as not similar by the content-based approach: (a) this page would be recognized as similar to the page in Figure 1 and recommended by the content-based approach; (b) this page would be recognized as different from both previous Web pages.

returns the five best matching questions together with their answers. Antagonomy, a system that composes personalized newspapers on the Web, monitors user operations on the articles and reflects them in the user profile. The layout of the composed newspaper is based on the scores given to articles that reflect the degree to which articles match the user profile—articles with the higher scores appear at the top of the newspaper.

Internet Fish is a class of resource-discovery tools designed to help users extract useful information from the Internet. The system includes a natural-language interface that currently permits only limited, structured interaction. The proposed system also includes help in browsing the Web by using existing search engines and user ratings of documents. Calendar Apprentice, a proposed system that helps users schedule meetings, connects to the user's electronic calendar and generates sets of rules that capture the user's scheduling preferences and other information about individual meeting attendees. It uses these rules to provide advice to the user for new, unscheduled meetings.

Music	Chopin	Bach	Matheny	Balasevic	Prodigy	Presley	ABBA	Enya
User1	6	7	5	7	1	2	3	7
User2	7	6	6	7	—	1	5	6
User3	2	1	1	—	7	6	6	3

Figure 3. Example of music ratings given by imaginary Web users, with 7 being the highest rating and '—' meaning no rating collected for that item. The collaborative approach would find User2 similar to User1, while User3 would be found different from User1.

The collaborative approach. In contrast to the content-based approach to text classification, which can be successfully applied to a single user, the collaborative approach assumes that there is a set of users using the system. In the collaborative approach (sometimes referred to as *social learning*[5]), advice to the user is based on the reaction of other users. The system searches for users with similar interests and recommends the items these users liked. Instead of computing the similarity between items, the system computes the similarity between users. In the collaborative approach, there is no analysis of the item content, so items of any content can be handled with equal success. Each item is assigned a unique identifier and a user-derived rating. The similarity rating between users is based on the comparison of the ratings they assigned to the same items.

With the collaborative approach, however, the small number of users relative to the number of items usually results in a sparse coverage of ratings. For any new item in the database, the system must collect information from different users to be able to recommend it, and similar users are not matched unless they have rated a sufficient number of similar items. Also, if a user has unusual tastes compared to the rest of the users, there will be no other similar user and system performance will be poor.

For instance, consider making a recommendation for User1, whose ratings for music are listed in Figure 3. The recommendation is based on the ratings of other users (User2 and

Table 2. Some collaborative approaches that use machine-learning-related techniques.

Agent	Where developed	Goal	Publication
Firefly, Ringo	MIT	Finding music, movie, book	P. Maes, "Agents that Reduce Work and Information Overload," *Comm. ACM,* Vol. 37, No. 7, July 1994, pp. 30–40.
GroupLense	Minnesota	Usenet news filtering	J.A. Konstan et al., "GroupLense: Applying Filtering to Usenet News," *Comm. ACM*, Vol. 40, No. 3, Mar. 1997, pp. 77–87.
Phoaks	AT&T Labs	Browsing WWW	T. Terveen et al., "PHOAKS: A System for Sharing Recommendations," *Comm. ACM*, Vol. 40, No. 3, Mar. 1997, pp. 59–62.
Referral Web	AT&T Labs	Finding experts	H. Kautz, B. Selman, and M. Shah, "Referral Web: Combining Social Networks and Collaborative Filtering," *Comm. ACM*, Vol. 40, No. 3, Mar. 1997, pp. 63–65. H. Kautz, B. Selman, and M. Shah, "The Hidden Web," *AI Magazine*, Vol. 18, No. 2, Summer 1997, pp. 27–36.
Siteseer	Imana	Browsing WWW	J. Rucker and J.P. Marcos, "Siteseer: Personalized Navigation for the Web," *Comm. ACM,* Vol. 40, No. 3, Mar. 1997, pp. 73–75.

Table 3. Systems that use both content-based and collaborative approaches.

Agent	Where developed	Goal	Publication
Fab	Stanford	Browsing WWW	M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Comm. ACM*, Vol. 40, No. 3, Mar. 1997, pp. 66–70.
Lifestyle Finder	AgentSoft	Browsing WWW	B. Krulwich, "Lifestyle Finder," *AI Magazine*, Vol. 18, No. 2, Summer 1997, pp. 37–46.
WebCobra	James Cook Univ.	Browsing WWW	O. de Vel and S.A. Nesbitt, "Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998: http://www.cs.cmu.edu/~conald/conald.shtml.

User3). Because most of the ratings User1 and User2 gave are similar (they both like Chopin and Bach, for example), the system will recognize User2 as having similar musical tastes and will recommend the music he or she likes to User1 as probably interesting. By contrast, it will recognize User3 as having different musical tastes from User1 and will not recommend the music she or he likes.

The collaborative approach is usually used for nontext data (movies or music, for example), but there are also systems that use it on text data (such as for news filtering). Table 2 lists some collaborative systems. Firefly and Ringo are two interface agents that learn from the user as well as from other agents. Such agents can serve for electronic mail handling, meeting scheduling, electronic news filtering, and entertainment recommendation. Some use the content-based approach and adopt information-retrieval methods (for example, news filtering). Others rely on the correlation between different users performing collaborative filtering (such as entertainment recommendations). For instance, the Ringo music-recommendation system recommends music that was highly scored by users with similar music tastes. Ringo tries to overcome the problem of sparse ratings coverage by building models of virtual users interested in a very narrow range of music. A developed system for music, movie, and book recommendations, Firefly requires users to start by rating several predefined items, to ensure the possibility of comparing any two users (two users can be compared only if they rated the same items).[6]

Siteseer, a Web-page recommendation system, uses an individual's bookmarks and the organization of bookmarks within folders for predicting and recommending relevant pages. The system measures the degree of overlap (such as common URLs) between the bookmark files of different users and then groups users according to that similarity. When making recommendation, Siteseer gives priority to URLs obtained from similar folders and URLs that appear in bookmark files of similar users.

Phoaks, a proposed system that automatically recognizes and redistributes recommendations of the Web resources mined from Usenet news messages, assumes that the roles of the provider and the recommendation recipient are specialized and different. It reuses recommendations from the existing online conversations. The system pays special attention to the problem of distinguishing recommended Web pages (URLs) from advertised or announced pages. It includes categorization rules that implement a strategy to distinguish different purposes for which the Web resources are mentioned.

GroupLense, a proposed collaborative filtering system for Usenet news, has a two-part database to store ratings that the users have given to messages and correlations between pairs of users based on their ratings. Because each user reads a small percentage of the total number of news messages, finding other users with whom to correlate is difficult, so an enormous number of ratings are needed to cover all the messages. This problem of ratings scarcity is common for collaborative approaches, and different systems address it in different ways. GroupLense partitions the set of news messages into clusters that are commonly read together, improving the local density of ratings.

Referral Web, an interactive system for reconstructing, visualizing, and searching the social networks on the Web, has a motivation similar to ContactFinder's, which is to help search for an expert on a given topic. This system models a social network by a graph, where nodes represent individuals and edges between nodes indicate that a direct relationship between the individuals has been detected. Referral Web constructs the network model incrementally with new users, searching for the co-occurrence of names in close proximity in any documents publicly available on the Web.

The Fab system for Web document recommendation combines content-based and collaborative approaches (Table 3 lists some systems that use both content-based and collaborative agents). It uses the content-based approach to generate a profile that represents a single user's interests and uses the collaborative approach to find similar users. The user's ratings are used to update that person's

personal profile. The two approaches are combined, and the pages matching the user's profile as well as the pages highly rated by similar users are recommended. Fab measures the similarity between users by the similarity of their profiles.

WebCobra uses a similar idea in combining the content-based and the collaborative approaches to text classification. It generates a user profile from relevant documents using a part-of-speech tagger. It groups users into collaborative clusters based on their profile similarity. Lifestyle Finder, a system for user-profile generation based on the usage of demographic data, also combines the content-based and the collaborative approaches. It uses a questionnaire to get a user's characteristics, and it groups similar users in terms of demographic data. Lifestyle Finder suggests a set of the 15 most highly scored Web documents to each user. It uses user evaluations of suggested documents to evaluate its performance.

Related systems. Robert Holte and Chris Drummond[7,8] designed a system that assists browsing of software libraries, taking keywords from the user and using a rule-based system with a forward-chaining inference. The system assumes that the library consists of one type of items and the user's goal is to find a single item. Oren Etizioni and Daniel Weld developed an integrated interface to the Internet combining a Unix shell and the World Wide Web to interact with Internet resources.[9] Their agent accepts high-level user goals and dynamically synthesizes the appropriate sequence of Internet commands to satisfy those goals. Matjaz Gams and Marko Grobelnik developed an employment agent available through the Internet that lets users browse data and order e-mails when interesting information appears.[10] Flippo Menczer developed adaptive intelligent methods to automate online information search and discovery in the Web (such as Web robots or crawlers) based on a population of intelligent agents.[11] He used genetic algorithms on the population of agents, rewarding an agent with energy for each relevant document found on the Web and charging them energy for the usage of network resources incurred by transferring documents.

See the "Machine learning on text data" sidebar for a discussion of the use of machine-learning techniques on text databases.

Text-learning for user-customized Web browsing

The Web is a rapidly growing information source, currently attracting many users with different interests. Because the interaction with the Web is through a computer, we can use computers to observe and record user actions, to communicate with users, and to use all the collected information to help users

Machine learning on text data

Most intelligent agents that use machine-learning techniques actually use the content-based approach and learn from the content of text documents. Text learning is the application of machine-learning techniques to text databases. There is considerable work underway involving learning on text documents that is not necessarily related to the Web. Here I take a look at some of this work through the prism of three questions I find important for using machine learning for text classification: what representation is used for documents, how is the high number of features dealt with, and which learning algorithm is used?

Table A summarizes these questions over some related publications to give an idea about the state of the art in using unsupervised learning on text data. This table includes systems given in Tables 1 to 3 in the main text with a more detailed analysis, if there is enough information available describing the insides of the system. I also include descriptions of some research work that does not necessarily include the development of a working intelligent agent and was therefore not included in Tables 1 to 3.

Representation. The so-called *vector representation* is the most frequently used document representation in information retrieval and text learning. It is a *bag-of-words representation*, meaning that all words from the document are taken and no ordering of words or any structure of the text is used. In a set of documents, each document is represented as a bag of words, including all the words that occur in the set of documents (see Figure A). Additional information in text documents could be used—for example, sentence structure, word position, or neighboring words. The question is how much can we gain in considering additional information in learning (and what information to consider), and what is the price we have to pay for it?

I am not aware of any current well-studied comparison or directions for text-document representation. Some information-retrieval research suggests that for long documents, considering information additional to the bag of words is not worth the effort. Work on document classification that extends the bag-of-words representation by using word sequences (*n*-grams) instead of single words suggests that using single words and word pairs as features in the bag-of-words representation improves performance of classifiers generated from short documents.[1,2]

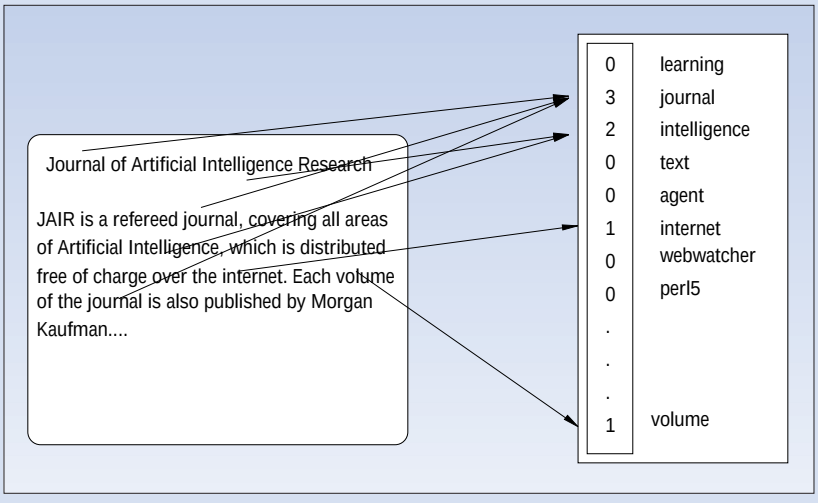

Figure A. Illustration of the bag-of-words document representation using frequency vector. Each word occurring in a set of documents is mapped into a feature. A document is represented as a vector of features where each feature is assigned a frequency (the number of times it occurs in the document).

(continued)

Table A. Document representation, feature selection, and learning algorithms used in some text-learning approaches. (The bag-of-words representation is used on Boolean features unless notified that word frequency is used—frq.)

Authors	Document representation	Feature selection	Classification
C. Apte, F. Damerau, and S.M. Weiss, "Toward Language Independent Automated Learning of Text Categorization Models," *Proc. Seventh Ann. Int'l ACM-SIGIR Conf. Research and Development in Information Retrieval*, ACM Press, New York, 1994, pp. 23–30.	Bag of words (frq)	Stop list + frequency weight	Decision rules
C. Apte, F. Damerau, and S.M. Weiss, "Text Mining with Decision Rules and Decision Trees," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Bag of words (frq)	Stemming + min.frq	Boosted decision trees
R. Armstrong et al., "WebWatcher: A Learning Apprentice for the World Wide Web," *AAAI 1995 Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, Menlo Park, Calif., 1995.	Bag of words	Informativity	TFIDF, Winnow, WordStat
M. Balabanovic and Y. Shoham, "Learning Information Retrieval Agents: Experiments with Automated Web Browsing," *AAAI 1995 Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, Menlo Park, Calif., 1995	Bag of words (frq)	Stop-list+stemming	TFIDF
B.T. Bartell, G.W. Cottrell, and R.K. Belew, "Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling," *Proc. ACM SIG Information Retrieval*, ACM Press, New York, 1992, pp. 161–167.	Bag of words (frq)	LSI (latent semantic indexing using SVD)	—
M.W. Berry, S.T. Dumais, and G.W. O'Brein, "Using Linear Algebra for Intelligent Information Retrieval," *SIAM Review*, Vol. 37, No. 4, Dec. 1995, pp. 573–595.	Bag of words (frq)	LSI	TFIDF
P.W. Foltz and S.T. Dumais, "Personalized Information Delivery: An Analysis of Information-Filtering Methods," *Comm. ACM*, Vol. 35, No. 12, 1992, pp. 51–60.			
R.M. Creecy et al., "Trading MIPS and Memory for Knowledge Eng.," *Comm. ACM,* Vol. 35, No. 8, Aug. 1992, pp. 48–64.	Bag of words	—	Memory-based reasoning
W.W. Cohen, "Learning to Classify English Text with ILP Methods," *Workshop on Inductive Logic Programming*, CS Dept., K.U. Leuven, 1995, pp. 3–24.	Bag of words + word position	Minimum frq	Decision rules, ILP
W.W. Cohen and Y. Singer, "Context-Sensitive Learning Methods for Text Categorization," *Proc. 19th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '96)*, ACM Press, New York, 1996, pp. 307–315.	Ordered word list	—	Decision rules, sleeping expert
B. Gelfand, M. Wulfekuhler, and W.F. Punch III, "Automated Concept Extraction from Plain Text," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Bag of words + WordNet	Minimum connectivity	Semantic relationship graph
T.A. Joachims, "Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," *Proc. 14th Int'l Conf. Machine Learning (ICML97)*, Morgan Kaufmann, San Francisco, 1997, pp. 143–151.	Bag of words (frq)	Minimum frq + informativity	TFIDF, PrTFIDF, naïve Bayes
T.A. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. 10th European Conf. Machine Learning (ECML 98)*, Springer-Verlag, Berlin, 1998, pp. 137–142.	Bag of words (frq)	Minimum frq	Support Vector Machines
W. Lam, K.F. Low, and C.Y. Ho, "Using Bayesian Network Induction Approach for Text Categorization," *15th Int'l Joint Conf. Artificial Intelligence (IJCAI97)*, AAAI Press, Menlo Park, Calif., 1997, pp. 745–750.	Bag of words (frq)	Mutual info	Bayesian network
W. Lam and C.Y. Ho, "Using A Generalized Instance Set for Automatic Text Categorization," *Proc. 21th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '98)*, ACM Press, New York, 1998, pp. 81–89.	Bag of words (frq)	Stop list	Generalized instance, set, *k*-nearest neighbor
D.D. Lewis and M. Ringuette, "Comparison of Two Learning Algorithms for Text Categorization," *Proc. Third Ann. Symp. Document Analysis and Information Retrieval*, Information Sciences Research Inst., Las Vegas, 1994, pp. 81–93.	Bag of words	Stop list + informativity	Naive Bayes, decision trees
D.D. Lewis and W.A. Gale, "A Sequential Algorithm for Training Text Classifiers," *Proc. Seventh Ann. Int'l ACM-SIGIR Conf. Research and Development in Information Retrieval*, ACM Press, New York, 1994.	Bag of words	Log likelihood ratio	Logistic regression with naive Bayes
D.D. Lewis et al., "Training Algorithms for Linear Text Classifiers," *Proc. 19th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '96)*, ACM Press, New York, 1996, pp. 298-306.	Bag of words (frq)	—	Widrow-Hoff, EG
R. Liere and P. Tadepalli, "Active Learning with Committees: Preliminary Results in Comparing Winnow and Perceptron in Text Categorization," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Bag of words	—	Winnow (in query by committee)

Authors	Document representation	Feature selection	Classification
P. Maes, "Agents that Reduce Work and Information Overload," *Comm. ACM,* Vol. 37, No. 7, July 1994, pp. 30–40.	Bag of words + header information	Selecting keywords	Memory-based reasoning
D. Mladenic, *Personal WebWatcher: Implementation and Design*, Tech. Report IJS-DP-7472, Carnegie Mellon Univ., Pittsburgh, 1996; http://www.cs.cmu.edu/~TextLearning/pww	Bag of words (frq)	Informativity	Naive Bayes, nearest neighbor
D. Mladenic and M. Grobelnik, "Feature Selection for Classification Based on Text Hierarchy, *Working Notes of Learning from Text and the Web, Conf. Automated Learning and (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998.	Bag of words using *n*-grams (frq)	Stop list + minimum frq + odds ratio	Naive Bayes
D. Mladenic and M. Grobelnik, "Word Sequences as Features in Text-Learning," *Proc. Seventh Electrotechnical and Computer Security Conf. (ERK '98)*, IEEE Region 8, Slovenia Section IEEE, Ljubljana, Slovenia., 1998, pp. 145–148.			
I. Moulinier and J.-G. Ganascia, "Applying an Existing Machine Learning Algorithm to Text Categorization," *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, S. Wermter, E. Riloff, and G. Scheler, eds., Springer-Verlag, Berlin, 1996, pp. 343–354.	Bag of words	Informativity	Decision rules
K. Nigam and A. McCallum, "Pool-Based Active Learning for Text Classification," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Bag of words	Minimum frq	EM with QBC
M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying Interesting Web Sites," *Proc. 13th Nat'l Conf. Artificial Intelligence AAAI 96*, AAAI Press, Menlo Park, Calif., 1996, pp. 54–61.			
M. Pazzani and D. Billsus, "Learning and Revising User Profiles: The Identification of Interesting Web Sites," *Machine Learning 27*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 313–331.	Bag of words	Stop list + informativity	TFIDF, naive Bayes, nearest neighbor, neural networks, decision trees
J. Shavlik and T. Eliassi-Rad, "Building Intelligent Agents for Web-Based Tasks: A Theory-Refinement Approach," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Localized bag of words	Stop-list + stemming	Theory refinement on neural networks
S. Slattery and M. Craven, "Learning to Exploit Document Relationships and Structure: The Case for Relational Learning on the Web," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://www.cs.cmu.edu/~conald/conald.shtml.	Bag of words + hypertext/graph	Informativity	Naive Bayes, ILP
M. Mc Elligott and H. Sorensen, "An Emergent Approach to Information Filtering," *Abakus. U.C.C. Computer Science J.*, Vol. 1, No. 4, Dec. 1993, pp. 1–19.	*n*-gram graph (only bigrams)	Weighting graph edges	Connectionist combined with
H. Sorensen and M. McElligott, "PSUN: A Profiling System for Usenet News," *CIKM'95 Intelligent Information Agents Workshop*, 1995.			genetic algorithms
E. Wiener, J.O. Pedersen, and A.S. Weigend, "A Neural Network Approach to Topic Spotting," *Proc. Fourth Ann. Symp. Document Analysis and Information Retrieval (SDAIR '95)*, Information Science Research Inst., Las Vegas, 1995; http://www.stern.nyu.edu/~aweigend/Research/Papers/TextCategorization.	Bag of words	Stop-list + minimum frq + stemming + relevancy or LSI	Neural networks, logistic regression
Y. Yang, "Expert Network: Effective and Efficient Learning form Human Decisions in Text Categorization and Retrieval," *Proc. Seventh Ann. Int'l ACM-SIGIR Conf. Research and Development in Information Retrieval*, ACM Press, New York, 1994, pp. 13–22.	Bag of words	Informativity, c^2-stat.	*k*-nearest neighbor LLSF
Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Information Retrieval J.*, May 1999.			

As Table A shows, any current systems that learn from text use the bag-of-words representation with Boolean features, which indicates if a specific word occurred in a document or the frequency of a word in a given document. Some work uses additional information such as word position or word tuples called *n*-grams (for example, "machine learning" is a 2 gram and "World Wide Web" is a 3 gram). Some recent work indicates that the use of hypertext structure and graph organization of Web pages improves classification results. There is currently no study that compares different document representations over several domains to show clear advantages of some representation.

Number of features. One of the frequently used approaches to reduce the number of different words is to remove words that occur in the "stop list" containing common English words like "a," "the," or "with," or pruning the infrequent words (word frequency < min.frequency)—again,

(continued)

navigate the Web. Text learning can be applied on collected information to help users browsing the Web. Our work on Personal WebWatcher[3] is mainly inspired by WebWatcher, a Learning Apprentice for the World Wide Web,[12,13] and other work related to learning apprentice and learning from text.[14-17] A learning apprentice lets us automatically customize to individual users, using each user interaction as a training example. Personal WebWatcher can be installed locally by the user and connected to

see Table A for examples of work being done in the various areas I describe. Connected to the particular language is also word stemming, which reduces the number of different words using a language-specific stemming algorithm—for example, "work" replaces "works," "working," and "worked").

Many approaches use a language-independent approach and introduce some sort of word scoring to select only the best words or reduce the dimensionality using latent semantic indexing (LSI) with singular value decomposition.

Experiments with different numbers of selected features used in text classification indicate that the best results come from either using only a small percentage of carefully selected features (up to 10% of all features) or, in some cases, using all the features (see the papers by Lewis and colleagues, Yang and Pedersen, and Weiner and colleagues in Table A). A comparison of different word-scoring measures used in feature-subset selection for text data shows that the most promising measures take into account the nature of the problem domain and the classification algorithm characteristics used.[3] Surprisingly good results are obtained using a simple frequency measure in a combination with a stop list.

Algorithms. One well-established technique for text classification in information retrieval is to represent each document with a bag of words as a *TFIDF*-vector in the space of words that appear in training documents, sum all interesting document vectors, and use the resulting vector as a model for classification (based on the relevance feedback method—see papers by Salton and Buckley and by Rocchio in Table A). Each component of a document vector $d^{(i)} = TF(w_i, d)IDF(w_i)$ is calculated as the product of *TF* (*term frequency*—number of times word w_i occurred in a document) and $IDF = log[D/DF(w_i)]$ (*inverse document frequency*), where D is the number of documents and document frequency $DF(w_i)$ is the number of documents where words w_i occurred at least once. The exact formulas used in different approaches might slightly vary (some factors are added and normalization is performed, but the idea remains the same). The approach then represents a new document as a vector in the same vector space as the generated model and measures the distance between them (usually using the cosine similarity measure) to classify the document. This technique is commonly used to get baseline results when testing a machine-learning algorithm on text data (see Mitchell in Table A). TFIDF classification has already been used in machine-learning experiments on Web data and, in most cases, it proved inferior to tested machine-learning methods.

An extension of TFIDF called probabilistic TFIDF takes into account document representation and outperforms TFIDF, while proving comparable to the naive Bayesian classifier (Joachims in Table A). The Naive Bayesian classifier and the *k*-nearest neighbor are two classifiers commonly used in text learning and reported to be among the best performing classifiers for text data. In addition to using the naive Bayesian classifier and nearest neighbor, several experimenters performed experiments on text data with symbolic learning using decision trees, and one group experimented using decision rules. Another group compared the performance of linear least-square fit (LLSF) and a variant of *k*-nearest neighbor, reporting that both classifiers achieved similar results.

As Table A shows, Creecy et al. and Maes used memory-based reasoning. Apte and colleagues used decision rules and later boosted decision trees. Cohen used decision rules, the sleeping experts algorithm, and two inductive logic programming (ILP) algorithms, FOIL and Flipper. Slattery and Craven used the Naive Bayesian classifier and two ILP algorithms FOIL and FOIL-PILFS (FOIL with Predicate Invention for Large Feature Spaces). Lewis et al. used a combination of the Naive Bayesian classifier and logistic regression, the Widrow-Hoff algorithm, and exponential gradient (EG). Using neural networks, Wiener et al. achieved slightly better results than logistic regression. McElligot and Sorensen used a connectionist approach combined with genetic algorithms, and Lam et al. used Bayesian network induction.

Gelfand used semantic relationship graphs (SRG) to represent documents based on the WordNet lexical database. This approach performs classification similar to TFIDF, defining each class by a group of training documents and representing it as a union of their SRG representation. Armstrong et al. used TFIDF, the Winnow algorithm, and a statistical approach called WordStat that assumes mutual independence of words. Shavlik and Eliassi used theory refinement on neural networks, where the user provides an initial advice that is compiled into neural networks and refined during the interaction with the user based on the user's page ratings and additionally provided advisories. Liere and Tadepalli used active learning involving a committee of Winnow learners. Nigam and McCallum also used active learning in a combination of *query by committee* and the expectation-maximization algorithm.

There is currently no strong evidence pointing to the superiority of any of these text-learning algorithms over different domains. Most experiments show the superiority of the tested algorithm over the TFIDF classification. In comparing learning algorithms, Pazzani and Billsus indicate that a document representation including feature selection is a more promising approach to classification-accuracy improvement than finding a better learning algorithm. In their experiments, the naive Bayesian classifier, nearest neighbor, and neural networks performed best on tested data. Yang found similar good performance for *k*-nearest neighbor, neural networks, and linear least-square fit, while also showing poor performance for the naive Bayesian classifier. However, Yang and Joachims reported that on their domains feature, subset selection was not crucial for the classifier performance. Joachims reported that Support Vector Machines outperform the naive Bayesian classifier, while Apte, Damerau, and Weiss obtained even better results using boosted decision trees. Lam and Ho observed that the generalized-instance-set algorithm achieved better results than either *k*-nearest neighbor or linear classifiers (Rocchio, Widrow-Hoff).

References

1. D. Mladenic and M. Grobelnik, "Feature Selection for Classification Based on Text Hierarchy," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://wwwcs.cmu.edu/~TextLearning/pww.

2. D. Mladenic and M. Grobelnik, "Word Sequences as Features in Text-Learning," *Proc. Seventh Electrotechnical and Computer Science Conf. (ERK'98)*, IEEE Region 8, Slovenia Section IEEE, Ljubljana, Slovenia, 1998, pp. 145–148.

3. D. Mladenic, *Machine Learning on Nonhomogeneous, Distributed Text Data*, PhD thesis, Faculty for Computer and Information Science, Univ. of Ljubljana, Slovenia, 1998; http://www-ai.ijs.si/DunjaMladenic/PhD.html.

the Web browser as a proxy server. A prototype version of the system is available and used for research purposes (http://cs.cmu.edu/~TextLearning/pww).

Personal WebWatcher is a content-based personal agent that helps users browse the Web. Imagine a user using a Web browser for requesting documents, most often by clicking the hyperlinks on already requested documents that are presented to that user. Predicting and highlighting the clicked hyperlinks is one way to help users navigate the Web. In the machine-learning setting we used, we considered all the hyperlinks shown to the user as training examples with a Boolean class value (clicked or unclicked). We build descriptions of hyperlinks and treat these as shortened documents for machine-learning training examples. Our approach collects training examples online from each interaction with the user. Data collected from a single user represents a domain to be handled by machine-learning techniques. The system's first version uses a bag-of-word document representation, feature selection using informativity, and a naive Bayesian classifier.[3]

Personal WebWatcher's structure. The idea is to help users browse the Web without putting any additional workload on them. The only work involved is to install the system and simply connect it to the Web browser as a proxy server. Each request goes from the user to our system, which retrieves the requested Web page's original document from the Web. The original is stored on disk for learning and processed for adding the suggestions. Personal WebWatcher sends the modified Web page to the user instead of to the original one. The modification we introduce does not remove any information from the Web page—it adds small icons highlighting potentially interesting hyperlinks on the page.

As Figure 4 shows, Personal WebWatcher consists of two main parts: a *proxy server* that interacts with the user through the Web browser and a *learner* that provides the user model to the server. The communication between them is through a disk; the proxy saves addresses of visited documents (URLs), and the learner uses them to retrieve documents used to generate a model of user interests.

Proxy waits in an infinite loop for a Web page request from the browser. On request, it fetches the requested document and, if it is an HMTL document, adds advice and forwards the document to the user. To add suggestions, proxy forwards the document to the adviser, which extracts descriptions of hyperlinks from the document and calls classification that uses the generated user model. Each hyperlink is classified—assigned a degree of relevance based on its similarity to the user model—and the most relevant hyperlinks are highlighted.

Figure 5 gives an example page presented

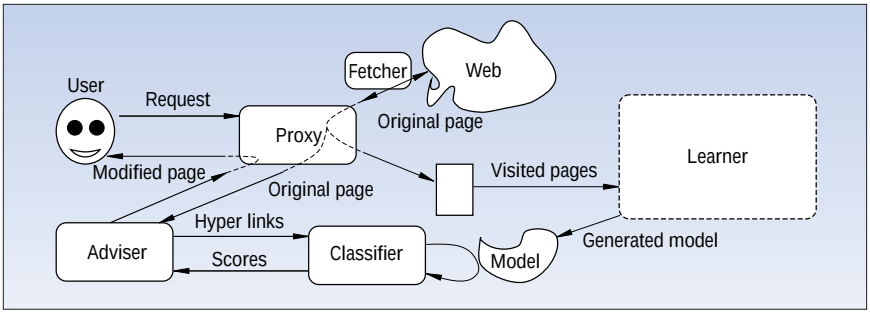

Figure 4. Structure of Personal WebWatcher, an assistant for Web browsing.

Figure 5. Example of HTML-page presented to the user by Personal WebWatcher. Notice that three hyperlinks are highlighted as interesting ("Machine Learning Information Services" and two project members: Dayne Freitag, Thorsten Joachims).

via the Netscape browser by Personal WebWatcher. The document is actually the WebWatcher project page. Once run, the system processes the requested pages by adding a banner to the top of the page showing that Personal WebWatcher is watching over the user's shoulder and highlighting interesting hyperlinks. A limited number of hyperlinks that are scored above a given threshold are recommended to the user, indicating their scores using graphical symbols placed around highly scored hyperlinks. For example, in Figure 5, three hyperlinks are suggested by Personal WebWatcher—"Machine Learning Information Services" and two project members (Dayne Freitag, Thorsten Joachims)—based on the model of interests built from about 500 documents I visited in 1996.

OUR PERSONAL WEBWATCHER project is involved in ongoing research on different aspects of using text learning for better Web browsing, including richer document representations using dynamically constructed background knowledge and making word-occurrence prediction based on very short documents.[18] An important direction of text-learning is using information extraction specialized for different domains, such as finding publication citations in the research papers available on the Web (CiteSeer[19]) or building a knowledge database from the Web (WebKB[20]). The current trend of using machine learning for intelligent agents includes a combination of text processing with speech recognition and content processing of image or video.[21]

On a broader scale, interesting research questions for future work include the study of scalability to domains having very short or very long documents. My colleagues and I have described experiments on very short documents (containing hyperlink descriptions).[18] Other interesting research questions include the influence of sparse-word statistics, the incorporation of temporal information in Web-browsing behavior, a combination of natural language and statistical methods, learning from structure in hypertext, the development of statistical models that represent hypertext structure, and combining evidence from multiple sources.[22] ■

Acknowledgments

This work was financially supported by the Slovenian Ministry for Science and Technology. Part of this work was performed during my stay at Carnegie Mellon University in Tom Mitchell's group. Many thanks to the anonymous reviewers for valuable comments and suggestions. I am grateful to Nada Lavrac and the magazine's editors for their valuable comments and suggestions on the latest version of this article.

References

1. T.M. Mitchell, *Machine Learning*, McGraw Hill, New York, 1997.
2. U. Fayyad et al., *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, Cambridge, Mass., 1996.
3. D. Mladenic, *Personal WebWatcher: Implementation and Design*, Tech. Report IJS-DP-7472, Carnegie Mellon Univ., Pittsburgh, 1996; http://cs.cmu.edu/TextLearning/pww.
4. M. Balabanovic and Y. Shoham, "Fab: Content-Based, Collaborative Recommendation," *Comm. ACM*, Vol. 40, No. 3, Mar. 1997, pp. 66–70.
5. P. Maes, "Agents That Reduce Work and Information Overload," *Comm. ACM*, Vol. 37, No. 7, July 1994, pp. 30–40.
6. S. Hedberg, "Agents for Sale: First Wave of Intelligent Agents Go Commercial, *IEEE Expert*, Vol. 11, No. 6, Dec. 1996, pp. 16–19.
7. R.C. Holte and C. Drummond, "A Learning Apprentice for Browsing," *AAAI Spring Symp. Software Agents*, AAAI Press, Menlo Park, Calif., 1994.
8. C. Drummond, D. Ionescu, and R. Holte, *A Learning Agent That Assists the Browsing of Software Libraries*, Tech. Report TR-95-12, Computer Science Dept., Univ. of Ottawa, Ottawa, Canada, 1995.
9. O. Etizioni and D. Weld, "A Softbot-Based Interface to the Internet," *Comm. ACM*, Vol. 37, No. 7, July 1994, pp. 72–79.
10. M. Gams and M. Grobelnik, "Intelligent Agents in Information Society," *Proc. Sixth Electrotechnical and Computer Science Conf. (ERK 97)*, IEEE Press, Piscataway, N.J., 1997, pp. 125–128.
11. F. Menczer, "Arachnid: Adaptive Retrieval Agents Choosing Heuristic Neighborhood for Information Discovery," *Proc. 14th Int'l Conf. Machine Learning*, 1997, pp. 227–235.
12. R. Armstrong et al., "WebWatcher: A Learning Apprentice for the World Wide Web," *AAAI 1995 Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*, AAAI Press, Menlo Park, Calif., 1995.
13. T. Joachims et al., "Machine Learning and Hypertext," *Fachgruppentreffen Maschinelles Lernen*, Dortmund, Aug. 1995.
14. T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization," *Proc. 14th Int'l Conf. Machine Learning (ICML 97)*, 1997, pp. 143–151.
15. H.C.M. de Kroon, T. Mitchell, and E.J.H. Kerckhoffs, "Improving Learning Accuracy in Information Filtering," *ICML-96 Workshop: Machine Learning Meets Human-Computer Interaction*, 1996.
16. K. Lang, "News Weeder: Learning to Filter Netnews, *Proc. 12th Int'l Conf. Machine Learning (ICML 95)*, Morgan Kaufmann, San Francisco, 1995, pp. 313–339.
17. T. Mitchell, "Experience with a Learning Personal Assistant," *Comm. ACM*, Vol. 37, No. 7, July 1994, pp. 81–91.
18. D. Mladenic, *Machine Learning on Non-homogeneous, Distributed Text Data*, PhD thesis, Faculty for Computer and Information Science, Univ. of Ljubljana, Slovenia, 1998; http://cs.cmu.edu/~TextLearning/pww/PhD.html.
19. K. Bollacker, S. Lawrence, and L. Giles, "CiteSeer: An Autonomous System for Processing and Organizing Scientific Literature on the Web," *Working Notes of Learning from Text and the Web, Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://cs.cmu.edu/~conald/conald.shtml.
20. T. Mitchell et al., "The World Wide Knowledge Base Project," 1998; http://cs.cmu.edu/~WebKB.
21. "Working Notes of Workshop on Mixed Media Databases," *Proc. Conf. Automated Learning and Discovery (CONALD-98)*, Carnegie Mellon Univ., Pittsburgh, 1998; http://cs.cmu.edu/~conald/conald.shtml.
22. J. Carbonell et al., *Report on the CONALD Workshop on Learning from Text and the Web*, Carnegie Mellon Univ., Pittsburgh, 1998.

Dunja Mladenic is a researcher in computer science at the J. Stefan Institute and a teaching assistant at Ljubljana University. Most of her research work is connected with the study and development of machine-learning techniques and their applications on real-world problems from different areas such as medicine, pharmacology, manufacturing, and economics. She is currently working on using machine learning in data analysis, with particular interest in learning from text and the Web. She received her PhD (http://www-ai.ijs.si/DunjaMladenic/PhD.html) from the Faculty for Computer and Information Science, University of Ljubljana. Contact her at the Dept. of Intelligent Systems, J. Stefan Inst., Jamova 39, 1000 Ljubljana, Slovenia; dunja.mladenic@ijs.si; dunja@cs.cmu.edu.