Text-learning and intelligent agents *

Dunja Mladenić
Department for Intelligent Systems, J.Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
Phone: (+386)(61) 1773 272, Fax: (+38)(61) 1251-038
E-mail: Dunja.Mladenic@ijs.si
http://www-ai.ijs.si/DunjaMladenic/

Abstract

We present an overview of some work in text-learning through the prism of the three research questions important for development of text-learning intelligent agents: what representation is used for documents, how is the high number of features dealt with and which learning algorithm is used. These questions are addressed here in an overview of the existing approaches to text-learning. Two frequently used approaches to development of intelligent agents using machine learning techniques are described here: content-based and collaborative approach. The usage of machine learning on text databases (usually referred to as text-learning) is an important part of the content-based approach applied on text documents. Example are agents for locating information on World Wide Web and Usenet news filtering agents. For illustration we give a brief description of the content-based personal assistant named Personal WebWatcher that uses text-learning for user customized Web browsing.

1 Introduction

With the growing availability of information sources, especially non-homogeneous, distributed sources like the World Wide Web, there is also a growing interest in tools that can help in making a good and quick selection of information we are interested in. Recent work that arises at the intersection of Information Retrieval and Machine Learning offers some novel solutions to this problem, as well as work in Intelligent Agents and Intelligent User Interfaces. Connected to this is intensive development of methods for using machine learning techniques on text

*Technical Report IJS-DP-7948, Department of Intelligent Systems, J.Stefan Institute, 1998.

databases referred to as *text-learning*, that combine research in machine learning with information retrieval.

The rest of this paper is organized as follows. Section 2 gives description of some work related to the usage of machine learning techniques in development of intelligent agents. Section 3 gives an analysis of different approaches to text-learning. Description of our personal intelligent agent named Personal WebWatcher that uses text-learning for user customized Web browsing is given in Section 4 together with the description of its inside structure. Conclusion of the paper outlines directions of some on-going research in text-learning for Web browsing.

2 Machine learning for intelligent agents

There are different types of systems that are called intelligent agents and different definitions of the term *intelligent agent* that we are not going to discuss here. We will rather present examples of some systems called intelligent agents (user assistant, recommendation systems) that involve usage of machine learning or data mining techniques.

Two approaches to development of intelligent agents based on machine learning techniques are frequently used [5]: *content-based* and *collaborative*. The next two Sections 2.1, 2.2 describe these two approaches and give examples of the systems based on them. Section 2.3 gives some more examples of other sorts of intelligent agents that use machine learning techniques but can not easily be classified to either content-based or collaborative approach.

2.1 Content-based approach

In the content-based approach the system searches for the items similar to those user liked based on the content comparison. The approach has its roots in information retrieval research and when applied to the text data, gives one of the possibility for direct application of different text-learning methods.

One of the main problems with this approach is to capture different aspects of the content (eg., music, movies, images). Even for the text domains, most representations capture only certain aspects of the content leading to poor system performance. Additionally to the representational problems, content-based systems tend to specialize for items similar to already seen items.

For instance, take the problem where we want to recommend Web pages to the user, based on their content. If the user was interested in the Web page given in Figure 1, the Web page in the left hand-side of Figure 2 will be recommended by content-based approach since it has similar content as the page in Figure 1. The Web page in the right-hand side of Figure 2 would be recognized as different

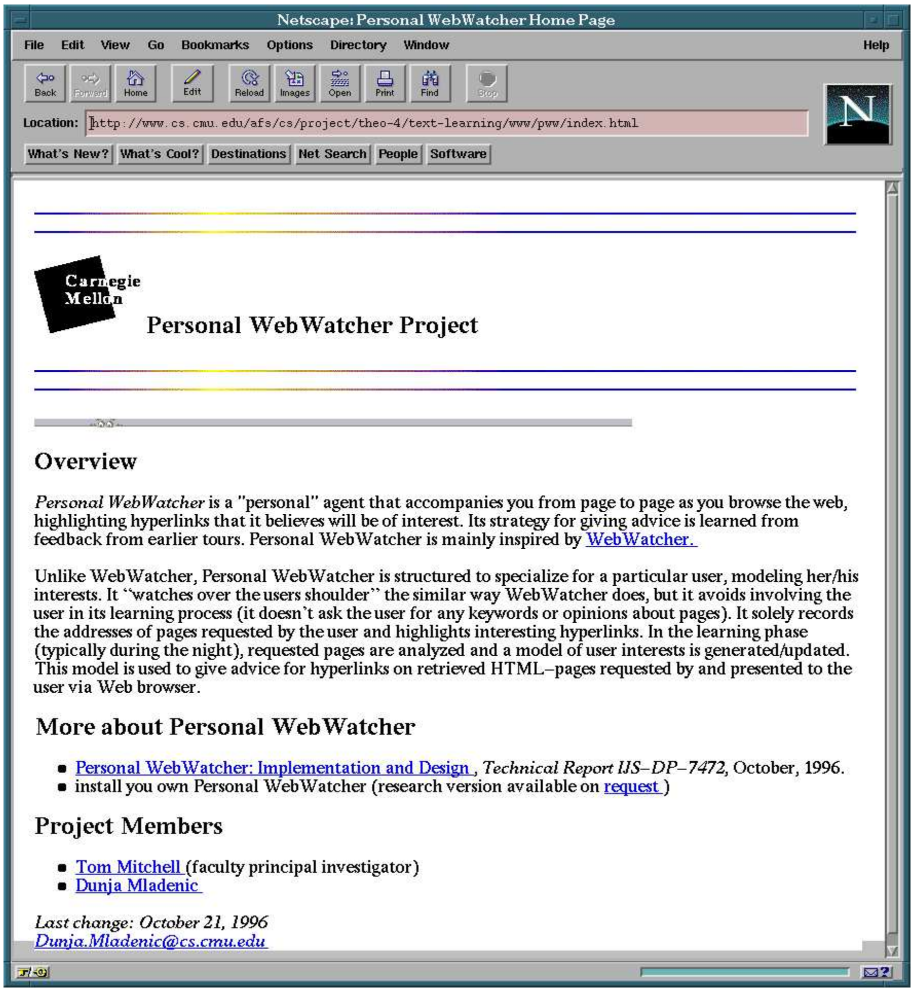

Figure 1: Example of the Web pages that an imaginary user found interesting.

and thus not recommended (unless the user also found interesting some other pages that are similar to it).

Content-based approach is popular for the systems that work on text data, for example, on World Wide Web documents or News. Armstrong et al. [3] developed `WebWatcher`, a system that assists user in locating information on the World Wide Web taking keywords from the user, suggesting hyperlinks and receiving evaluation. The system also offers a possibility to get some more similar documents. The data saved by the system contains information about the search keywords different users typed in, hyperlinks that were followed and the evaluation given by the users at the end of each search. Balabanović and Shoham [4] developed "a system which learns to browse the Internet on behalf of a user". It searches the World Wide Web taking bounded amount of time, selects the best pages and receives an evaluation from the user. The evaluation is used to update the search

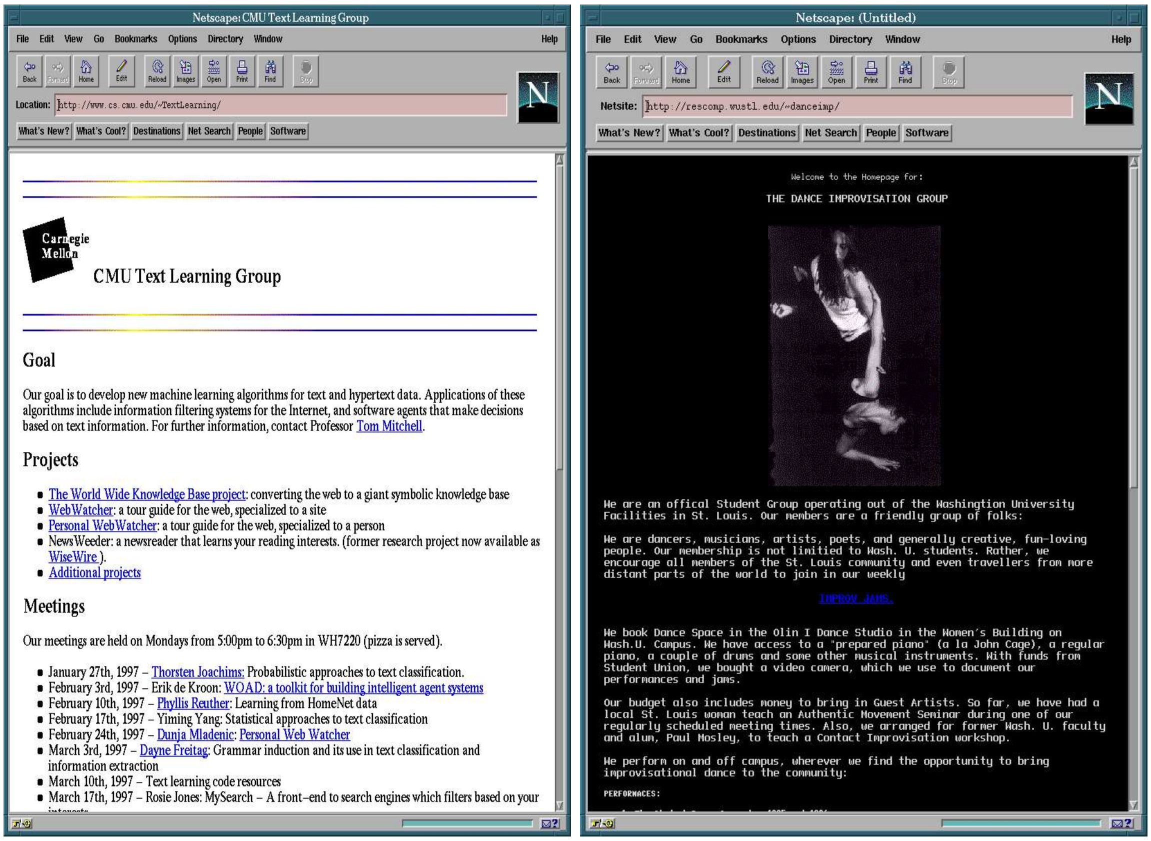

Figure 2: Example of two Web pages having different content and thus recognized as not similar by the content-based approach. The left-hand side page would be recognized as similar to the page in the previous Figure and recommended by content-based approach, while the right-hand side page would be recognized as different from the both previous Web pages.

and selection heuristics. Goldman et al. [16] developed `Musag`, a system that 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. Generated thesaurus is used in document retrieval to extend a set of given keywords. Lieberman [33] developed `Letizia`, "a user interface agent that assists a user browsing the World Wide Web". The system doesn't require any keywords or rating from the user, it infers the user interests from the browsing behavior. The users of the Web browsers usually perform depth-first search. While the user is reading a Web document `Letizia` performs a breath-first search from the current document. Potentially interesting hyperlinks found during the search are suggested to the user in a separate NETSCAPE window. Pazzani et al. [40] and Ackerman et al. [1, 41] developed `Syskill & Webert`, a system that collects ratings of the explored Web pages from the user and learns a user profile from them. Pages are separated according to their topic and a separate profile is learned for each topic. Generated user profile is used to form queries for the existing search engines in order to get more potentially interesting documents. Lang [31] developed `NewsWeeder`, a system for electronic news filtering that uses text-learning to generate model of user interests. The system uses World Wide Web interface to enable user to access the news in a usual way and to enable the system to collect user's ratings as feedback. NewsWeeder additionally assigns predicted rating to each article and generates a personalized list of the top articles (eg., 50 articles predicted as the most interesting) found among all articles.

Krulwich and Burkey [27] proposed "The `ContactFinder` agent" that reads and responds to bulletin board messages; assists users by referring them to the other people who can help them and; categorizes messages and extracts their topic areas. The system operates in two phases. In the first phase, it searches bulletin board looking for the contact persons and their topic areas and storing found information in the database. In the second phase, the system searches new messages looking for questions, extracting the topic area of the found question and finding in the database the contact person for that topic area. To extract the topic area, ContacFinder uses some heuristics. For instance, extracts semantically significant phrases (eg., fully capitalized words, short phrases: 1 to 5 words, words in a different format from surrounding text). Hammond et al. [17] and Burke et al. [9, 10] developed `FAQFinder`, a system that uses a "natural language question-based interface to access distributed text information sources" and helps the user to find answers to her/his question in databases such as FAQ files. Questions from relevant FAQ files are matched against user's question and five best matching questions are returned together with their answers. Kamba et al. [22] developed `Antagonomy`, a system that composes personalized newspaper on the Web. The system 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 of article matching the user profile. For instance, more articles with the higher score are placed on the top of the newspaper. LaMacchia [30] proposed `Internet Fish`, 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 system also includes help in browsing the Web by the usage of some existing search engines and user's rating of the documents. Mitchell et al. [36] proposed `Calendar Apprentice`, a system that helps a user in meeting scheduling. The system is connected to the user's electronic calendar and generates sets of rules capturing the user's scheduling preferences and some other information about individual attendees of the meetings. It uses these rules to provide advice to the user for new, unscheduled meetings.

2.2 Collaborative approach

For the difference from content-based approach that can be successfully applied to a single user, collaborative approach assumes that there is a set of users using the system. In the collaborative approach (sometimes referred to as social-learning [34]) advise to the user is based on the reaction of other users. The system searches for users with similar interest and recommends the items those users liked. Instead of computing similarity between items the system computes similarity between the users. The assumption is that users provide ratings for the seen items. In collaborative approach there is no analysis of the item content, meaning that items of any content can be handled with equal success. Each item is assigned a unique identifier and a rating given by the user. The similarity between users is based on the comparison of the ratings they assigned to the same items.

One of the main problems with the collaborative approach is that the small number of users relative to the number of items brings a danger of a sparse coverage of ratings. For any new item in the database, we need to collect information from different users to be able to recommend it, and similar users are not matched unless they have rated the sufficient number of the same items. The problem is also if some user has unusual taste compared to the rest of the system users, since there will be no other similar user and the system performance will be poor.

For instance, consider making recommendation for `User1`, whose ratings for music are given in Figure 3. The recommendation is based on the ratings of other users (`User2` and `User3` in Figure 3). Since most of the ratings given by `User1` and `User2` are similar (they both like Chopin, Bach, ...), `User2` will be recognized as having similar music taste and the music that she/he liked will be recommended to `User1` as probably interesting. On the contrary, `User3` would be recognized as having different music taste from User1 and the music she/he liked will not be recommended.

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. Regarding the music taste, collaborative approach would find `User2` similar to `User1`, while `User3` would be found different from `User1`.

Collaborative approach is usually used for non-text data (eg., movie, music), but there are also systems that use it on text data (eg., for news filtering). Maes [34] described ideas for "interface agents" that learns from the user as well as from other agents. As examples of such agents they developed agents for: electronic mail handling, meeting scheduling, electronic news filtering and entertainment recommendation. Some of these agents use the content-based approach and adopt Information Retrieval approaches (eg., news filtering). Others relay on correlation between different users performing collaborative filtering (eg., entertainment recommendation). They developed a music recommendation system `Ringo`, that recommends music that was highly scored by users with similar music tastes. The system trays to overcome the problem of sparse coverage of ratings with building models of virtual users interested in a very narrow range of music. The same group developed system `Firefly` for music, movie and books recommendation [18]. `Firefly` requires from the user to begin with rating several predefined items, to ensure the possibility of comparing any two user (recall that two users can be compared only if they rated the same items). Terveen et al. [46] propose `PHOAKS`, a system that automatically recognizes and redistributes recommendations of the Web resources mined from the Usenet news messages. The systems assumes the roles of provider and recommendation recipient are specialized and different. It reuses recommendations from the existing online conversations. The special attention is given to the problem of distinguishing recommended Web pages (URLs) from advertised or announced pages. The system includes categorization rules that implement a strategy to distinguish different purposes for which the Web resources are mentioned.

Rucker and Marcos [43] developed `Siteseer`, a "Web-page recommendation system that 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, priority is given to URLs from more similar folders and URLs that appear in bookmark files of similar users. Konstan et al. [25] developed `GroupLense`, "a collaborative filtering system for Usenet news". The system has a two-part database to store (1) ratings that the users have given to messages and (2) correlations between pairs of users based on their ratings. Since each user reads a small percentage of the total number of news messages, it is difficult to find other users with whom to correlate and an enormous number of ratings is needed to cover all of the messages. This problem of ratings sparsity is common for the 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. Kautz et al. [23, 24] developed `Referral Web`, "an interactive system for reconstructing, visualizing, and searching the social networks on the World Wide Web". The motivation is similar as for already mentioned content-based system `Contact Finder`, to help in searching for an expert on a given topic. A social network is modeled by a graph, where nodes represent individuals, and edges between nodes indicate that a direct relationship between the individuals has been detected. The network model is constructed incrementally with new users, searching for the co-occurrence of names in close proximity in any documents publicly available on the Web.

Balabanović and Shoham [5] developed a system for Web documents recommendation that combines content-based and collaborative approach. Content-based approach is used to generate profile that represent a single user's interests and collaborative approach is used to find similar users. The user's ratings are used to update their personal profile. The two approaches are combined and the pages matching user's profile as well as the pages highly rated by similar users are recommended. Similarity between users is measured by the similarity of their profiles. Krulwich [28] developed `Lifestyle finder`, a system for user profiles generation based on the usage of demographic data. The system combines content-based and collaborative approach. Questionare is used to get users characteristics and group similar users in terms of demographic data. A set of the 15 highly scored Web documents is suggested to each user. Users evaluation of suggested documents is used to evaluate the system performance.

2.3 Some other related systems

Holte and Drummond [19], Drummond et al. [12] designed a system that assists browsing of software libraries, taking keywords from the user and using a rule-based system with forward chaining inference. The system assumes that the library consists of one type of items and the user goal is finding a single item. Etizioni and Weld [14] offer an integrated interface to the Internet combining UNIX shell and the World Wide Web to interact with Internet resources. Their agent accepts high-level user goals and dynamically synthesizes the appropriate sequence of Internet commands to satisfy those goals. Menczer [35] developed "adaptive intelligent methods to automate on-line information search and discovery in the Web" (like Web robots or crawlers) based on population of intelligent agents. He used genetic algorithms on the population of agents, rewarding 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.

There are many other intelligent agents not mentioned here but our hope is to give a general idea about the current trends. Table 1 contains a summary of the described systems. For each system we give reference to the paper describing the system, name, organization that developed it and a few word summary of its functionality.

Paper reference	Agent name	Development org.	Goal is to help user
Content-based agents			
Armstrong et al. [3]	WebWatcher	CMU	browsing WWW
Balabanović and Shoham [4]	Lira	Stanford	browsing WWW
Goldman et al. [16]	Musag	Hebrew Univ.	browsing WWW
Lieberman [33]	Letizia	MIT	browsing WWW
Pazzani et al. [40, 41]	Syskill & Webert	UCI	browsing WWW
Lang [31]	NewsWeeder	CMU	Usenet news filtering
Krulwich and Burkey [27]	ContactFinder	Andersen Consult.	finding expert
Burke et al. [9, 10]	FAQFinder	Univ. of Chicago	answer question
Kamba et al. [22]	Antagonomy	NEC corp.	personalized newspaper
LaMacchia [30]	Internet Fish	MIT	extract info. from Internet
Mitchell et al. [36]	Calendar Apprentice	CMU	meeting scheduling
Mladeni''c [38]	Personal WebWatcher	CMU, IJS	browsing WWW
Collaborative agents			
Maes [34]	Ringo Firefly	MIT MIT	finding music music, movie, book
Terveen et al. [46]	PHOAKS	AT&T labs.	browsing WWW
Rucker and Marcos [43]	Siteseer	Imana, Inc.	browsing WWW
Konstan et al. [25]	GroupLense	Univ. Minnesota	Usenet news filtering
Kautz et al. [23], [24]	Referral Web	AT&T labs.	finding expert
Combination of content-based and collaborative agent			
Balabanović and Shoham [5]	Fab	Stanford	browsing WWW
Krulwich [28]	Lifestyle Finder	AgentSoft Ltd.	browsing WWW

Table 1: Summary of some content-based and collaborative systems that use some machine lerarning related techniques. We give for each system name, development organization and main functionality.

3 Text-learning

Most of the intelligent agents that use machine learning techniques actually use content-based approach and learn from the content of text documents. Text-learning can be described as the usage of machine learning techniques on text databases. There is quite some work in the area of learning over text documents that is not necessary related to the Web. Here we take look at some of these work through the prism of three questions important for machine learning: (1) what representation is used for documents, (2) how is the high number of features dealt with and (3) which learning algorithm is used. Table 2 summarizes them over some related papers in order to give an idea about the current trends. Systems given in Table 1 are included in this more detailed analysis, if there was

Paper reference	Document Representation	Feature Selection	Learning
Apté et al. [2]	bag-of-words (freq)	stop list+ frequency weight	Decision Rules
Armstrong et al. [3]	bag-of-words	informativity	TFIDF Winnow, WordStat
Balabanović and Shoham [4]	bag-of-words (freq)	stop list+stemming+ keep 10 best words	TFIDF
Bartell et al. [6]	bag-of-words (freq)	latent semantic indexing using SVD	—
Berry et al. [7] Foltz and Dumais [15]	bag-of-words (freq)	latent semantic indexing using SVD	TFIDF
Cohen [11]	bag-of-words	infrequent words pruned	Decision Rules ILP
Joachims [21]	bag-of-words (freq)	in/frequent words+ informativity	TFIDF, PrTFIDF, Naive Bayes
Lam et al. [29]	bag-of-words (freq)	mutual info.	Bayesian Network
Lewis et al. [32]	bag-of-words	log likelihood ratio	logistic regression with Naive Bayes
Maes [34]	bag-of-words+ header info.	mail/news header info.+ selecting keywords	Memory-Based reasoning
Mladenić [38]	bag-of-words (freq)	informativity	Naive Bayes Nearest Neighbor
Pazzani et al. [40, 41]	bag-of-words	stop list+ informativity	TFIDF, Naive Bayes, Nearest Neighbor, Neural Networks, Decision Trees
Sorensen and Mc Elligott [13, 45]	n-gram graph (only bigrams)	weighting graph edges	connectionist combined with Genetic Algorithms
Yang [47]	bag-of-words	stop list	k-Nearest Neighbor

Table 2: Document representation, feature selection and learning algorithms used in some text-learning approaches. Bag-of-words is used on Boolean features unless notified that word frequency is used (freq).

enough information available describing the insides of the system. We additionally included descriptions of some research work that does not necessary include development of a working intelligent agents and thus was not included in Table 1.

(1) The frequently used document representation in Information Retrieval and text-learning is the so called $TFIDF$-vector representation. It is a *bag-of-words representation*: all words from the document are taken and no ordering of words or any structure of the text is used (see Figure 4). There is additional information in text documents that could be used, for example, structure of sentences, position of words or neighboring words. The question is how much can we gain considering additional information in learning (and what information to consider) and what is the price we have to pay for it? There is currently no well studied comparison or directions for text document representation that we are aware of. There is some evidence in Information Retrieval research, that for long documents, considering

information additional to bag-of-words is not worth efforts.

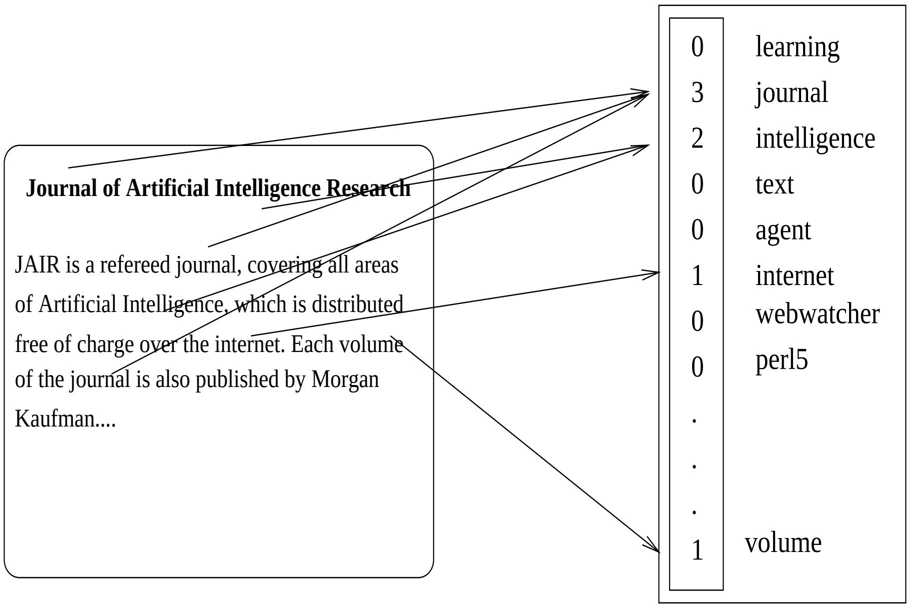

Figure 4: Illustration of bag-of-words document representation using frequency vector. Each word from a document is mapped into a feature and assigned a frequency (the number of times it occurs in the document).

Many current systems that learn on text use the bag-of-words representation using either Boolean features indicating if a specific word occurred in a document (eg. [3, 11, 32, 34, 40, 41, 47]) or frequency of a word in a given document (eg. [2, 4, 6, 7, 21, 29, 47]). There is also some work that uses additional information such as word position [11] or word tuples called n-grams [13, 45] (eg., "machine learning" is a 2-gram and "World Wide Web" is a 3-gram).

(2) One of the frequently used approaches to reduce the number of different words is to use "stop-list" containing common English words like "a", "the", "with" (eg. [2, 4, 40, 41, 47]) or pruning the infrequent and/or very frequent words (eg. [11, 21]). Connected to the particular language is also word stemming that reduces the number of different words using a language-specific stemming algorithm (eg., "works", "working", "worked" are all replaced by "work"). Many approaches use language independent approach and introduce some sort of word weighting in order to select only the best words (eg. [2, 3, 4, 21, 29, 32, 40, 41, 48]) or reduce dimensionality using latent semantic indexing with singular value decomposition (eg. [6, 7, 15]).

(3) One of the well-established techniques for text classification in Information Retrieval is to represent each document with bag-of-words as a $TFIDF$-vector in the space of words that appear in training documents [44], sum all interest-

ing document vectors and use the resulting vector as a model for classification (based on [42] relevance feedback method). 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 $IDF = log\frac{D}{DF(w_i)}$ (Inverse document Frequency), where D is the number of documents and document frequency $DF(w_i)$ is the number of documents word w_i occurred in at least once. The exact formulas used in different approaches may slightly vary (some factors are added, normalization performed [44]) but the idea remains the same. A new document is then represented as a vector in the same vector space as the generated model and the distance between them is measured (usually using cosine similarity measure) in order to classify the document. This technique has already been used in Machine Learning experiments on World Wide Web data (eg. [3, 4, 7, 21, 40, 41]).

Armstrong et al. [3] used a statistical approach they called WordStat that assumes mutual independence of words. Pazzani et al. [40, 41] used in their experiments a Naive Bayesian classifier on Boolean word-vectors, Nearest Neighbor and symbolic learning using Decision Trees. A variant of k-Nearest Neighbor was also used by Yang [47]. Joachims [21] introduced Probabilistic $TFIDF$ that takes into account document representation. Apté et al. [2] used Decision Rules. Cohen [11] used Decision Rules and the Inductive Logic Programming (ILP) systems FOIL and FLIPPER. Lewis and Gale [32] used a combination of a Naive Bayesian classifier and logistic regression. Maes [34] used Memory-Based reasoning. McElligot and Sorensen [13, 45] used a connectionist approach combined with Genetic Algorithms, Lam et al. [29] used Bayesian Network Induction. There is currently no strong evidence about the superiority of any of the given algorithms for text-learning over different domains. Comparison of some learning algorithms given by Pazzani and Billsus [41] shows that document representation including feature selection is more promising source of Classification accuracy improvement than finding better learning algorithm.

4 Text-learning for user customized Web browsing

One of the available information sources is the World Wide Web and it is currently growing quickly, attracting many users with different interests. Since interaction with the World Wide Web (WWW) is by means of computer, one can use computers to "observe" and record user actions, to communicate with users and use all the collected information to help the users find their way in the WWW. Text-learning can be applied on collected information to help user browsing the Web. Our work on Personal WebWatcher [38] is mainly inspired by WebWatcher: "a Learning Apprentice for the World Wide Web" [3, 20] and some other work related to learning apprentice and learning from text [21, 26, 31, 36]. The idea of a learning apprentice is to automatically customize to individual users, using each user interaction as a training example. Personal WebWatcher is designed to be installed locally by the user and connected to the Web browser as a proxy server. The prototype version of the system is available and used for research purposes.

Personal WebWatcher can be seen as a content-based personal assistant that helps user browsing the Web. Imagine a user sitting behind a Web browser and requesting documents, most of the time by clicking on the hyperlinks on already requested documents that are presented to the user. Predicting and highlighting the clicked hyperlinks is one of the ways to help users finding their way on the Web. In the machine learning setting we used, all the hyperlinks that were shown to the user are considered as training examples with a Boolean class value (clicked or un-clicked). We build descriptions of hyperlinks, and treat these as shortened documents used as machine learning examples. Training examples are collected on-line from each interaction with the user. Data collected for a single user represents a domain to be handled by machine learning techniques. The first version of the system uses a bag-of-word document representation, feature selection using informativity and a naive Bayesian classifier.

4.1 Structure of Personal WebWatcher

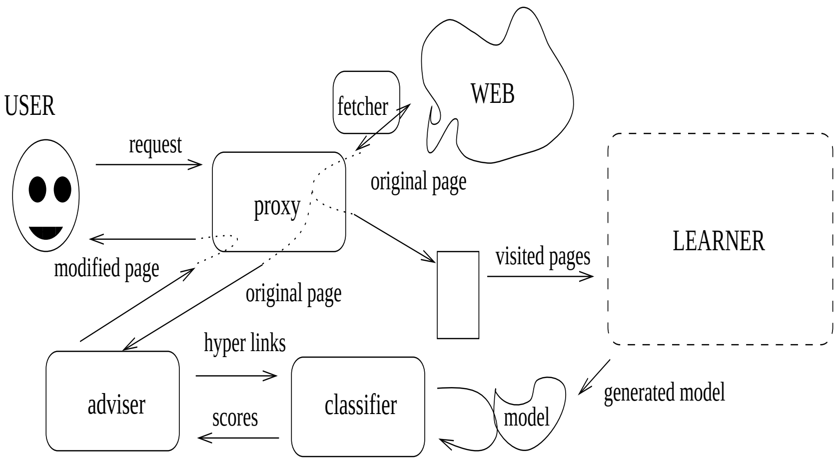

Figure 5: Structure of Personal WebWatcher, an assistant for Web browsing.

The idea is to help user browsing the Web without putting any additional work on the user. The only work is the initial work with installing the system and simply connecting it to the used Web browser as a proxy server. Each request goes from the user to our system and the original of the requested Web page is retrieved from the Web. The original is stored to disk for learning and processed for adding the suggestions. The modified Web page is sent to the user instead of the original one. The modification we introduce does not remove any information from the Web page, but only adds small icons highlighting potentially interesting hyperlinks on the page.

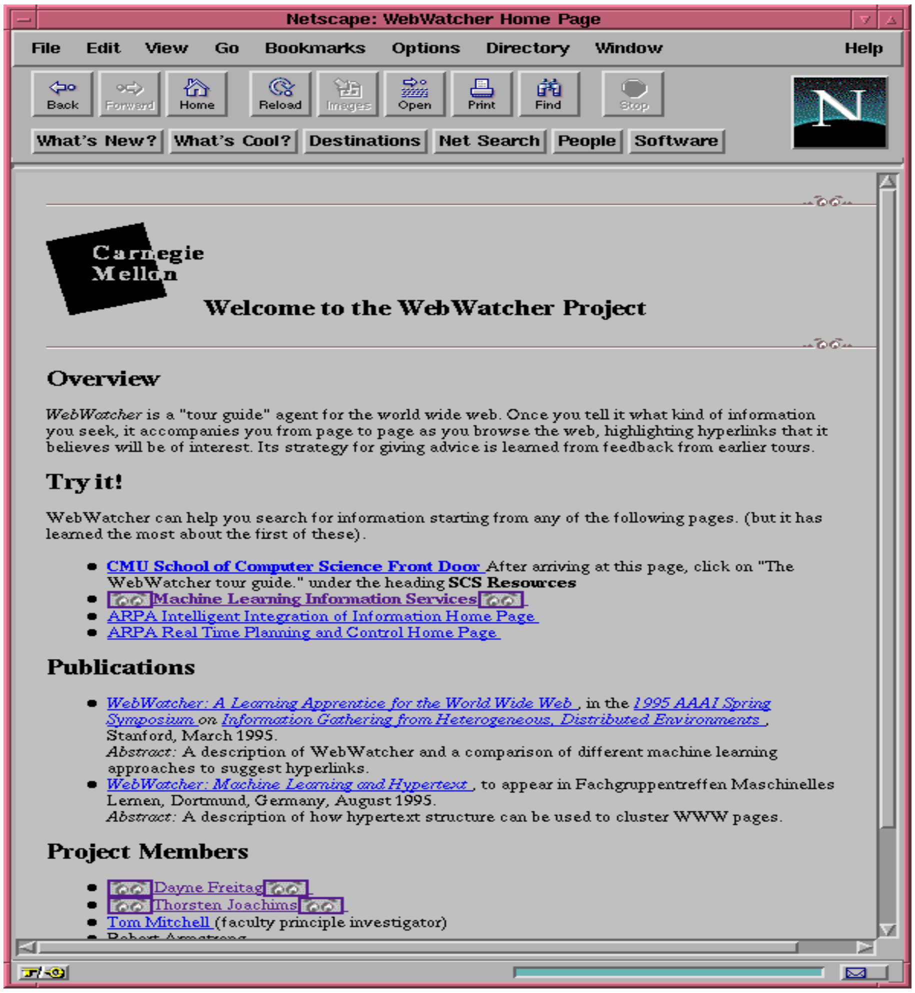

Figure 6: Example of HTML-page presented to the user by Personal WebWatcher. Notice that three hyperlinks are highlighted as interesting ("Machine Learning Information Services" and two project members: Dayne Freitag, Thorsten Joachims).

Personal WebWatcher consists of two main parts: a **proxy server** that interacts with the user via Web browser and a **learner** that provides the user-model to the server (see Figure 5). Communication between them is via disk; the **proxy** saves addresses of visited documents (URLs) and the **learner** uses them to generate 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 **adviser**, that extracts descriptions of hyperlinks from the document and calls **classification** that uses generated user-model.

Figure 6 gives an example page presented via NETSCAPE browser by Personal WebWatcher. The presented document is actually the WebWatcher project page. Once run, the systems processes 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 some threshold are recommended to the user, indicating their scores using graphical symbols placed around highly scored hyperlinks. For example, in Figure 6 three hyperlinks are suggested by Personal WebWatcher: "Machine Learning Information Services" and two project members (Dayne Freitag, Thorsten Joachims).

5 Summary and future directions

The paper gives an overview of some of the recent work in intelligent agents that is based on the usage of machine learning techniques. Two frequently used approaches are described: content-based and collaborative approach. Most of the current agents aim at helping user in browsing World Wide Web, Usenet news filtering and finding experts for a given topic. Content-based agent that work on text-data use methods from Information Retrieval and Machine Learning. Different methods were used on different data, but it is still not clear what is the best approach to use when developing a new system. We analyze recent work in text-learning through the prism of the three interesting research questions: (1) what representation is used for documents, (2) how is the high number of features dealt with and (3) which learning algorithm is used.

In our Personal WebWatcher project there is an on-going research on different aspects of using text-learning on that problem including richer document representation and dynamic construction of background knowledge, comparison of different feature scoring measures on Web data and making prediction based on very short documents [39]. The current trends of using machine learning for intelligent agents include combination of text processing with speech recognition and content processing of image/video. Important direction of text learning is work on information extraction specialized for different domains, such as finding paper citations on the Web (CiteSeer [8]) or building a knowledge database from the Web (WebKB [37]).

Acknowledgements

This work was financially supported by the Slovenian Ministry for Science and Technology. Part of this work was performed during the authors stay at Carnegie

Mellon University in Tom Mitchell's group.

References

[1] Ackerman, M., Billsus, D., Gaffney, S., Hettich, S., Khoo, G., Kim, D.J., Klefstad, R., Lowe, C., Ludeman, A., Muramatsu, J., Omori, K., Pazzani, M.J., Semler, D., Starr, B., Yap, P., Learning Probabilistic User Profiles, *AI magazine*, pp. 47-56, Vol. 18, no. 2, Summer 1997.

[2] Apté, C., Damerau, F., Weiss, S.M., Toward Language Independent Automated Learning of Text Categorization Models, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dubline, 1994.

[3] Armstrong, R., Freitag, D., Joachims, T., Mitchell, T., WebWatcher: A Learning Apprentice for the World Wide Web, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. http://www.cs.cmu.edu/afs/cs/project/theo-6/web-agent/www/webagents-plus.ps.Z

[4] Balabanović, M., Shoham, Y., Learning Information Retrieval Agents: Experiments with Automated Web Browsing, *AAAI 1995 Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. http://robotics.stanford.edu/people/marko/papers/lira.ps

[5] Balabanović, M., Shoham, Y., Fab: Content-Based, Collaborative Recommendation, *Communications of the ACM*, pp. 66-70, Vol. 40, no. 3, March 1997.

[6] Bartell, B.T., Cottrell, G.W., Belew, R.K., Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling, *Proceedings of the ACM SIG Information Retrieval*, Copenhagen, 1992.

[7] Berry, M.W., Dumais, S.T., OBrein, G.W., Using linear algebra for intelligent information retrieval. *SIAM Review*, Vol. 37, No. 4., pp. 573–595, December 1995.

[8] Bollacker, K., Lawrence, S., Giles, L., CiteSeer: An Autonomous System for Processing and Organizing Scientific Literature on the Web, Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98, 1998.

[9] Burke, R., Hammond, K., Kozlovsky, J., Knowledge-based Information Retrieval for Semi-Structured Text, *Working Notes from AAAI Fall Symposium on AI Applications in Knowledge Navigation and Retrieval*, pp. 19–24, American Association for Artificial Intelligence, 1995.

[10] Burke, R., Hammond, K., Kulyukin, V., Lytinen, S., Tomuro, N., Schoenberg, S., Question Answering from Frequently Asked Question Files, *AI magazine*, pp. 57-66, Vol. 18, no. 2, Summer 1997.

[11] Cohen, W.W., Learning to Classify English Text with ILP Methods, *Workshop on Inductive Logic Programming*, Leuven, September 1995.

[12] Drummond,C., Ionescu, D., Holte, R., A Learning Agent that Assists the Browsing of Software Libraries, *Technical Report TR-95-12*, Computer Science Dept., University of Ottawa, 1995.

[13] Mc Elligott, M., Sorensen, H., An emergent approach to information filtering, *Abakus. U.C.C. Computer Science Journal*, Vol 1, No. 4, December 1993.

[14] Etizioni, O., Weld, D., A Softbot-Based Interface to the Internet, *Communications of the ACM* Vol. 37, No. 7, pp.72–79, July 1994.

[15] Foltz, P. W. and Dumais, S. T., Personalized information delivery: An analysis of information filtering methods, *Communications of the ACM*, 35(12), pp.51-60, 1992.

[16] Goldman, C.V., Langer, A., Rosenschein, J.S., Musag: an agent that learns what you mean, *Applied Artificial Intelligence*, 11, pp. 413-435, 1997.

[17] Hammond, K., Burke, R., Schmitt, K., A Case-Based Approach to Knowledge Navigation, *AAAI Workshop on Indexing and Reuse in Multimedia Systems*, pp. 46–57, American Association for Artificial Intelligence, 1994.

[18] Hedberg, S., Agents for sale: first wave of intelligent agents go commercial, *IEEE Expert Intelligent systems & their applications*, pp. 16-19, Vol 11, no. 6, December 1996.

[19] Holte, R.C., Drummond, C., A Learning Apprentice For Browsing, *AAAI Spring Symposium on Software Agents*, 1994.

[20] Joachims, T., Mitchell, T., Freitag, D., Armstrong, R., WebWatcher: Machine Learning and Hypertext, *Fachgruppentreffen Maschinelles Lernen*, Dortmund, August 1995.

[21] Joachims, T., A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp. 143—151, 1997.

[22] Kamba, T., Sakagami, H., Koseki, Y., ANATAGONOMY: a personalized newspaper on the World Wide Web, *International Journal Human-Computer Studies*, 46, pp. 789-803, 1997.

[23] Kautz, H., Selman, B., Shah, M., Referral Web: Combining Social Networks and Collaborative Filtering, *Communications of the ACM*, pp. 63-65, Vol. 40, no. 3, March 1997.

[24] Kautz, H., Selman, B., Shah, M., The Hidden Web, *AI magazine*, pp. 27-36, Vol. 18, no. 2, Summer 1997.

[25] Konstan, J.A., Miller, B.N., Maltz, D., Herlocker, J.L., Gordon, L.R., Riedl, J., GroupLense: Applying Filtering to Usenet News, *Communications of the ACM*, pp. 77-87, Vol. 40, no. 3, March 1997.

[26] de Kroon, H.C.M, Mitchell, T., Kerckhoffs, E.J.H., Improving Learning Accuracy in Information Filtering, *ICML-96 Workshop Machine learning meets human computer interaction*, 1996. http://www.ics.forth.gr/ moustaki/ICML96_HCI_ML/kroon.ps

[27] Krulwich, B., Burkey, C., The ContactFinder agent: Answering buletin board questions with referrals, *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, pp.10-15, 1996.

[28] Krulwich, B., Lifestyle Finder, *AI magazine*, pp. 37-46, Vol. 18, no. 2, Summer 1997.

[29] Lam, W., Low, K.F., Ho, C.Y., Using Bayesian Network Induction Approach for Text Categorization, *15th International Joint Conference on Artificial Intelligence IJCAI97*, pp. 745—750, 1995.

[30] LaMacchia, B.A., Internet Fish, *A revised version of a thesis proposal at MIT*, 1996.

[31] Lang, K., News Weeder: Learning to Filter Netnews, *Proc. of the 12th International Conference on Machine Learning ICML95*, 1995.

[32] Lewis, D.,D., Gale, W., A., A Sequential Algorithm for Training Text Classifiers, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dubline, 1994.

[33] Lieberman, H., Letizia: An Agent that Assists Web Browsing, *14th International Joint Conference on Artificial Intelligence IJCAI95*, 1995.

[34] Maes, P., Agents that Reduce Work and Information Overload, *Communications of the ACM* Vol. 37, No. 7, pp.30–40, July 1994.

[35] Menczer, F., ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhood for Information Discovery, *Proc. of the Fourteenth International Conference on Machine Learning ICML '97*, pp. 227-235, July, 1997.

[36] Mitchell, T., Caruana, R., Freitag, D., McDermott, J., Zabowski, D., Experience with a Learning Personal Assistant, *Communications of the ACM* Vol. 37, No. 7, pp.81–91, July 1994.
http://www.cs.cmu.edu/afs/cs/user/mitchell/ftp/cacm.ps.Z

[37] Mitchell, T., et al., The World Wide Knowledge Base project, `small http://www.cs.cmu.edu/~TextLearning`, 1998.

[38] Mladenić, D., Personal WebWatcher: Implementation and Design, *Technical Report IJS-DP-7472*, October, 1996.
http://www.cs.cmu.edu/~TextLearning/pww/

[39] Mladenić, D., Machine Learning on non-homogeneous, distributed text data, *PhD thesisi proposal*, University of Ljubljana, Slovenia, June, 1997.
http://www.cs.cmu.edu/~TextLearning/pww/

[40] Pazzani, M., Muramatsu, J., Billsus, D., Syskill & Webert: Identifying interesting web sites, *AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, March 1996 and *Proceedings of the Thirteenth National Conference on Artificial Intelligence AAAI 96*, pp.54-61, 1996.

[41] Pazzani, M., Billsus, D., Learning and Revising User Profiles: The Identification of Interesting Web Sites, *Machine Learning 27*, Kluwer Academic Publishers, pp. 313—331, 1997.

[42] Rocchio, J., Relevance Feedback in Information Retrieval, in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Chapter 14, pp.313-323, Prentice-Hall Inc., 1971.

[43] Rucker, J., Marcos, J.P., Siteseer: Personalized Navigation for the Web, *Communications of the ACM*, pp. 73-75, Vol. 40, no. 3, March 1997.

[44] Salton, G., Buckley, C., Term Weighting Approaches in Automatic Text Retrieval, *Technical report, COR-87-881*, Department of Computer Science, Cornell University, November 1987.

[45] Sorensen, H., McElligott, M., PSUN: A Profiling System for Usenet News, *CIKM'95 Intelligent Information Agents Workshop*, Baltimore, December 1995.

[46] Terveen, L., Hill, W., Amento, B., McDonald, D., Creter, J., PHOAKS: A System for Sharing Recommendations, *Communications of the ACM*, pp. 59-62, Vol. 40, no. 3, March 1997.

[47] Yang, Y., Expert Network: Effective and Efficient Learning form Human Decisions in Text Categorization and Retrieval, *Proc. of the 7th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dubline, 1994.

[48] Yang, Y., Pedersen, J.O., A Comparative Study on Feature Selection in Text Categorization, *Proc. of the 14th International Conference on Machine Learning ICML97*, pp. 412-420 1997.