Text-learning and related intelligent agents

Dunja Mladenić
Department of Intelligent Systems, J.Stefan Institute,
Jamova 39, 1000 Ljubljana, Slovenia
Phone: +386-61-1773 272, Fax: +386-61-1251-038

Revised version In IEEE Expert special issue on
Applications of Intelligent Information Retrieval, July-August 1999

Abstract

Analysis of text data using intelligent information retrieval, machine learning, natural language processing or other related methods is becoming an important issue for the development of intelligent agents. There are two frequently used approaches to the development of intelligent agents using machine learning techniques: a content-based and a collaborative approach. In the first approach, the content (eg., text) plays an important role, while in the second approach, the existence of several knowledge sources (eg., several users) is required. We can say that the usage of machine learning techniques on text databases (usually referred to as *text-learning*) is an important part of the content-based approach. Examples are agents for locating information on World Wide Web and Usenet news filtering agents. There are different research questions important for the development of text-learning intelligent agents. We focus on three of them: 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 in an overview of the existing approaches to text classification. For illustration we give a brief description of the content-based personal intelligent agent named Personal WebWatcher that uses text-learning for user customized Web browsing.

1 Introduction

One of the known characteristics of this time is the information overload people are faced with. There is a frequent impression that simply a day has too few hours and there are so many things waiting that 'had to be finished by yesterday'. This impression is so realistic that any help in handling at least some simpler tasks

would be appreciated. Recently there is a rapidly growing importance of the Internet not only in the research community but also broader in many every-day activities. This results in a number of systems that aim to help users in different activities. The growing interest and number of ideas where computers could be used is also pressing the research community to develop new methods and extend the existing approaches. One of especially popular ways of using Internet is via World Wide Web (WWW) as a decentralized way of sharing information.

General trends in our modern society toward decentralization and synergy based on the respect for differences between people, also contribute to the Web popularity. As usually when different people are put together without centralized rules and guidance, many creative and nice as well as some less nice aspects come to the surface. When putting information on the Web, every person can decide what to put and how to organize it. This enables the development of a distributed, world wide accessible information source containing non-homogeneous data organized according to different human association models and value schemes. This 'personal touch' and opportunity for creativity can be a pleasant experience for a human when providing and obtaining information but hard to cope with for most computer systems and many times also for people that want a fast way to find information. Large amounts of information available in electronic form on the Web presents a challenge for the research community by its distributed organization, arbitrary mixture of text, speech, image and video in the same document, dynamic nature of the provided information, the fact that the current trends and interests in different areas reflect on the Web, different target public and communities that meet on the Web talking about research, advertising and selling, providing useful advises like how to prepare food or where to go hiking. Development of machine learning techniques for learning on large, distributed text data covers just a small part of the raising problems [1]. One of the commonly addressed, rather general problems is providing help to users in browsing the Web. Many of the systems described here (see also Table 1) address this problem, some based on the content analysis using mostly text from the Web documents while the others based on some other information about document relevancy such as users' rating.

Recent developments at the intersection of Information Retrieval and Machine Learning, as well as work in Intelligent Agents and Intelligent User Interfaces offer some novel solutions helping users in making a good and quick selection of information they are interested in. This results in the intensive development of methods for using machine learning techniques on text databases referred to as *text-learning*, that combine research in machine learning with information retrieval. This paper covers a part of text-learning where supervised learning methods are used for text classification.

The rest of this paper is organized as follows. Section 2 gives the description of some work related to the usage of machine learning techniques in the development

of intelligent agents. Section 3 gives an analysis of different approaches to the classification of text data. 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 the structure of the system. Section 5 gives the summary and outlines some interesting directions for future research in text-learning and learning on the Web.

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 are not discussed in this paper. We rather present examples of some systems called intelligent agents (user assistant, recommendation systems) that involve the usage of machine learning [2] or data mining techniques [3]. All these systems aim at helping the user by assisting her/him in finding information or performing some simpler tasks on behalf of the user. For instance, providing help in Web browsing by retrieving documents similar to the already requested documents [4].

Two approaches to the development of intelligent agents based on machine learning techniques are frequently used [5]: *content-based* and *collaborative*. Both approaches have been successfully used for finding and retrieving relevant information from the Web. The next two sections, Sections 2.1 and 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 the content-based or the collaborative approach. Table 1 contains a summary of the described systems. For each system we give a reference to the paper describing the system, a name of the system, a name of the organization that developed it and a few word summary of its functionality. There are many other intelligent agents not mentioned here but our hope is to give a general idea about the current trends.

2.1 The content-based approach

In the content-based approach to text classification the system searches for the items similar to those user liked based on the content comparison. The approach has its roots in information retrieval. One of the main problems with this approach is to capture different aspects of the content (eg., music, movies, images). Even for text domains, most representations capture only certain aspects of the content which results in poor system performance. Additionally to the representational problems, content-based systems tend to learn in a way that items similar to the already seen items are recommended. When the content based approach is applied on text data, different text-learning methods can be used.

Paper reference	Agent name	Developed at	Goal is to help in
Content-based agents			
Armstrong et al. [6]	WebWatcher	CMU	browsing WWW
Balabanović et al. [7]	Lira	Stanford	browsing WWW
Goldman et al. [8]	Musag	Hebrew Uni.	browsing WWW
Lieberman [9]	Letizia	MIT	browsing WWW
Mladenić [4]	Personal WebWatcher	CMU, IJS	browsing WWW
Pazzani et al. [10, 11]	Syskill & Webert	UCI	browsing WWW
Shavlik et al. [12]	WAWA	Wisconsin Uni.	browsing WWW
Bollacker et al. [13]	CiteSeer	TX, NEC corp., UMIACS	finding papers on WWW
Lang [14]	NewsWeeder	CMU	Usenet news filtering
Krulwich et al. [15]	ContactFinder	Andersen Consult.	finding expert
Burke et al. [16, 17]	FAQFinder	Uni. of Chicago	answer question
Kamba et al. [18]	Antagonomy	NEC corp.	personalized newspaper
LaMacchia [19]	Internet Fish	MIT	find info. on Internet
Mitchell et al. [20]	Calendar Apprentice	CMU	meeting scheduling
Collaborative agents			
Rucker and Marcos [21]	Siteseer	Imana, Inc.	browsing WWW
Terveen et al. [22]	PHOAKS	AT&T labs.	browsing WWW
Konstan et al. [23]	GroupLense	Uni. Minnesota	Usenet news filtering
Kautz et al. [24, 25]	Referral Web	AT&T labs.	finding expert
Maes [26]	Ringo Firefly	MIT MIT	finding music music, movie, book
Combination of content-based and collaborative agent			
Balabanović et al. [5]	Fab	Stanford	browsing WWW
Krulwich [27]	Lifestyle Finder	AgentSoft Ltd.	browsing WWW
de Vel et al. [28]	webCobra	James Cook Uni.	browsing WWW

Table 1: Summary of some content-based and collaborative systems that use some machine learning related techniques. For each system we give a reference, a name, a development organization and main functionality.

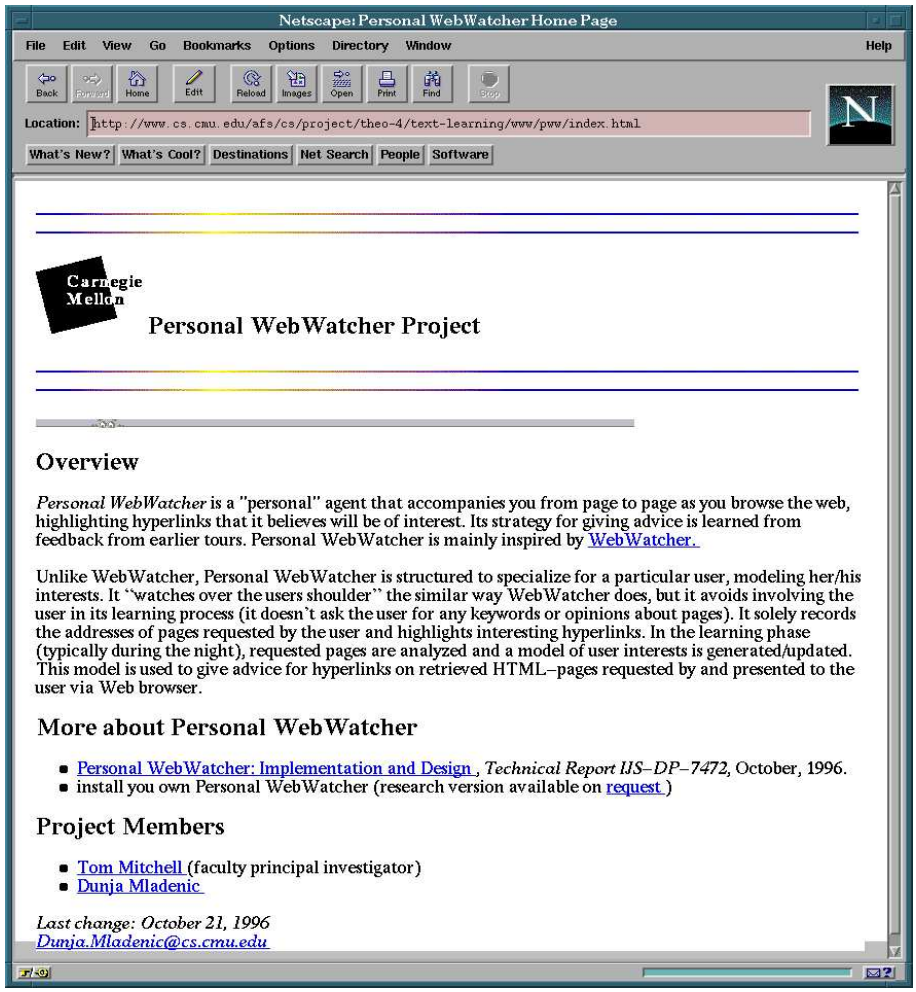

Figure 1: Example of the Web pages that a user found interesting.

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 given in Figure 1, the Web page at the left hand-side of Figure 2 will be recommended by the content-based approach since it has a similar content as the page in Figure 1. The Web page in the right-hand side of Figure 2 would be recognized as different and thus would not be recommended (unless the user found interesting also some other pages that are similar to it).

The 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. [6] developed `WebWatcher`, a system that assists the user in locating information on the World Wide Web by 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 typed by different users, hyperlinks that were followed and

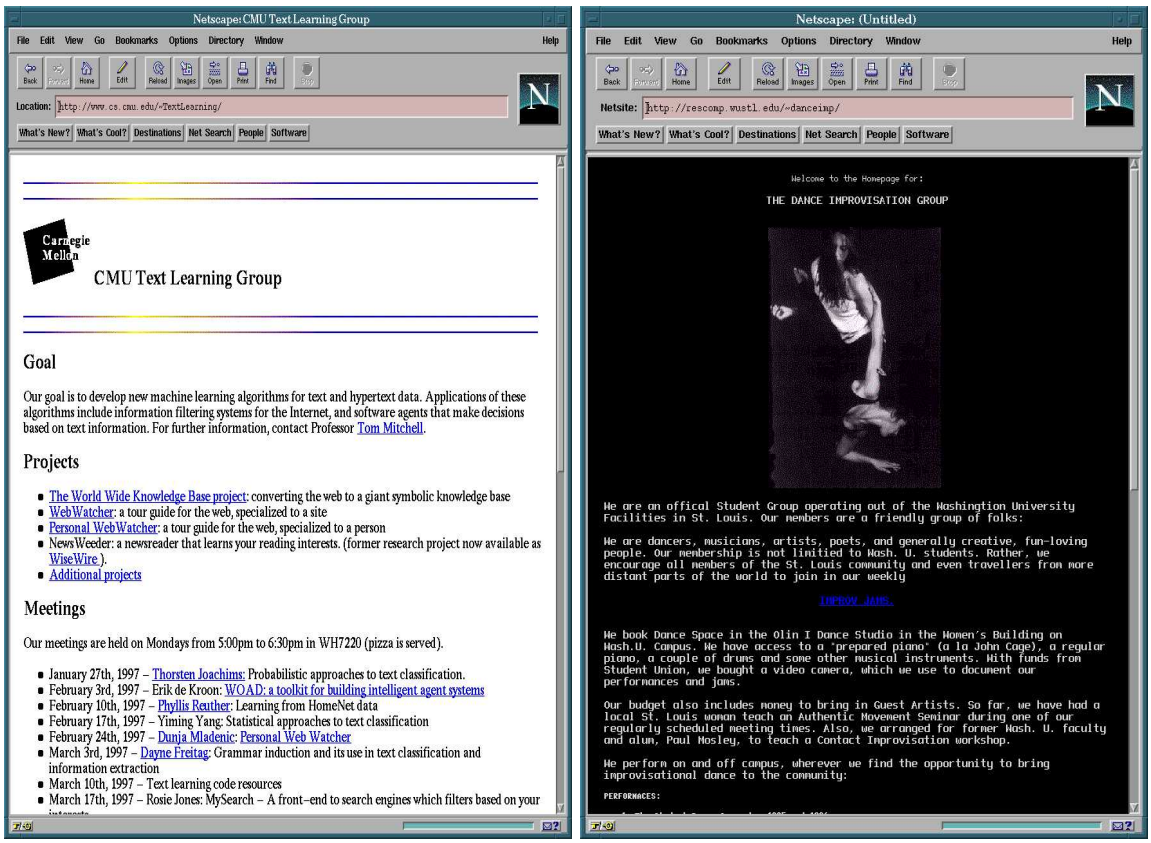

Figure 2: Example of two Web pages having different content and thus recognized as not similar by the content-based approach. The left-hand side page would be recognized as similar to the page in the previous Figure and recommended by content-based approach, while the right-hand side page would be recognized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

the evaluation given by the users at the end of each search. Balabanović and Shoham [7] developed "a system which learns to browse the Internet on behalf of a user". It searches the World Wide Web by taking bounded amount of time, selects the best pages and receives an evaluation from the user. The evaluation is used to update the search and selection heuristics. Goldman et al. [8] 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 [9] 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. Mladenić [4] developed Personal WebWatcher, a personal assistant for Web browsing that accompanies the user from page to page and highlights interesting hyperlinks. The system generates a user profile based on the content analysis of the requested pages without requesting any keywords or ratings from the user. Pazzani et al. [10] and Ackerman et al. [29, 11] 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. Shavlik and Eliassi [12] developed WAWA, an intelligent agent for Web-based tasks. Their system enables the user to input personal interests and preferences, stores then in a neural network and uses theory-revision to refine the obtained knowledge. They present a case study on the problem of finding a person's home-page on the Web.

Bollacker et al. [13] developed CiteSeer that assists the user in finding relevant Web-based research publications. The system gets keywords from the user and calls search engines to find relevant papers (relevant PostScript files on the Web) and extracts header, abstract, citations from the papers. It also enables finding of similar papers based on the common citations in the papers. Lang [14] developed NewsWeeder, a system for electronic news filtering that uses text classification to generate a model of user's interests. The system uses a 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 [15] 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 the bulletin board looking for the contact persons and their topic areas and storing the 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, it extracts semantically significant phrases (eg., fully capitalized words, short phrases: 1 to 5 words, words in a different format from the surrounding text). Hammond et al. [30] and Burke et al. [16, 17] 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 the five best

matching questions are returned together with their answers. Kamba et al. [18] 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 [19] 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. [20] proposed `Calendar Apprentice`, a system that helps the 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 meetings. It uses these rules to provide advice to the user for new, unscheduled meetings.

2.2 The collaborative approach

In contrast with the content-based approach to text classification that 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 [26]) advise 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 the users. The assumption is that the users provide ratings for the seen items. In the 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 usually results in 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 a sufficient number of same items. The problem is also if some user has an 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 a similar music taste and the music that she/he likes will be recommended to `User1` as probably interesting. On the contrary, `User3` would be recognized as having a different music taste from User1 and the music she/he likes 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, the collaborative approach would find `User2` similar to `User1`, while `User3` would be found different from `User1`.

The collaborative approach is usually used for non-text data (eg., movies, music), but there are also systems that use it on text data (eg., for news filtering). Maes [26] describes "interface agents" that learn from the user as well as from other agents. Examples of interface agents they developed are 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 the correlation between different users performing collaborative filtering (eg., entertainment recommendation). For instance, a music recommendation system `Ringo`, that they developed, recommends music that was highly scored by users with similar music tastes. The system tries to overcome the problem of the sparse coverage of ratings by 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 [31]. `Firefly` requires from the user to start by 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). Rucker and Marcos [21] 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 obtained from similar folders and URLs that appear in bookmark files of similar users. Terveen et al. [22] propose `PHOAKS`, a system that automatically recognizes and redistributes recommendations of the Web resources mined from the Usenet news messages. The systems

assumes that the roles of the provider and the recommendation recipient are specialized and different. It reuses recommendations from the existing online conversations. A special attention is paid 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. Konstan et al. [23] 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 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. [24, 25] 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 the content-based and the collaborative approach. The content-based approach is used to generate a profile that represents a single user's interests and the 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. The similarity between users is measured by the similarity of their profiles. De Vel and Nesbitt [28] used a similar idea as Balabanović and Shoham [5] in combining the content-based and the collaborative approach to text classification. They generate a user profile from relevant documents using a part-of-speech tagger. Users are grouped into collaborative clusters based on their profile similarity. Krulwich [27] developed `Lifestyle finder`, a system for user profile generation based on the usage of demographic data. The system combines the content-based and the collaborative approach. A questionnaire is used to get user's characteristics and group similar users in terms of demographic data. A set of the 15 highly scored Web documents is suggested to each user. User's evaluation of suggested documents is used to evaluate the system performance.

2.3 Some other related systems

Holte and Drummond [32], Drummond et al. [33] 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 finding a single item. Etizioni and Weld [34] developed an integrated interface to the Internet combining a 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. Gams and Grobelnik [35] developed an employment agent available through the Internet that enables user browsing data and ordering e-mails when an interesting information appears. Menczer [36] developed "adaptive intelligent methods to automate on-line information search and discovery in the Web" (like Web robots or crawlers) based on a population of intelligent agents. 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.

3 Machine learning on text data

Most of the intelligent agents that use machine learning techniques actually use the 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 on text documents that is not necessarily related to the Web. Here we take a look at some of this work through the prism of three questions we find important for using machine learning for text classification: (1) what representation is used for documents, (2) how is the high number of features dealt with and (3) which learning algorithm is used. Tables 2 and 3 summarize them over some related papers in order to give an idea about the state-of-the-art in using unsupervised learning on text data. Systems given in Table 1 are included in this more detailed analysis, if there is enough information available describing the insides of the system. We additionally include descriptions of some research work that does not necessary include the development of a working intelligent agents and was therefore not included in Table 1.

(1) The frequently used document representation in information retrieval and text-learning is the so called 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. When having 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 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 the bag-of-words is not worth the efforts. There is also work on document classification that extends the bag-of-words representation by using word sequences (n-grams) instead of single words [37, 38]. This work suggests that the usage of single words and word pairs as features in the bag-of-words representation improve performance of classifiers generated from short documents.

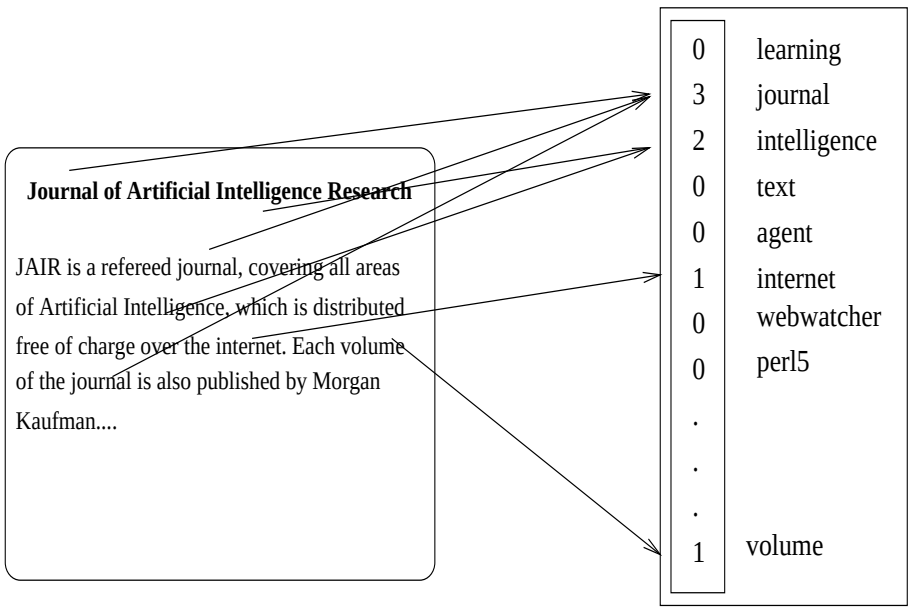

Figure 4: Illustration of the bag-of-words document representation using frequency vector. Each word occurring in a set of documents is mapped into a feature. Document is represented as a vector of features where each feature is assigned a frequency (the number of times it occurs in the document).

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., [6, 39, 40, 41, 42, 43, 44, 26, 45, 46, 10, 11, 12, 47, 48]) or the frequency of a word in a given document (eg., [49, 50, 7, 51, 52, 53, 54, 55, 56, 4, 37, 48, 57]). There is also some work that uses additional information such as word position [40, 58, 12] or word tuples called n-grams [59, 37, 38, 60] (eg., "machine learning" is a 2-gram and "World Wide Web" is a 3-gram). Some recent work [47] indicates that the usage 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 showing clear advantages of some representations.

(2) 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", "with" (eg., [49, 7, 56, 43, 37, 10, 11, 12, 61]) or pruning the infrequent words (word frequency $<$ min.frequency) (eg., [40, 53, 54, 37]). Connected to the particular language is also word stemming, used for example in [50, 7, 12, 61], 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 a language independent approach and introduce some sort of word scoring in order to select only the best words (eg., [49, 6, 7, 53, 55, 42, 43, 37, 45, 10, 11, 47, 61, 62]) or reduce the dimensionality using latent semantic indexing with singular value decomposition (LSI) (eg., [51, 52, 63, 61]).

Experiments with different numbers of selected features used in text classification [64, 1, 62, 61] indicate that the best results are obtained either using only a small percentage of carefully selected features (up to 10% of all features) or in some case using all the features. 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 used classification algorithm characteristics [1]. Surprisingly good results are obtained using a simple frequency measure in a combination with a 'stop-list' [1, 65, 37, 62].

(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 [66], sum all interesting document vectors and use the resulting vector as a model for classification (based on the relevance feedback method [67]). 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 [66]) 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 the cosine similarity measure) in order to classify the document. This technique is commonly used to get baseline results when testing some machine learning algorithm [2] on text data. TFIDF classification has already been used in Machine Learning experiments on the World Wide Web data (eg., [6, 7, 52, 53, 10, 11]) and in most cases shown to be inferior to the tested machine learning methods.

An extension of TFIDF proposed by Joachims [53] called Probabilistic $TFIDF$ takes into account document representation and was shown to achieve results

better than TFIDF and comparable to the Naive Bayesian classifier. 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. For instance, the Naive Bayesian classifier was used by Joachims [53], Lewis and Ringuette [43], Mladenić and Grobelnik [4, 37], Pazzani et al. [10, 11]. The Nearest Neighbor algorithm was used by Mladenić [4], Pazzani et al. [10, 11], Yang [48, 57]. In addition to using the Naive Bayesian classifier and Nearest Neighbor, Pazzani et al. [10, 11] as well as Lewis and Ringuette [43] performed experiments on text data with symbolic learning using Decision Trees. Moulinier and Ganascia [45] experimented using Decision Rules. Yang [48, 57] compared the performance of Linear Least Square Fit (LLSF) and a variant of k-Nearest Neighbor, reporting that similar results are achieved by both classifiers. Creecy et al. [39] and Maes [26] used Memory-Based reasoning. Apté et al. [49] used Decision Rules and in [50] boosted Decision Trees. Cohen [40, 58] used Decision Rules, the Sleeping experts algorithm and two Inductive Logic Programming (ILP) algorithms `FOIL` and `FLIPPER`. Slattery and Craven [47] used the Naive Bayesian classifier and two ILP algorithms `FOIL` and `FOIL-PILFS` (`FOIL` with Predicate Invention for Large Feature Spaces). Lewis et al. [42, 64] used a combination of the Naive Bayesian classifier and logistic regression, the Widrow-Hoff algorithm and Exponential Gradient (EG). Wiener et al. [61] performed experiments showing that Neural Networks achieve slightly better results than Logistic Regression. McElligot and Sorensen [59, 60] used a connectionist approach combined with Genetic Algorithms. Lam et al. [55] used Bayesian Network Induction. Gelfand [41] used Semantic Relationship Graphs (SRG) to represent documents based on the WordNet lexical database. Classification is performed in a similar way TFIDF is used; each class is defined by a group of training documents and represented as a union of their SRG representation. Armstrong et al. [6] used the Winnow algorithm and a statistical approach they called WordStat that assumes mutual independence of words. Shavlik and Eliassi [12] used theory-refinement on Neural Networks, where the user provides an initial advice that is compiled into Neural Network and refined during the interaction with the user based on the users page ratings and additionally provided advises. Active learning was used by Liere and Tadepalli [44] where a committee of Winnow learners is used. Active learning was also used by Nigam and McCallum [46] in a combination of Query by Committee and the Expectation Maximization (EM) algorithm.

There is currently no strong evidence about the superiority of any of the given algorithms for text-learning over different domains. Most experiments show the superiority of the tested algorithm over the TFIDF classification. A comparison of some learning algorithms given by Pazzani and Billsus [11] indicates that a document representation including feature selection is more promising approach to classification accuracy improvement than finding a better learning algorithm. In their experiments the Naive Bayesian classifier, Nearest Neighbor and Neural Net-

works achieve the best results on tested data. Similar observations about a good performance of k-Nearest Neighbor, Neural Networks and Linear Least Square Fit was reported by Yang [57], while in the same experiments the Naive Bayesian classifier didn't perform well. On the other hand, Yang [62] and Joachims [54] reported that on their domains feature subset selection was not crucial for the classifier performance. Joachimas [54] reported that Support Vector Machines outperform the Naive Bayesian classifier, while Apté et al. [50] obtained even better results using boosted Decision Trees. Lam et al. [56] observed that the Generalized Instance Set algorithm achieved better results than either k-Nearest Neighbor or linear classifiers (Rocchio, Widrow-Hoff).

Paper reference	**Document Representation**	**Feature Selection**	**Classification**
Apté et al. [49]	bag-of-words (frq)	stop list+ frequency weight	Decision Rules
Apté et al. [50]	bag-of-words (frq)	stemming+min.frq	boosted Decision Trees
Armstrong et al. [6]	bag-of-words	informativity	TFIDF Winnow, WordStat
Balabanović et al. [7]	bag-of-words (frq)	stop-list+stemming	TFIDF
Bartell et al. [51]	bag-of-words (frq)	LSI	—
Berry et al. [52] Foltz et al. [63]	bag-of-words (frq)	LSI (latent semantic indexing using SVD)	TFIDF
Creecy et al. [39]	bag-of-words	—	Memory Based reasoning
Cohen [40]	bag-of-words + word position	min.frq	Decision Rules ILP
Cohen et al. [58]	ordered word list	—	Decision Rules, Sleeping expert
Gelfand [41]	bag-of-words + WordNet	min. connectivity	Semantic Relationship Graph
Joachims [53]	bag-of-words (frq)	min.frq + informativity	TFIDF, PrTFIDF, Naive Bayes
Joachims [54]	bag-of-words (frq)	min.frq	Support Vector Machines
Lam et al. [55]	bag-of-words (frq)	mutual info.	Bayesian Network
Lam et al. [56]	bag-of-words (frq)	stop list	Generalized Instance Set, k-Nearest Neighbor
Lewis et al. [43]	bag-of-words	stop list+ informativity	Naive Bayes, Decision Trees
Lewis et al. [42]	bag-of-words	log likelihood ratio	logistic regression with Naive Bayes
Lewis et al. [64]	bag-of-words (frq)	—	Widrow-Hoff, EG

Table 2: 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). Notice that this is the first part of the whole Table.

Paper reference	Document Representation	Feature Selection	Classification
Liere et al. [44]	bag-of-words	—	Winnow (in Query by Committee)
Maes [26]	bag-of-words+ header info.	selecting keywords	Memory-Based reasoning
Mladenić [4]	bag-of-words (frq)	informativity	Naive Bayes, Nearest Neighbor
Mladenić et al. [37, 38]	bag-of-words using n-grams (frq)	stop list+min.frq+ Odds ratio	Naive Bayes
Moulinier et al. [45]	bag-of-words	informativity	Decision Rules
Nigam et al. [46]	bag-of-words	min.frq	EM with QBC
Pazzani et al. [10, 11]	bag-of-words	stop list+ informativity	TFIDF, Naive Bayes, Nearest Neighbor, Neural Networks, Decision Trees
Shavlik et al. [12]	localized bag-of-words	stop-list+stemming	Theory-Refinement on Neural Networks
Slattery et al. [47]	bag-of-words+ hyper-text/graph	informativity	Naive Bayes, ILP
Sorensen et al. [59, 60]	n-gram graph (only bigrams)	weighting graph edges	connectionist combined with Genetic Algorithms
Wiener et al. [61]	bag-of-words	stop-list+min.frq+ stemming+ relevancy or LSI	Neural Networks, Logistic Regression
Yang [48, 57]	bag-of-words	informativity, χ^2-stat.	k-Nearest Neighbor, LLSF

Table 3: Document representation, feature selection and learning algorithms used in some text-learning approaches. Notice that this is the second part of the whole Table.

4 Text-learning for user customized browsing of the Web

One of the available information sources is the World Wide Web and it is currently growing rapidly, attracting many users with different interests. Since the interaction with the World Wide Web 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 Web. Text-learning can be applied on collected information to help users browsing the Web. Our work on `Personal WebWatcher` [4] is mainly inspired by `WebWatcher`: "a Learning Apprentice for the World Wide Web" [6, 68] and some other work related to learning apprentice and learning from text [53, 69, 14, 20]. 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.

A prototype version of the system is available and used for research purposes (`http://www.cs.cmu.edu/ TextLearning/pww/`).

`Personal WebWatcher` can be seen as a content-based personal agent that helps the user browsing the Web. Imagine a user using some Web browser for requesting documents, most of the time by clicking of 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 training examples for machine learning. 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 [4] uses a bag-of-word document representation, feature selection using informativity and a Naive Bayesian classifier.

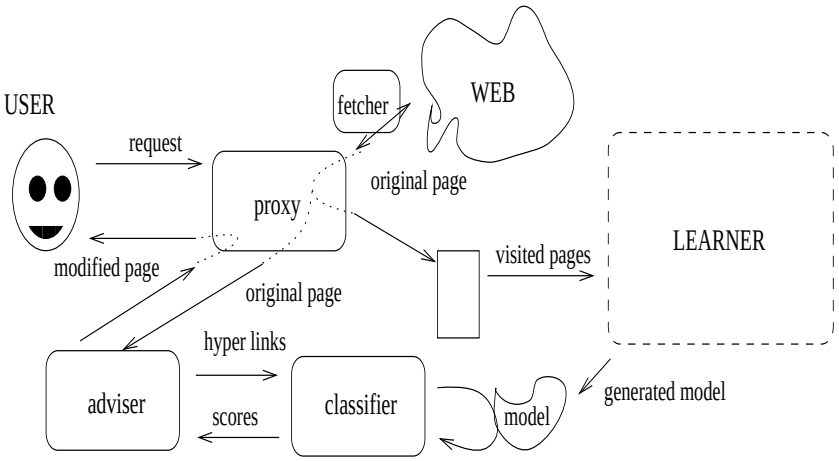

Figure 5: Structure of Personal WebWatcher, an assistant for Web browsing.

4.1 The structure of Personal WebWatcher

The idea is to help the user browsing the Web without putting any additional workload on the user. The only work is the installing of 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 on a 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.

`Personal WebWatcher` consists of two main parts: a *proxy server* that interacts with the user via the Web browser and a *learner* that provides the user-model to the server (see Figure 5). The communication between them is via a disk; the *proxy* saves addresses of visited documents (URLs) and the *learner* uses them for retrieving 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 *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 the NETSCAPE browser by `Personal WebWatcher`. The presented document is actually the WebWatcher project page. Once run, the systems 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 6 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 build from about 500 documents visited in 1996 by Dunja Mladenić.

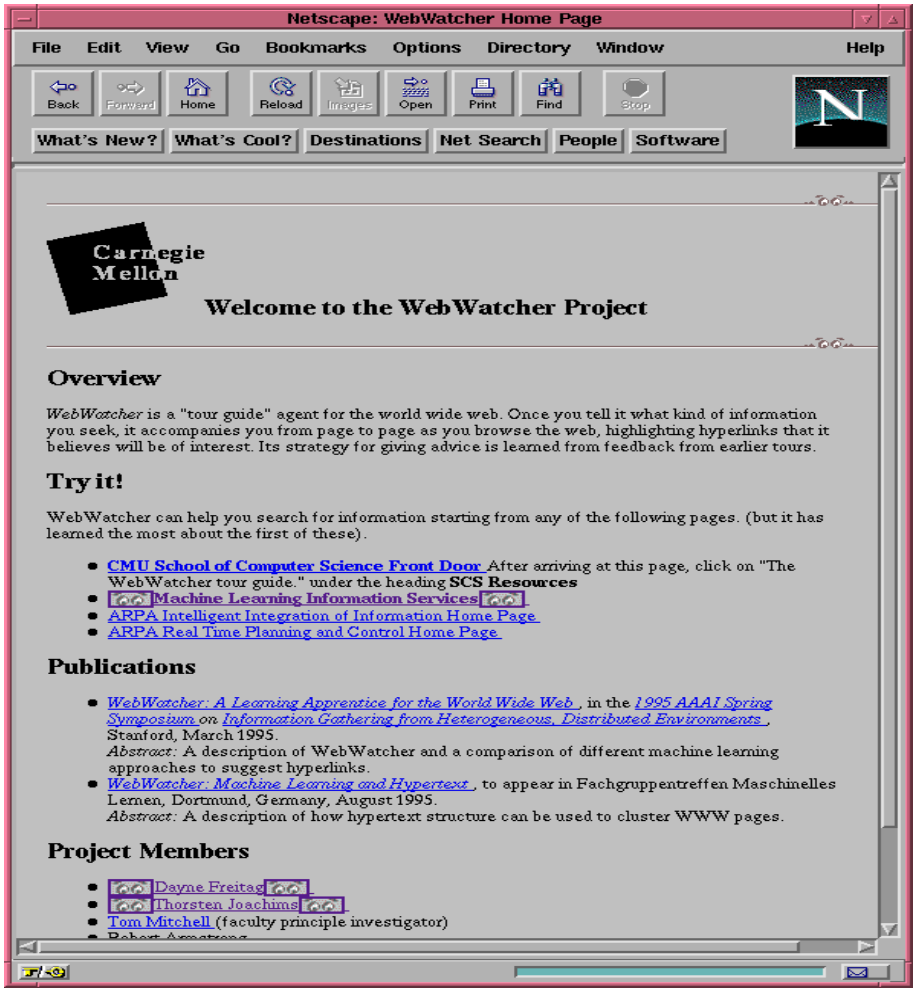

Figure 6: Example of HTML-page presented to the user by Personal WebWatcher. Notice that three hyperlinks are highlighted as interesting ("Machine Learning Information Services" and two project members: Dayne Freitag, Thorsten Joachims).

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: the content-based and the collaborative approach. Most of the presented agents aim at helping the user in browsing World Wide Web, Usenet news filtering and finding experts for a given topic. The content-based agents that work on text data usually 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 the recent work in supervised learning methods for text classification 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. Most of the current work uses a bag-of-words document representation. There is some evidence that the extension of the bag-of-words representation (eg., by word position, word sequences, hypertext structure) significantly improves results in some domains. The high number of features is usually handled by using a 'stop-list' and by selecting only the best features. Comparative experiments show that the reduction of up to 90% in the number of features resulted in either an improvement or no loss in the system performance. The best performing feature selection measures are frequency, Odds ratio, χ^2-statistics, or (for problems with rather balanced class distribution) informativity. Among the best performing classification algorithms are boosted Decision Trees, Support Vector Machines and Generalized Instance Sets closely followed by k-Nearest Neighbor, Neural Networks and Linear Least Square Fit, but additional comparison of classification algorithms on more data sets under the same experimental conditions is still needed.

Some interesting research questions for future work include the study of scalability to domains having very short or very long documents. Some experiments on very short document (containing hyperlink descriptions) are described in [1, 65]. Other interesting research questions [70] 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.

In our Personal WebWatcher project there is on-going research on different aspects of using text-learning for better Web browsing including richer document representation using dynamically constructed background knowledge, and making word occurrence prediction based on very short documents [1]. An important direction of text-learning is using information extraction specialized for different domains, such as finding paper citations in the research papers available on the Web (`CiteSeer` [13]) or building a knowledge database from the Web (`WebKB` [71]).

The current trends of using machine learning for intelligent agents include a combination of text processing with speech recognition and content processing of image/video [72].

Acknowledgements

This work was financially supported by the Slovenian Ministry for Science and Technology. Part of this work was performed during the author's stay at Carnegie Mellon University in Tom Mitchell's group. Many thanks to the anonymous reviewers for valuable comments and suggestions. The author is grateful to Nada Lavrač for valuable comments and suggestions on the latest version of this paper.

References

[1] Mladenić, D., Machine Learning on non-homogeneous, distributed text data, *PhD thesis*, University of Ljubljana, Slovenia, October, 1998. http://www.cs.cmu.edu/~TextLearning/pww/PhD.html

[2] Mitchell, T.M. (1997). Machine Learning. The McGraw-Hill Companies, Inc..

[3] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., Advances in Knowledge Dicsovery and Data Mining, AAAI Press/MIT Press, 1996.

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

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

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] Shavlik, J., Eliassi-Rad, T., Building intelligent agents for Web-based tasks: a theory-refinement approach, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

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

[20] 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.

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

[22] 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.

[23] 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.

[24] 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.

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

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

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

[28] de Vel, O., Nesbitt, S., A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[29] 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.

[30] 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.

[31] 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.

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

[33] 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.

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

[35] Gams, M., Grobelnik, M., Intelligent agents in information society. *Proceedings of the 6th Electrotechnical and Computer Science Conference ERK'97*, pp. 125-128, Ljubljana, Slovenia: Slovenia Section IEEE, 1997.

[36] 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.

[37] Mladenić, D., Grobelnik, M., 1998. Feature selection for classification based on text hierarchy, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[38] Mladenić, D., & Grobelnik, M., Word sequences as features in text-learning. *Proceedings of the Seventh Electrotechnical and Computer Sc. Conference ERK'98*, pp.145-148, Ljubljana, Slovenia: IEEE section, 1998.

[39] Creecy, R.M., Masand, B.M., Smith, S.J., Waltz, D.L., Trading MIPS and Memory for Knowledge Ebgineering, *Communications of the ACM* Vol. 35, No.8, pp.48–64, August 1992.

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

[41] Gelfand, B., Wulfekuhler, M., Punch III, W.F., Automated Concept Extraction from Plain Text, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[42] 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.

[43] Lewis, D.,D., Ringuette, M., Comparison of two learning algorithms for text categorization, *Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94)*, 1994.

[44] Liere, R., Tadepalli, P., Active Learning with Committees: Preliminary Results in Comparing Winnow and Perceptron in Text Categorization, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[45] Moulinier, I., Ganascia, J.-G., Applying an Existing Machine Learning Algorithm to Text Categorization, In *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, (S.Wermter, E.Riloff, G.Scheler Eds.), Springer-Verlag, 1996.

[46] Nigam, K., McCallum, A., Pool-Based Active Learning for Text Classification, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[47] Slattery, S., Craven, M., Learning to Exploit Document Relationships and Structure: The Case for Relational Learning on the Web, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[48] 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.

[49] 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.

[50] Apté, C., Damerau, F., Weiss, S.M., Text Mining with Decision Rules and Decision Trees, *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98*, 1998.

[51] 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.

[52] 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.

[53] 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.

[54] Joachims, T., Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of the 10th European Conference on Machine Learning ECML98*, pp. 137-142, 1998.

[55] 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, 1997.

[56] Lam, W., Ho, C.Y., Using A Generalized Instance Set for Automatic Text Categorization, *Proc. of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pp. 81-89, 1998.

[57] Yang, Y., An evaluation of statistical approaches to text categorization, *Journal of Information Retrieval*, (to appear) 1998.

[58] Cohen W.W., Singer, Y., Context-sensitive learning metods for text categorization, *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pp.307-315, 1996.

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

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

[61] Wiener, E., Pedersen, J.O., Weigend, A.S., A Neural Network Approach to Topic Spotting, *Proc. of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95)*, 1995.

[62] 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.

[63] 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.

[64] Lewis, D.D., Schapire R.E., Callan, J.P., Ron Papka, R., Training Algorithms for Linear Text Classifiers, *Proc. of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, pp.298-306, 1996.

[65] Mladenić, D., Feature subset selection in text-learning, *Proceedings of the 10th European Conference on Machine Learning ECML98*, pp.95-100, 1998.

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

[67] 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.

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

[69] 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.

[70] Carbonell, J., Craven, M., Fienberg, S., Mitchell, T., Yang, Y., Report on the CONALD Workshop on Learning from Text and the Web, 1998.

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

[72] *Working notes of Workshop on Mixed Media Databases, Conference on Automated Learning and Discovery CONALD-98*, 1998.