
User Modeling and User-Adapted Interaction **10:** 147–180, 2000.

User Modeling for Adaptive News Access

DANIEL BILLSUS and MICHAEL J. PAZZANI
Dept. of Information and Computer Science, University of California, Irvine, CA, USA,
E-mail: dbillsus@ics.uci.edu

(Received 16 November 1999; in final form 30 June 2000)

Abstract. We present a framework for adaptive news access, based on machine learning techniques specifically designed for this task. First, we focus on the system's general functionality and system architecture. We then describe the interface and design of two deployed news agents that are part of the described architecture. While the first agent provides personalized news through a web-based interface, the second system is geared towards wireless information devices such as PDAs (personal digital assistants) and cell phones. Based on implicit and explicit user feedback, our agents use a machine learning algorithm to induce individual user models. Motivated by general shortcomings of other user modeling systems for Information Retrieval applications, as well as the specific requirements of news classification, we propose the induction of hybrid user models that consist of separate models for short-term and long-term interests. Furthermore, we illustrate how the described algorithm can be used to address an important issue that has thus far received little attention in the Information Retrieval community: a user's information need changes as a direct result of interaction with information. We empirically evaluate the system's performance based on data collected from regular system users. The goal of the evaluation is not only to understand the performance contributions of the algorithm's individual components, but also to assess the overall utility of the proposed user modeling techniques from a user perspective. Our results provide empirical evidence for the utility of the hybrid user model, and suggest that effective personalization can be achieved without requiring any extra effort from the user.

Key words: user modeling, machine learning, information retrieval, intelligent agents, recommender systems.

1. Introduction

Driven by the explosive growth of information available online, the World-Wide-Web is currently witnessing an ongoing trend towards personalized information access. As part of this trend, numerous personalized news services are emerging. For example, Internet portals such as *Yahoo, Lycos* and *Excite* offer personalized access to daily news stories from a large range of categories. These services are based on static questionnaires that users fill out in order to make use of news filtering capabilities. In this paper we argue that questionnaire-based personalization has disadvantages that can be overcome through the use of machine learning techniques for adaptive information access. Personalization based on static questionnaires is neither fine-grained enough to accurately reflect an individual user's interests,

nor flexible enough to take a user's interest changes into account. In addition, it requires additional work from the user.

The use of machine learning algorithms for user modeling purposes has recently attracted much attention. In general, the growth of the Internet has been the driving force underlying the recent surge in research in this field. The number of recent workshops on the subject (Bauer et al. 1999; Joachims et al. 1999; Rudstrom et al. 1999; Papatheodorou 1999), and sessions in major conferences (Jameson et al. 1997; Kay 1999) document this development. As the amount of information available on-line grows with astonishing speed, people feel overwhelmed navigating through today's information and media landscape. Information overload is no longer just a popular buzzword, but a daily reality for most of us. This leads to a clear demand for automated methods, commonly referred to as intelligent information agents, that locate and retrieve information with respect to users' individual preferences (Lang, 1995; Pazzani and Billsus, 1997; Balabanovic, 1998). As intelligent information agents aim to automatically adapt to individual users, the development of appropriate user modeling techniques is of central importance.

In contrast to the recent surge in research activity, the field still has a dearth of fielded applications, and the impact of learning techniques for personalized information access on the average web user has been fairly limited. We speculate that one reason for this effect is the tremendous increase in system complexity that has to be addressed as soon as a system is to be made available to a large number of users. While research prototypes of intelligent information systems rarely address issues such as efficient database management, computational complexity, scalability to large numbers of users and concurrency of parallel requests, we suspect that precisely these issues often determine the success of information systems in real-world scenarios. As a result, even the most accurate learning or user modeling algorithm is of limited practical use if it cannot easily be turned into a practical system that holds up to the constraints and requirements of real-world deployment.

The focus of this paper is the design, deployment and evaluation of a client/server-based framework for adaptive news access. At the center of this framework is the *Adaptive Information Server (AIS)*, which uses a multi-strategy machine learning algorithm designed to acquire detailed user models, based on explicit and implicit user feedback. Due to the client/server-based architecture of our system, multiple different news agents are supported by the same server. In this paper we describe two different versions of the *Daily Learner*, an agent for adaptive news access. While the first version provides personalized news through a web-based interface, the second system is geared towards wireless information devices such as PDAs and cell phones. We use Palm, Inc.'s wireless *Palm VII*™ organizer as an example of such a device.

Clearly, the need for intelligent information agents is not limited to the web, as we are currently witnessing an increasing trend towards "ubiquitous information access". Different types of wireless information devices, designed to tap into the

Internet's vast information resources without physical constraints, are currently being released into the marketplace. For example, cell phones can access Internet-based information services, and pagers can alert users of late-breaking news. While these devices undoubtedly enhance the utility of online information and are likely to open up opportunities for revolutionary information-centric applications, they are cramped by several technical constraints. First, the small size of wireless information devices leads to inherently limited user interfaces. Second, bandwidth constraints impose limits on the amount of information to be transferred. Third – and most importantly under current conditions – wireless information transmission is expensive. Service providers charge users based on the amount of data transmitted, turning wireless information access into a costly luxury compared to regular Internet access. For example, transmission costs for the *Palm VII*™ organizer amount to approximately $25 per month for 150 KB, with an extra charge of 20 cents for each additional KB. We believe that adaptive information access, based on the use of automatically acquired user models, has the potential to simplify access to relevant information, and significantly reduce the amount and cost of data transmitted.

Building a system for adaptive news access is a challenging problem for several reasons. Traditional Information Retrieval approaches are not directly applicable to this problem setting, as most IR systems assume the user has a specific, well-defined information need. In our setting, however, this is not the case. If at all, the user's query could be phrased as: "What is new in the world that I do not yet know about, but should know?". Computing satisfactory results for such a query is non-trivial. The difficulty stems from the range of topics that could interest the user, and the user's changing interest in these topics. We must also take into account that it is the novelty of a story that makes it interesting. Even though a certain topic might match a user's interests perfectly, the user will not be interested in the story if it has been heard before. Therefore, we need to build a system that acquires a model of a user's multiple interests, is flexible enough to account for rapid interest changes, and keeps track of information the user knows.

This paper is outlined as follows. First, we briefly summarize the main findings of our previous work on news classification to motivate the goals of our current work. We then describe the functionality and design of the *Adaptive Information Server (AIS)*, which is the central component of a client/server architecture for adaptive news access. Next, we introduce two clients that are part of this architecture: one provides access to adaptive news on the web, the other is geared towards wireless information devices. We then describe a multi-strategy machine learning algorithm specifically designed to acquire individual users' interests in daily news stories. We focus on two unique aspects of this algorithm. First, we motivate the induction of a hybrid user model that consists of separate models of a user's long-term and short-term interests. Second, we show how the user model keeps track of information that has already been presented to the user. We evaluate the proposed algorithm on user data collected with both system versions, and quan-

tify the performance contributions of the system's individual components separately. In an additional experiment, we assess the overall utility of our system from a user perspective by comparing its adaptive characteristics to static, non-personalized news access. Discussions of related and future work conclude the paper.

2. Previous Work

In previous work we reported on the design, architecture and algorithms of an agent intended to become part of an intelligent, IP-enabled car radio. The purpose of this system was to compile a set of news stories likely to interest the driver. Based on the driver's explicit voice feedback, the system was to construct a user model over time, allowing it to adapt its news presentation to the driver's personal interests. A detailed description of the system and underlying algorithms is available in (Billsus and Pazzani, 1999a/b). Here, we summarize the main functionality and our initial findings, as our experience with the system forms the basis for our current research.

In order to evaluate the proposed algorithms for adaptive news recommendation, we implemented a web-based prototype and made it publicly accessible for several months. Figure 1 shows the user interface of *News Dude*, the car radio prototype and predecessor to our current work. This system was implemented as a Java Applet

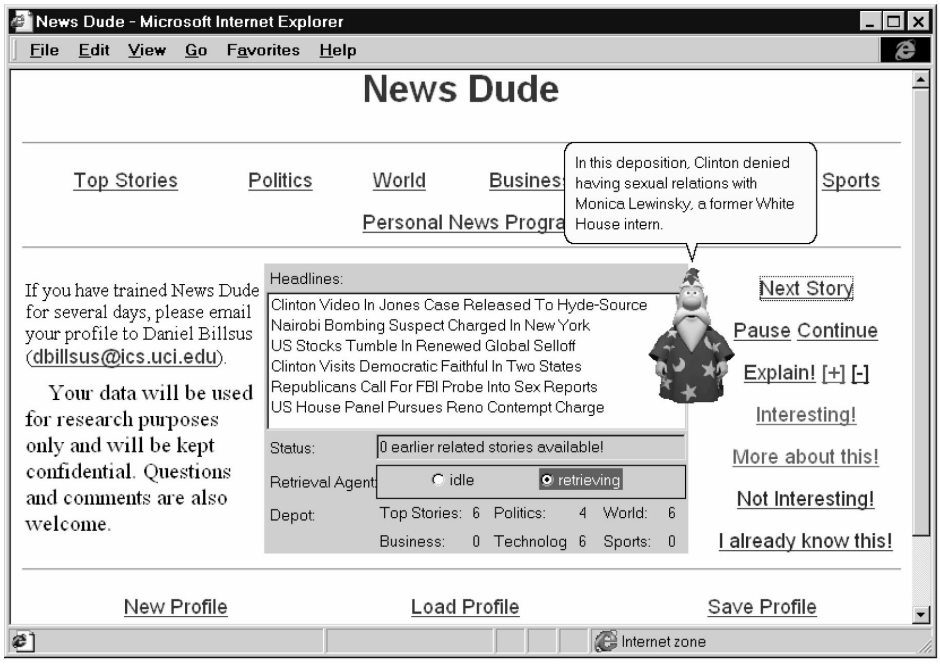

Figure 1. News Dude user interface.

that used Microsoft's *Agent* library to display an animated character that used a speech synthesizer to read news stories to the user. Although our ultimate goal was to work towards speech-driven agents that do not need graphical user interfaces, we used the web as a medium that allowed us to make the system available to a large user base for data collection and testing purposes.

The agent provided access to stories from six different news channels: Top Stories, Politics, World, Business, Technology and Sports. When the user selected a news channel, the Applet connected to a news site on the Internet (*Yahoo!News*) and started to download stories. Since the Applet was multi-threaded, download of stories continued in the background while the synthesizer was reading. The user was allowed to interrupt the synthesizer at any point and provide feedback for the story being read. One of the design goals for the system was to provide a variety of feedback options that allowed users to communicate preferences in more informative ways than through the commonly used *interesting/uninteresting* rating options. We considered an intelligent information agent to be a personal assistant that gradually learns about users' interests. In this context, it seemed natural to have more informative ways to communicate preferences. For example, a user might want to tell the agent that he or she already knows about a certain topic, or request further information related to a certain story. An additional way to provide feedback was based on the system's ability to form explanations for recommendations. Users were then allowed to critique these explanations. For example, a user could ask the system not to use a certain explanation anymore if it was considered to lead to poor recommendations. In summary, the system supported the following feedback options: *interesting, not interesting, I already know this, tell me more and explain* – combined with the option to critique explanations. After an initial training phase, the system used a special-purpose machine learning algorithm to compute a sequence of news stories ordered with respect to the user's interests. We describe the current version of this algorithm in Section 4.

Using standard performance measures from machine learning and information retrieval, we evaluated the system's predictive performance, based on data collected from 10 regular system users that provided feedback on roughly 3000 news stories. We found that the system learned to recommend interesting stories with reasonably high precision after only a very short training period. In short, the precision measured at the top 5 recommendations reached on average 75% after only 3 training sessions (here, *precision* is the fraction of stories recommended by the system that the user rated as interesting). While these results were highly encouraging, they were based on a very small set of users – despite the fact that the system had been publicly available for several months.

Looking at the system as a whole, it became clear that a good user modeling algorithm alone does not make a truly useful system. Since the system's main purpose was to serve as a car-radio prototype, it was not specifically designed to be used on the web, and thus suffered from usability problems in this context. We identified the following shortcomings:

- Listening to a speech synthesizer is an uncommon way to access news content on the Internet. Given the current state of the art of speech synthesis, most users perceive this form of news access as unacceptable.
- Due to the absence of a central news server, the system architecture led to inefficient news access. Every client had to retrieve a set of current news stories, before this set could be reordered with respect to the user's interests. Even though noticeable pauses were kept at a minimum through the multithreaded implementation of the application, this design resulted in overall degraded system responsiveness.
- The absence of a central server prohibited data collection from all system users. Since all personal profiles were stored locally on the user's computer, users that were willing to share their data for research purposes had to send in their profiles via email. In addition, this architecture has algorithmic implications, as every user profile is acquired individually, i.e. without potentially beneficial knowledge about the preferences of other users. Using knowledge about the preferences of user communities for personalization purposes is a common approach to adaptive information access known as *collaborative filtering* (Shardanand and Maes, 1995).
- Compatibility restrictions imposed by the speech synthesizer limited the system to Microsoft's *Internet Explorer*. In addition, the speech synthesizer required an initial download of more than one megabyte for first-time users, which was clearly unattractive for users connecting through slow modem lines.

The next phase of our work focused on the redesign of the existing system. Our goal was to deploy the proposed user modeling approach in a context that did not suffer from the previous limitations in order to study our algorithms with data obtained from a significantly larger user base. The architecture, algorithms and evaluation of the redesigned system form the core of this paper.

3. The Adaptive Information Server

In this section we introduce the focus of our current work: the *Adaptive Information Server* We first report on the general goals that guided the system design. We then illustrate how these goals were realized in the form of a client/server architecture for adaptive information access. Finally, we introduce two clients that are part of this architecture: one provides access to adaptive news on the web, the other is geared towards wireless information devices.

3.1. Design goals

The design goals of the proposed news delivery architecture are based on our experience with the prototype implementation discussed in Section 2. The system was designed with respect to the following guidelines:

- *Central data storage.* All user data should be stored in a central repository and not on the user's computer. This requirement facilitates data collection and allows the system to incorporate preferences of user communities into the recommendation algorithm. In addition, users do not need to explicitly load or save profiles.
- *Scalability.* The user modeling algorithm must be efficient. This means that the system must be capable of producing useful recommendations, even if a potentially large number of users request recommendations simultaneously. Even the most accurate user modeling algorithm is of limited practical use, if it cannot be deployed in a scalable system due to its prohibitive computational complexity.
- *Platform and device independence.* The system must not be inherently limited to an individual platform in order to attract a large user base. Ideally, it should not even be limited to workstations in order to support a wide range of information devices. Consequently, the system must not rely on the user's local system to perform any critical functions, i.e. the user's system should act primarily as an information display.
- *Ubiquitous access.* Access to personal profiles should not be linked to individual devices. Rather, users should be able to access the same personal profile through any workstation or any other supported Internet-enabled information device.
- *Responsiveness.* The system should be responsive at all times, i.e. waiting times should be reduced to a minimum.

While some of these requirements might appear obvious or not of academic interest, we believe that paying close attention to these guidelines is a key factor in the successful deployment and evaluation of intelligent information systems. Likewise, the effectiveness of user models can only be determined realistically in the context of a successfully deployed system. We believe that there is a fundamental difference between information systems that are being used because of their superior functionality along all dimensions and systems that are being treated as test applications in order to evaluate the effectiveness of a single idea.

3.2. System architecture

We realized the design goals listed in Section 3.1 through the design and implementation of a client/server architecture for adaptive news access. At the center of this architecture is the *Adaptive Information Server*, which handles a variety of functions ranging from downloading and storing news content to maintaining user profiles and recommending stories with respect to individual users' personal interests. Figure 2 provides a simple overview of the *AIS* server and its client applications.

AIS periodically accesses various news providers on the Internet (in the experiments reported here we use *Yahoo!News*), downloads news stories and stores them in a local relational database. Users can access these news stories through multiple client applications. When users provide feedback for news stories, *AIS* stores this

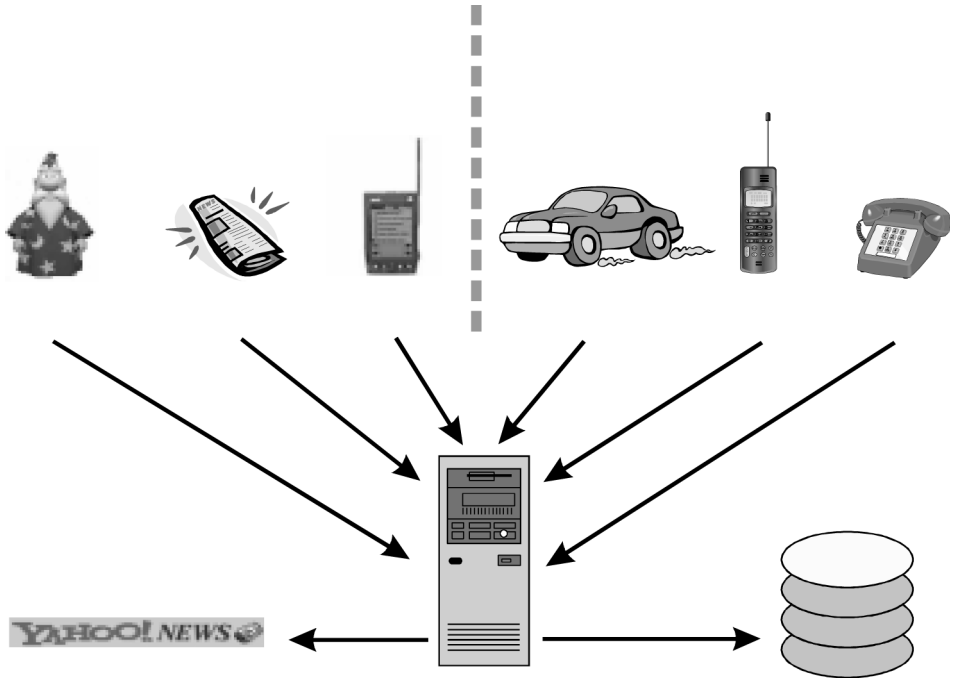

Figure 2. The Adaptive Information Server.

feedback in the same relational database, allowing it to maintain preference information of its user base in one central location. When users log into the server through a client application, *AIS* regenerates a personal user model based on previous ratings retrieved from the database, and uses this model to order current news stories with respect to the user's individual interests (the implementation caches large numbers of user ratings and news stories in memory to limit latency to a fraction of a second).

The current *News Dude* client supports the same functionality as the prototype described in Section 2. While the new architecture addresses some of the previous system's shortcomings, the user interface is still limited by the usability constraints imposed by the speech synthesizer. Thus, we do not include this version in our analysis in this paper. Instead, we focus on two new clients for the *AIS* architecture: the *Daily Learner* is a web-based adaptive news service (see Section 3.3), and the *Daily Learner Palm VII*™ Edition supports adaptive news access on Palm Inc.'s wireless *Palm VII*™ organizer (see Section 3.4). The University of California has licensed *the Adaptive Information Server to AdaptiveInfo.com*, which plans to make the system available on additional wireless information platforms. For example, *AdaptiveInfo* will extend the *Adaptive Information Server* to support Internet- enabled cell phones through the "Wireless Application Protocol" (WAP). In addition, *DaimlerChrysler* continues to explore the use of an adaptive news

client in the car radio context. We outline further directions for future work in Section 7.

3.3. THE DAILY LEARNER ON THE WWW

The *Daily Learner* is an adaptive news service, which was publicly available on the World Wide Web from May 1999 to June 2000. The system displays news stories in a web-based interface and communicates with the *Adaptive Information Server* through a Java Applet.

Figure 3 shows the Daily Learner user interface. Users can set up personal accounts, which can subsequently be used to log into the server and access personalized news. The system offers a choice of 9 different news categories (*Top Stories, Politics, World, Business, Technology, Science, Health, Entertainment* and *Sports*). As soon as a user requests a story, its full text is transmitted from the server to the client and displayed in the center of the screen. After reading a story, users can provide explicit feedback and either rate the story as *interesting* or *not interesting*, request *more* information on the story currently displayed, or let the system know that they *already know* about a certain event. It is important to note that users are never required to rate a news story, i.e. the user can decide

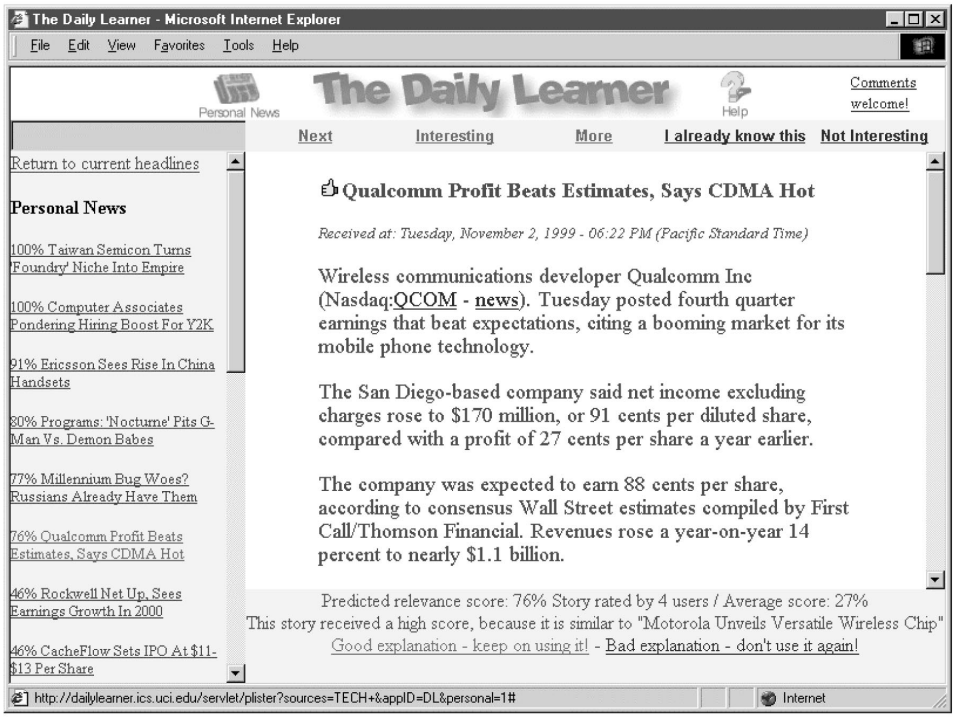

Figure 3. The Daily Learner Web Interface.

to rate stories that are clearly *interesting* or *not interesting*, but skip ratings for stories that do not clearly fall into one of these categories. In a separate section of the screen, the system displays information about its current relevance prediction, i.e. its assessment of how likely it is that the user will be interested in the story. In addition, the overall number of users that have rated the story, as well as the average rating are displayed. Furthermore, the system displays an automatically constructed explanation for its relevance prediction and allows the user to critique the formed explanation as an additional form of feedback. The analysis of the explanation function was the subject of previous work (Billsus and Pazzani, 1999a). We describe the underlying algorithms for all other feedback options in Section 4. As soon as the user has provided feedback for an initial set of stories, the system can construct a personal news program from a set of categories selected by the user. This results in a list of headlines ordered with respect to the current user model. Users can either select specific headlines from this list, or step through personal news programs sequentially with a *next* function.

3.4. THE DAILY LEARNER ON THE PALM VII

The interface design for the wireless *Daily Learner* version is based on bandwidth, transmission-cost and usability constraints imposed by portable information devices. We aim to minimize both the required interaction between user and device, as well as the amount of data transmitted between device and server. Both of these goals are in conflict with explicit story ratings: users would have to communicate explicit ratings to the device, and these ratings would then have to be transmitted to the server. In contrast, an ideal agent for wireless information access would not require any additional work from the user, and would not increase the number of slow network operations or the amount of data transmitted. Therefore, we collect preference information implicitly, simply by observing the user's news access patterns. Successful use of implicit user feedback has previously been reported in the context of learning web agents (Lieberman, 1995). The *Palm VII*™ version of the *Daily Learner* is available for public download at *http://www.palm.net*. Figure 4 shows the main menu on the *Palm VII*™.

As a first step, users can associate the unique device ID of their Palm device with an existing or newly created *Daily Learner* account. This is a one-time operation and ensures that users can access personal profiles without any explicit login. The system supports the same nine news channels as the web version. Users can tap the on-screen button of a news category, which causes the device to retrieve a first set of personalized headlines from *AIS*. An example of a headline screen is shown in Figure 5. Due to the display size, bandwidth- and transmission-cost constraints, only 4 headlines are displayed at once – additional sets of 4 can be requested at any time. Headlines are transmitted and displayed rank-ordered with respect to the user's current interest profile. This helps reduce the amount of data transmitted, because headlines that are likely to interest the user are sent to the

Figure 4. Daily Learner Main Menu on the Palm VII.

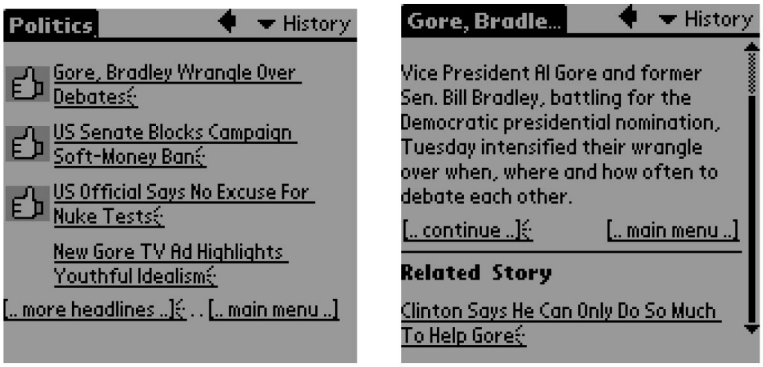

Figure 5. Daily Learner Headline and Summary Screens.

device first. Headlines may be annotated with a thumbs-up icon to indicate that the system highly recommends the story to the user. When the user taps on a headline, the first paragraph and, if available, the headline of a related story, are displayed (see Figure 5). *AIS* finds related stories automatically, based on textual similarity computations using *tf-idf* weights (Salton, 1989). From the summary screen, the

user can either return to the previous headline screen, or read the next page of the story. Here, a page refers to a few paragraphs (approximately 512 bytes of text). The remaining text of a story can be requested page by page. This helps the user save transmission costs and allows us to determine how much interest the user has in a story.

An additional function allows users to use keyword queries to search for news stories. This can save time and transmission costs for users looking for an update on a particular story. A novel aspect of this search function is that the order of the returned results is based on the user's personal interest profile. For example, the user could search for the term ''Microsoft'', which is likely to appear in multiple contexts. From the user's past access history, *AIS* might have learned that the user has an interest in Microsoft's anti-trust trial, but is not interested in its role in the presidential fund raising campaign. Although both stories might match the submitted query equally, *AIS* can give preference to a certain context, and thus reduce transmission of unwanted information. In the current implementation, *AIS* follows a simple combination strategy: it first identifies all current stories that match a query, and then rank-orders the returned results with respect to the user's interests. In future research, we will explore more sophisticated combination strategies that take into account both the degree of the query match and the predicted relevance score.

4. Learning User Models

In this section we describe the user models that our system forms and, in particular, how these user models can be acquired automatically using a special-purpose machine learning algorithm. First, we focus on the user feedback available to the algorithm. This is the key difference between the two clients described in this paper. While the web-based *Daily Learner* learns from the user's explicit feedback, the wireless *Palm VII*™ client learns from implicit feedback, i.e. just by observing the user. Following, we focus on the characteristics of news classification and describe the multi-strategy machine learning algorithm that forms the core of the *Adaptive Information Server*.

4.1. Explicit feedback

There has been a substantial amount of work on learning user preferences from text documents (Lang, 1995; Pazzani and Billsus, 1997). In these scenarios, users rate text documents with respect to their interests and assign either class labels or scores on a certain scale. Labeled documents can then be used as training examples for a learning algorithm, and the resulting hypothesis can be seen as a user model that allows classifying new documents. Here, we adopt this approach and treat a rated news story as a training example labeled by the user. In the web-based Daily Learner version, users rate news stories explicitly as *interesting*, *not interesting* or *known*. In

addition, requesting *more* information is interpreted as an alternative way to express interest in a particular story. Since we are ultimately interested in separating interesting from uninteresting stories, we map the user's explicit ratings to two classes: *interesting* and *uninteresting*. Intuitively, the *interesting* class label is assigned to stories rated as *interesting* as well as to stories for which the user requested more information. The *uninteresting* class is only assigned to stories rated as *not interesting*. It is important to note that a story rated as *known* is not labeled as *uninteresting*. Rather, known stories remain unlabeled. Their purpose is to prevent the system from recommending very similar stories multiple times (see Section 4.3.1).

Internally, the system assigns a score ranging from 0 to 1 to every story labeled as *interesting* or *uninteresting*. This allows for a more fine-grained distinction between ratings. For example, we assign a score of 0.9 to stories the user rated as *interesting*, and a score of 1.0 if the user requested *more* information. These constants reflect our intuition that a user's request to read more is a stronger indicator for the story's appeal than a rating as *interesting* (however, the exact values were assigned ad-hoc). The conversion to a numeric scale also allows us to incorporate additional information into the scoring process. For example, the *News Dude* client uses *time-coded-feedback*, taking advantage of the fact that users tend to listen to interesting stories for a longer time than to stories considered uninteresting. In previous work we showed that this approach can increase predictive accuracy significantly (Billsus and Pazzani, 1999a). Likewise, a fine-grained distinction between different levels of ratings is important for the implicit feedback approach described in Section 4.2.

4.2. Implicit feedback

Due to the usability and bandwidth constraints described in Section 3.4, the *Palm VII*™ Daily Learner version learns exclusively from implicit feedback, i.e. just by observing the user's browsing behavior. The described interface, combined with the current pricing scheme of Internet access on the *Palm VII*™, lends itself to capturing preference information without requiring any additional work from the user. Since both Daily Learner versions draw on the same user modeling algorithm, both systems use the same class labels and scoring scale. In general, we assign both a class label and a score to every displayed headline. As soon as the user taps on a headline and requests the first paragraph of a story, we label the story as *interesting*. In order to make use of all the information available and to allow for a fine-grained rating scheme, we use the following technique to derive a corresponding score. Initially, we set the score for a selected story to 0.8, and increase this score as the user requests page by page of the story's body (Figure 5 shows how the user can request the next page of a story). A rating of 1.0 corresponds to a story that the user downloaded completely. We believe that this is a reasonable heuristic to collect preference information. Since users are charged for all transmissions, we expect them to only select headlines they find interesting.

Likewise, we assume that the proportion of a story for which a user is willing to pay is positively correlated with the user's interest in the story. In contrast, we interpret skipping a story as negative feedback and assign the *uninteresting* class label. However, instead of using a constant score for all skipped stories, we take the system's prediction for the story into account and determine an implicit score by subtracting a constant from the system's prediction. Intuitively, we can assign low implicit scores more confidently to skipped stories that already received a low prediction than to stories that were thought to be of high interest to the user.

If a user does not select any story, we do not generate any implicit ratings at all, because in this case the user does not express any preference relation of one story over another. Furthermore, it is unlikely that the lack of any ratings can only be attributed to the user's lack of interest, because external factors such as loss of connection to the server in areas with weak signal strength can cause the same effect. All implicitly labeled stories are entered into the user model immediately, allowing AIS to adjust to the user's interests not only during subsequent sessions, but also during the current session as more headline pages are requested.

4.3. A HYBRID USER MODEL FOR NEWS STORY CLASSIFICATION

The specific design of the agent's user model is motivated by a number of observations and requirements. First, the model must be capable of representing a user's multiple interests in different topics. Second, the model must be flexible enough to adapt to a user's changing interests reasonably quickly, even after a long preceding training period. Third, the model should take into account that a user's information needs change as a direct result of interaction with information (Belkin, 1997). Surprisingly, this aspect has received little attention in the IR community. For our application, we take into account the stories the user has seen before, to avoid presenting the same information multiple times.

The above requirements led to the development of a multi-strategy learning approach that learns two separate user-models: one represents the user's short-term interests, the other represents the user's long-term interests. Distinguishing between short-term and long-term models has several desirable qualities in domains with temporal characteristics (Chiu and Webb, 1998). Learning a short-term model from only the most recent observations may lead to user models that can adjust more rapidly to the user's changing interests. Here, we restrict the short-term model to the n most recently rated stories (set to 100 in the current version). The need for two separate models can be further substantiated by the specific task at hand, i.e. classifying news stories. Users typically want to track different ''threads'' of ongoing recent events – a task that requires short-term information about recent events. For example, if a user has indicated interest in a story about a current Space Shuttle mission, the system should be able to identify follow-up stories and present them to the user during the following days. In addition, users have general news

preferences, and modeling these general preferences may prove useful for deciding if a new story, which is not related to a recent rated event, would interest the user. With respect to the Space Shuttle example, we can identify some of the characteristic terminology used in the story and interpret it as evidence for the user's general interest in technology and science related stories. While the distinction between a long-term and a short-term model might be particularly useful for news story classification, it is likely that this principle can be applied to a large range of intelligent information access applications.

For the following description of the user model and its automated induction, we assume that the system has access to a set of labeled news stories, obtained through either explicit or implicit user feedback. First, we present the two components of the user model individually. Following, we illustrate the hybrid algorithm that combines both models.

4.3.1. *Modeling Short-Term Interests with the Nearest Neighbor Algorithm*

The purpose of the short-term model is two-fold. First, it should contain information about recently rated events, so that stories which belong to the same threads of events can be identified. Second, it should allow for identification of stories that the user already knows. A natural choice to achieve the desired functionality is the nearest neighbor algorithm (NN). The NN algorithm simply stores all its training examples, in our case labeled news stories, in memory. In order to classify a new, unlabeled instance, the algorithm compares it to all stored instances given some defined similarity measure, and determines the "nearest neighbor" or the k nearest neighbors. The class label assigned to the new instance can then be derived from the class labels of the nearest neighbors. The utility of the NN algorithm has previously been explored in other text classification applications (Cohen and Hirsh, 1998; Yang, 1998; Allan et al. 1998).

To apply the algorithm to natural language text, we must define a similarity measure that quantifies the similarity between two text documents. This is a well-studied problem in Information Retrieval, and we rely on a commonly used document representation and associated similarity measure. We convert news stories to *tf-idf* vectors (term-frequency/inverse-document-frequency), and use the cosine similarity measure to quantify the similarity of two vectors (Salton, 1989).

Each rated story is converted to its *tf-idf* representation and stored in the user model. A score prediction for a new story is then computed as follows. All stories that are closer than a threshold *t_min* to the story to be classified become voting stories. The predicted score is then computed as the weighted average over all the voting stories' scores, where the weight is the similarity between a voting story and the new story. If one of the voters is closer than threshold *t_max* to the new story, the story is labeled as known, and its computed score is multiplied by a factor $k \ll 1.0$, because the system assumes that the user is already aware

of the event reported in the story. If a story does not have any voters, the story cannot be classified by the short-term model at all, and is passed on to the long-term model (see Section 4.3.2).

The nearest neighbor-based short-term model satisfies our requirements that a user model be able to represent a user's multiple interests, and it can quickly adapt to a user's novel interests. The main advantage of the nearest-neighbor approach is that only a single story of a new topic is needed to allow the algorithm to identify future follow-up stories from the same story thread. In contrast, most other learning algorithms would require a large number of training examples to identify a strong pattern.

4.3.2. *Modeling Long-Term Interests with a Naïve Bayesian Classifier*

The long-term model is intended to model a user's general preferences. Since most of the words appearing in news stories are not useful for this purpose, the *Adaptive Information Server* periodically selects an appropriate vocabulary for each individual news category from a large sample of stories. After feature selection, the same set of features is used for all users. The goal of the feature selection process is to select *informative* words that reoccur over a long period of time. In this context, an informative word is one that distinguishes documents from one another, and can thus serve as a good topic indicator. With respect to individual documents, *tf-idf* weights (Salton, 1989) can be interpreted as a measure of the amount of information that an individual word contributes to the overall content of a document. In order to determine the n most informative words for each document, we sort words with respect to their *tf-idf* values and select the n highest-scoring words (where n is currently set to 10). We assume a word to be useful for the long-term model if it frequently appears in top n lists over a large set of documents from one category (currently we use 10,000 news stories per category for feature selection). Our feature selection approach sorts all words that appear in the overall vocabulary with respect to the number of times they appear in top n lists. Finally, the k most frequent words are selected (currently set to 150). This approach performs well at selecting the desired vocabulary: it selects words that reoccur frequently throughout one news category, but are still informative as measured by their *tf-idf* weights. Table 1 shows the set of the 50 most informative words selected with the described strategy from a collection of 10,000 news stories from the ''Technology'' and ''Science'' categories. The numbers in parentheses indicate the number of times that each word was included in a story's list of the 10 most informative words.

Even though feature selection is a well-studied topic in machine learning research, and most of the approaches reported in the literature are directly applicable to our problem setting, we decided to use the described special-purpose algorithm for several additional reasons. First, performing feature selection from a classification perspective, i.e. using selection criteria such as expected information gain (Quinlan, 1986), is computationally much more expensive than the described method. This

Table 1. Long-Term features for "Technology" and "Science" selected in October 1999

Technology:	million (126) quarter (125) cents (102) shares (100) software (87) network (84) computer (72) chip (68) wireless (65) microsoft (64) company (61) filed (59) internet (56) problems (54) percent (52) cable (52) stock (50) venture (47) sales (47) service (47) intel (46) sprint (46) deal (46) earnings (45) ibm (45) system (44) worldcom (43) bugs (41) telecom (41) mobile (41) bank (40) motorola (40) bell (40) game (40) business (39) distance (39) mci (39) fcc (38) satellite (37) merger (37) billion (37) phone (37) ipos (36) semiconductor (36) data (33) price (33) suns (33) offer (33) telekom (32) analysts (32)
Science:	drug (94) cancer (72) space (58) cells (53) patients (51) women (45) crops (39) gene (38) launched (37) disease (36) food (35) virus (34) rocket (33) city (33) mission (32) bacteria (32) infection (32) children (31) heart (30) hiv (30) satellite (29) eclipse (28) blood (28) genetic (28) suns (28) winds (27) trial (27) mice (27) orbit (27) antibiotics (26) vaccine (26) resistance (25) russian (25) human (25) aides (25) storm (25) percent (24) brain (24) fda (24) cdc (23) mosquitoes (23) energy (23) test (23) damage (22) hurricane (22) computer (21) baby (21) government (21) hospital (21) texas (21)

is primarily because such an approach would have to be performed on a per user basis. With scalability being one of the major criteria guiding our algorithm design, we select just one set of features per category and use it for all users. In addition, our previous work on text classification and information theoretic feature selection (Pazzani and Billsus, 1997) suggested that a large number of training examples is needed to select feature sets that lead to accurate text classifiers. In contrast, the method proposed here does not depend on the number of rated stories. Finally, we use the selected features as part of the system's explanation component. Since the use of large story sets per category leads to the selection of words that frequently occur in a specific news category, we can construct category-specific explanations for story recommendations. The explanation construction approach is described in (Billsus and Pazzani, 1999a).

The long-term model uses a probabilistic learning algorithm, a naÿve Bayesian classifier (Duda and Hart, 1973), to assess the probability of stories being interesting, given that they contain a specific subset of features. Each story is represented as feature-value pairs, where features are the words from the selected feature set that appear in the story, and feature values are the corresponding word frequencies. In order to take advantage of the word frequency information, the system uses the multinomial formulation of naïve Bayes (McCallum and Nigam, 1998). Making the ''naïve'' assumption that features are independent given the class label (*interesting* vs. *not interesting*), the probability of a story belonging to class j given its feature values, $p(class_j|f_1, f_2 \ldots f_n)$ is proportional to:

$$p(class_j) \prod_i^n p(f_i|class_j)^{N_i}$$

where N_i is the frequency of feature f_i and $p(class_j)$ and $p(f_i|class_j)$ can be easily estimated from training data. We use Laplace smoothing to prevent zero

probabilities for infrequently occurring words. A news story to be classified can thus be labeled with its probability of belonging to the *interesting* class.

To prevent the long-term model from classifying items that do not contain sufficient evidence to allow for accurate predictions, we introduce two constraints that determine whether the long-term model should be used to classify an item. First, *AIS* only considers a feature as informative if it appears at least *i* times in the training data. In the current implementation and in all experiments reported in Section 5, *i* is set to 10. Second, the long-term model requires an item to contain at least f_min informative features for which $p(f|interesting) > p(f|\neg interesting)$ in order to allow a classification as interesting, and likewise, at least *f_min* informative features for which $p(f|interesting) > p(f|\neg interesting)$ in order to allow a classification as not interesting. In the current implementation *f_min* is set to 2, which means an item must contain at least two informative features that are indicators for the same class (these constants were determined empirically, using a tuning set of collected data).

4.3.3. *A Multi-Strategy Learning Approach*

We predict relevance scores for stories by incorporating the short-term and long-term models into one unifying algorithm. A previously unseen news story, *u*, is classified as follows:

If $\exists$ *d:* $d \in \{short\text{-}term\text{-}stories\} \wedge cosine\text{-}similarity(d, u) > t\_min$
{
  $score = nearest\text{-}neighbor\text{-}prediction(u, \{short\text{-}term\text{-}stories\})$
  If $\exists$ *n:*$\in$ ($\{short\text{-}term\text{-}stories\} \wedge cosine\text{-}similarity(d, n) > t\_max$
    $score = score * k$, *where* $k \ll 1.0$
}
Else
  If $\exists \{f_{u1}, f_{u2}, \ldots, f_{un}\}$: $\forall f \in \{f_{u1}, f_{u2}, \ldots, f_{un}\}\ p(f|c) > p(f|\neg c)$
    $score = naïve\text{-}Bayes\text{-}prediction(u, \{all\ stories\})$
Else
    $score = default$

In summary, the approach uses the short-term model first, because it is based on the most recent observations only, allows the user to track news threads that have previously been rated, and can label stories as already known. If a story cannot be classified with the short-term model, the long-term model is used. If the long-term model decides that the story does not contain sufficient evidence to be classified, a default score is assigned. In the reported experiments, we set the default score to 0.3, so that stories that cannot be reliably classified do not appear too high in the recommendation queue, but still receive a higher score than stories that are classified as not interesting. In a machine learning context, this idea is similar to *active learning*. Stories that cannot be classified with high confidence are likely

to be presented to the user, so that areas that are only sparsely populated with training examples can be filled quickly.

5. Evaluation

In this section we report on a set of experiments designed to quantify and understand the predictive performance of our hybrid user modeling algorithm. First, we introduce the test methodology and performance measures used. Next, we apply these measures to data collected with the *Adaptive Information Server*, assess the achieved overall performance, and analyze performance contributions of individual algorithmic aspects. We provide empirical evidence for the utility of the hybrid user model by showing that the hybrid model performs better than the short-term and long-term models individually. In addition, we present results that demonstrate the utility of explicitly representing the user's knowledge. Finally, we evaluate the system's performance from a user perspective: here, the goal is not only to quantify predictive performance, but also to assess the utility of adaptive news presentation in general, by comparing user-adaptive news access to news presentation in static order.

5.1. Methodologies and performance measures

The most commonly used performance measure in work on classification learning is *classification accuracy*, i.e. the proportion of correctly classified instances. We decided not to adopt this measure, because we believe that it does not reflect the system's performance appropriately. While we could easily argue, for example, that the accuracy of the wireless *Daily Learner* version is higher than 80% after using the system for only a few training sessions, it does not say much about the algorithm's ability to recommend interesting stories with high precision. Due to the high ratio of *uninteresting* (skipped) to *interesting* (selected) stories, a classifier that classifies every story as *uninteresting* can easily achieve a higher classification accuracy than a classifier that successfully locates a few interesting stories. Since the latter classifier is more desirable from a user perspective, we do not report the algorithm's classification accuracy. Instead, we adopt common information retrieval performance measures (*precision* and *recall*) to evaluate the system, because these measures directly reflect a system's ability to locate information, and do not suffer from the aforementioned limitation. In the context of the *Adaptive Information Server*, precision is the percentage of items classified as interesting that are interesting, and recall is the percentage of interesting items that were classified as interesting.

It is important to evaluate precision and recall in conjunction, because it is easy to optimize either one separately. A classifier is most useful for recommendation purposes if it can locate many interesting items with high precision. In order to quantify this quality with a single measure, Lewis and Gale (1994) proposed the *F-measure*, a

weighted combination of precision and recall that produces scores ranging from 0 to 1. In the evaluation reported here, we assign equal importance to precision and recall, resulting in the following definition for F_1:

$$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

While the F_1 measure is an informative way to compare the relative performance of two algorithms, its interpretation is not intuitive. A measure that has a more intuitive interpretation is precision at the top N recommendations, i.e. the percentage of correct predictions among the system's top N recommendations (where "correct" is with respect to explicitly rated stories for the web version, and implicitly rated stories for the wireless version). Here, we report precision at the top four recommendations. Since the wireless *Daily Learner* client displays four news stories on one page, this measure has an intuitive interpretation in this context: it indicates the percentage of correct predictions on the first personalized news page.

In order to corroborate the reported empirical measurements, we test results for statistical significance. In the following sections, *statistical significance* refers to a single-tailed, paired-differences t-test. We consider results *statistically significant* if they pass the t-test at the 5% level ($p <= 0.05$). The t-test has recently received much criticism in the machine learning community, primarily due to its elevated Type I error (Dietterich, 1998). The elevated Type I error in much of the reported work on classification learning is due to overlapping training and test sets in test methodologies involving multiple training and test splits, such as cross-validation. Due to the sequential nature of the Daily Learner data, the experiments are not based on multiple training or test splits per user; this would not have realistically modeled the system's behavior. Therefore, the t-test is a valid method to assess the statistical significance of the reported results.

The results reported in the following sections are based on the *Adaptive Information Server* user base as of February 2000, consisting of roughly 3,000 registered users. While the large number of users is clearly helpful in order to derive meaningful results, it also introduces new challenges. Many users that log into the system might be more interested in the system's approach to adaptive news than actual news content. Therefore, it is likely that data obtained from these users is not an accurate reflection of individual interests, but merely the result of curiosity-driven system exploration. Therefore, we only include users that logged into the system on at least three different days – we assume that these users had an actual interest in using the system as a news source.

Evaluating the performance of the *Adaptive Information Server* and the *Daily Learner* is challenging for several reasons. First, standard evaluation methodologies commonly used in the machine learning literature, for example n-fold cross-validation, are not applicable to this scenario. This is mainly due to the chronological order of the training examples, which cannot be presented to the learning algorithm in random order without skewing results. Second, changes of the system's

daily predictive performance are not only caused by the effects of the system's updated user models, but are also affected by the changing topics of current news stories. Third, the system is trying to approximate models of its users' interests, and these interests are neither static nor consistent. A user going through the same list of stories at a later time might assign different labels. Although these factors clearly affect the accuracy of an overall assessment of the algorithm's utility, it is still possible to gain valuable insight from the collected data. Specifically, it is possible to measure performance contributions of the algorithm's individual components to better understand their relative strengths and weaknesses.

We evaluated the hybrid learning algorithm in two consecutive steps. First, we used a subset of the *Daily Learner* data as a *tuning set* to analyze the performance characteristics of the long-term and short-term models individually, and to find parameter settings that maximize the performance of the hybrid model. Next, we used these parameter settings on a *test set* that did not overlap with the *tuning set* to compare the hybrid model's performance to the performance of the individual models. Here, we report final results measured on the *test set* with respect to two different experimental methodologies. The first methodology measures the system's performance on individual days. For each day, all collected user ratings are ordered chronologically, and the resulting stream of feedback is used to simulate the system's recommendation algorithm. For every rated story, the system first predicts a class label and relevance score, and then records the difference between the user's actual rating and the prediction. Subsequently, the user's rating is entered into the user model. This process is repeated over a sequence of days, and the daily results are averaged to form final performance scores. In contrast, the second methodology quantifies the system's predictive performance as a function of the amount of training data provided. The analysis groups the collected data for each individual user in terms of *training sessions.* Here, a *training session* refers to a session during which a user logged into the server, accessed items and provided ratings. Only training sessions containing more than 4 ratings, with at least one positive and one negative rating are included in the experiment. We quantify the system's performance as a function of the number of training sessions. For each user, the algorithm is trained with all rated examples from the first session. Then, predictions for items from the user's following training sessions are compared to the user's actual ratings. This process is repeated by incrementing the training data session by session. Finally, the results are averaged over all users included in the experiment.

5.2. Performance of the hybrid model

In this section we present results that quantify the utility of the hybrid user model, and we show that the hybrid model outperforms its individual components. Table 2 summarizes the results with respect to the first methodology (averaged over a test set containing one week of *Daily Learner* data).

As can be expected, the results for the wireless client and the web client differ significantly. However, the much higher performance scores of the web client should not be interpreted as an indicator for the web client's superiority. Rather, the difference is primarily an effect of the data collection process. The web version is based on explicit feedback, and only stories rated by the user take part in the evaluation. In contrast, the wireless client implicitly labels all displayed items, and only items that the user selects are considered interesting. As a result, the ratio of items labeled as interesting differs significantly between both versions (approximately 58% for the web version vs. 18.3% for the wireless version), which explains the observed performance difference. However, the general outcome of the experimental comparison is similar for both system versions: the hybrid model outperforms both individual models with respect to all reported performance measures (the differences are statistically significant at $p <= 0.05$).

In Figure 6 we use the F_1 measure to characterize the system's performance as a function of the number of training sessions. The learning curves are plotted over three training sessions for the web version and ten training sessions for the wireless version. The number of reported training sessions differs for two reasons. First, users

Table 2. The hybrid model vs. short-term and long-term models

		Web Client		
	Precision Top 4	Precision	Recall	F_1
Hybrid	73.15	71.72	55.55	62.61
Short-Term	72.51	68.17	45.12	54.3
Long-Term	68.76	66.82	44.23	53.22
		Wireless Client		
	Precision Top 4	Precision	Recall	F_1
Hybrid	32.67	32.42	29.26	30.75
Short-Term	30.33	29.22	24.65	26.74
Long-Term	25.82	26.22	22.87	24.43

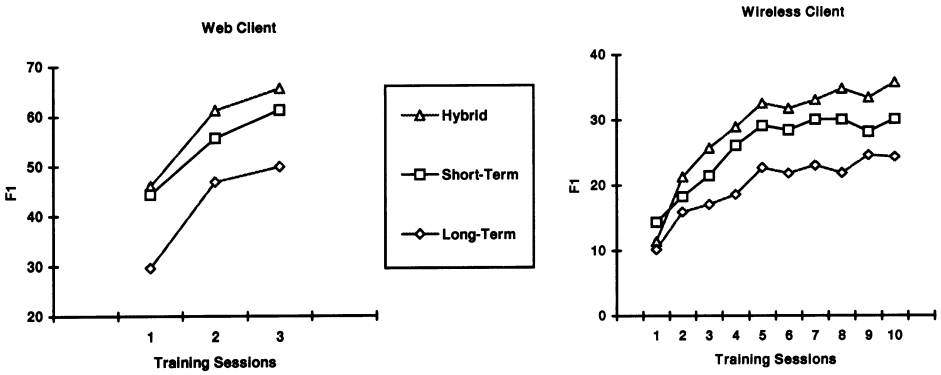

Figure 6. Learning Curves.

rate a larger number of stories per training session on the web version, which means that the user models reach their maximum size faster. When the maximum size is reached, the system's predictive performance plateaus and starts to fluctuate as a result of changing distributions of daily news stories. Second, the wireless version has a larger number of regularly returning users, which allows for averaging over a large number of users who used the system on more than 10 days. The wireless results are averaged over 185 users, which would not have been possible for ten sessions for the web version (instead, the web results are based on 150 users who used the system on more than three days). In addition to scores for the hybrid model, both figures also show the individual performance of the short-term and long-term model.

As expected, the hybrid algorithm performs better than each individual approach with respect to the F_1 measure. The key to this result lies in the sequential application of the individual models. The short-term model filters out stories that can be classified with high precision. Following, the long-term model only considers the remaining set of stories and helps locate additional relevant items. As a result, the hybrid user model has higher F_1 values than each individual model. In summary, these results suggest that the hybrid combination of the short-term and long-term model outperforms each individual model. The sequential combination of the two models allows for taking advantage of the short-term model's accelerated learning rate, while retaining the long-term model's ability to prioritize examples based on the user's general preferences.

5.3. Utility of knowledge-dependent classification

The experiments reported in this section quantify the utility of knowledge- dependent classification, i.e. explicitly representing items previously presented to the user. Table 3 compares the performance of two approaches, here labeled as *knowledge-dependent* and *knowledge-independent* classification. The knowledge-dependent approach can classify stories as *known* if they are assumed to be known by the user. In contrast, the knowledge-independent approach does not take into account that the user might already know about certain events. Knowledge-

Table 3. Effect of knowledge-dependent classification

	Web Client			
	Precision Top 4	Precision	Recall	F_1
Knowledge Independent	71.66	68.26	55.92	61.47
Knowledge Dependent	73.15	71.72	55.55	62.61
	Wireless Client			
	Precision Top 4	Precision	Recall	F_1
Knowledge Independent	30.31	29.35	30.31	29.82
Knowledge Dependent	32.67	32.42	29.26	30.75

dependent classification reduces the relevance scores for stories that are classified as *known*, and therefore leads to higher precision, but lower recall (because fewer stories are classified as interesting). Since the overall increase in precision is larger than the decrease in recall, the F_1 statistic increases overall.

Arguably, the differences reported here are small and raise the question whether knowledge-dependent classification leads to noticeable improvements. However, taking into account the goal of knowledge-dependent classification, it is not surprising that the differences observed by using the above methodology is small. Knowledge-dependent classification aims to prevent recommending content that the user already knows. As a result, items classified as *known* typically appear near the bottom of a recommendation list, and are therefore frequently not displayed at all. The main benefit of knowledge-dependent classification lies in not displaying items that are likely to be known. Since the methodology used above is based on ratings for stories that were presented to the user, it does not capture the full effect of this technique.

An additional experiment helps to further clarify the utility of knowledge-dependent classification. Figure 7 plots the probability that a user will select a story as a function of the story's similarity to the nearest *interesting* instance in the user model. The access probability increases with increasing proximity to stories that users have previously indicated as interesting. However, the probability drops as stories become very similar to previously accessed content. As shown in Figure 7, this happens at a cosine of approximately 0.7. In particular, stories with similarities that exceed 0.7 are less likely to be selected than stories that exceed 0.2. Knowledge-dependent classification takes this effect into account by preventing stories beyond the 0.7 similarity threshold from being recommended.

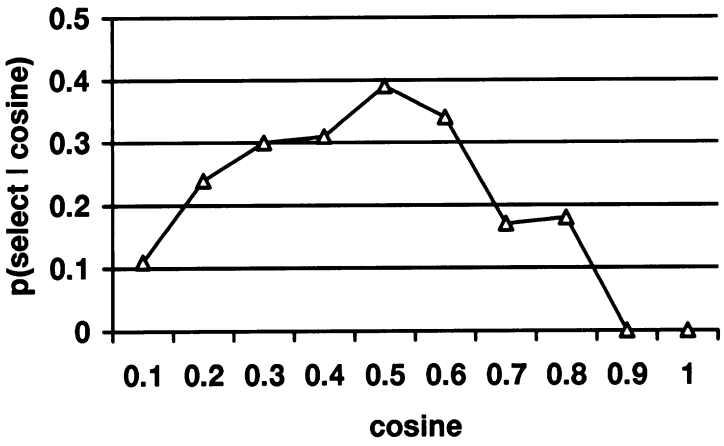

Figure 7. *p*(select | cosine of closest positive example).

5.4. Evaluation from a user perspective

In this section, we evaluate the *Adaptive Information Server*'s recommendation performance from a user perspective. In the preceding sections, we presented empirical results that focused on performance contributions of individual algorithmic aspects, and we showed that the hybrid user model that combines a short-term and long-term model performs better than each individual model alone. However, these results are not sufficient to conclude that the resulting system is indeed useful from a user perspective, leading users to adopt it on a regular basis.

The *Adaptive Information Server*'s goal is to reorder items with respect to users' individual interests. The main intuition is that such a modified order helps users access relevant content. However, information is rarely presented in random order. For example, editors prioritize news stories based on human judgement, which means that, in this case, users access content in an order deemed appropriate by human professionals. While such an order is static in the sense that it is the same for every user, it is possible that it is sufficient for most users to easily access relevant content. In this section, we present results from two studies that compare the personalized information access provided by the *Adaptive Information Server* to static information access. These results show that the system's user modeling algorithm generates an adaptive order that has two closely related effects: it simplifies locating relevant content, and leads to an overall increase in accessed information. The main idea underlying both experiments is to present items either in static or adaptive order, so that resulting differences in users' selection and browsing behavior can be quantified. Since it is essential for these experiments that users be unaware of the system's current display strategy, these experiments are limited to the wireless *Daily Learner* version. Since the web-based *Daily Learner* client displays explanations for recommendations (see Figure 3), this methodology cannot be applied to the web version. Furthermore, applying a similar methodology to the web-based version is problematic due to the small number of system users. Comparing static and adaptive news access requires disabling all personalization capabilities when the system is in static mode, which clearly reduces the system's utility. Therefore, the experiments were conducted over a short period of time in order to reduce potential user dissatisfaction: two weeks for experiment no. 1, and four weeks for experiment no. 2. While it was possible to derive meaningful results for the wireless version during these weeks, the web-based version would have required data collection over a much longer period of time.

The system's ultimate goal is to simplify access to interesting content. A simple and informative measure that quantifies progress towards this goal is the average display rank of selected stories. If the system successfully learned to order items with respect to users' individual interests, this would, on average, result in interesting stories moving toward the top of users' personalized lists of items. Therefore, the average display rank quantifies the system's ability to recommend interesting items. Since this measure does not depend on a predicted numeric score or a class label,

it is possible to apply it to static information access, allowing for a comparison of both strategies.

The "alternating sessions experiment" quantifies the difference between static and adaptive information access by randomly determining whether a user receives content in static or adaptive order. During a period of two weeks in October 1999 the *Adaptive Information Server* used its user modeling approach for approximately half of the users, while the other half received news stories in static order determined by an editor at the news source (*Yahoo! News*). On odd days, users with odd account registration numbers received news in personalized order and even users received a static order. On even days, this policy was reversed. To quantify the difference between the two approaches, we measure the mean rank of all selected stories for the personalized and static operating modes. Since a difference between static and adaptive access can only be determined for users that previously retrieved several items, we restrict the analysis to users with a minimum of five selected stories. Comparing both access modes for this subset of users revealed a significant difference. The average display rank of selected stories was 6.7 in the static mode, and 4.2 in the adaptive mode (based on 50 users that selected 340 stories out of 1882 headlines). The practical implications of this difference become apparent by analyzing the distribution of selected stories over separate headline screens (every screen contains 4 stories). Figure 8 illustrates these two distributions visually. In the static mode, 68.7% of the selected stories were on the top two headline screens, while this was true for 86.7% of the stories in the personalized mode. It is reasonable to argue that this difference makes a noticeable difference when working with handheld devices, and we interpret this result as promising evidence for the utility of adaptive news access. In addition, this result suggests that effective personalization can be achieved without requiring any extra effort from the user.

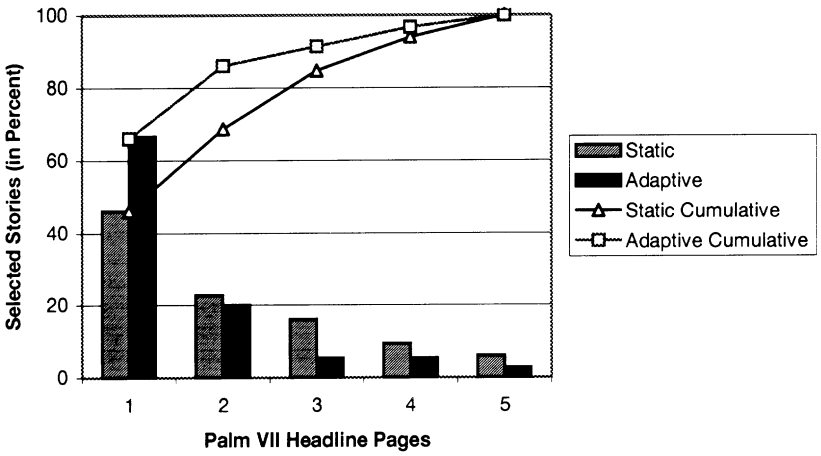

Figure 8. Distribution of Selected Stories/Alternating Sessions Experiment.

While the reported results are promising, the described experiment suffers from several shortcomings. First, the results are based on a small data set, consisting of only 50 users who selected 340 stories. Since 3Com's *Palm VII*™ device had only been publicly available for a short time when the experiment was conducted, the number of regular *Palm VII*™ users was limited. Second, due to the high cost of information access on the *Palm VII*™, users typically only select a small number of headline screens in each session. It is likely that users select from these few screens the stories that interest them most, and that this is true for both the static and adaptive access modes. Therefore, a drawback of the ''alternating sessions experiment'' is that users might not see stories they would have seen in the adaptive mode. Likewise, in the adaptive mode, users might not see stories they would have seen in static mode. The following experiment addresses this problem by displaying both adaptive and static stories on the same screen.

The ''alternating stories experiment'' is similar in principle to the ''alternating sessions experiment'', i.e. it is designed to quantify the difference between static and adaptive information access. However, the ''alternating stories experiment'' displays stories selected with respect to both the adaptive and static strategies on the same screen. Since the wireless *Daily Learner* client displays four stories on each screen, every screen contains two adaptive stories and two static stories. The server determines randomly if the first displayed story is a static or adaptive story, and the remaining stories are selected by alternating between the two strategies. The ''alternating stories'' methodology has two advantages. First, the system still adapts to the users' interests, because every screen contains two stories that were selected adaptively. This results in a change of system behavior that is much more subtle from a user perspective than the resulting change of the ''alternating sessions experiment''. Therefore, it is possible to run the experiment over a longer period of time, because users still receive a useful service. Second, users see the current top-ranked adaptive and static stories on the same screen, allowing for a direct comparison between the two selection strategies. If the system learns to adjust to users' individual interests, users can be expected to select more adaptive stories when presented with a choice between adaptive and static content.

The Adaptive Information Server used the ''alternating stories'' methodology during a period of four weeks, from February to March 2000, to collect access data for 5000 adaptive stories and 5000 static stories that were shown to users who had previously selected a minimum of 5 stories. Using these criteria, data obtained from 222 different users were included in the experiment. Similar to the ''alternating sessions experiment'' the average display rank can be used to quantify the difference between the two display strategies. However, using the ''alternating stories'' methodology, the difference between the two average display ranks was not as pronounced as in the ''alternating sessions'' experiment: 5.8 for the static mode vs. 5.27 for the adaptive mode. Likewise, the distributions of selected stories over *Palm VII*™ headline pages revealed only a small difference between the two display modes: for the static mode, 75.57% of the selected stories were on the top two head-

line screens, while this was true for 80.44% of the stories in the adaptive mode. We attribute the smaller difference between the two modes mainly to the presence of adaptive stories on every page. As a result, the user's information need might be satisfied after seeing only a small number of headline pages. If users do not have to request multiple screens to find relevant information, the observable difference in display ranks is reduced. However, this explanation only holds if users indeed select more adaptive stories than static stories. The percentage of selected stories for the two display modes clearly indicates that users are more likely to select adaptive stories than static stories. In particular, users selected 13.26% of all displayed static stories (663 stories), vs. 19.02% (951 stories) of all displayed adaptive stories, which amounts to a 43.44% increase in selected content. Figure 9 shows how this difference is distributed over separate *Palm VII*™ headline screens. For each headline screen, this plot compares the probability that a selected story was an adaptive story to the probability that the story was presented in static order. More formally, the plot compares the conditional probabilities $p(adaptive \mid selected)$ and $p(static \mid selected)$ for separate headline screens.

Figure 9 shows that the difference in selection probabilities is particularly noticeable on the first headline screen and then decreases gradually from page to page. On the first headline screen, $p(static \mid selected)$ is 0.33 vs. 0.66 for $p(adaptive \mid selected)$. This difference indicates that the adaptive display strategy indeed helps users locate relevant content, as users prefer adaptive stories over static stories on average.

In summary, the "alternating sessions" and "alternating stories" experiments both show that adaptive information access is superior to static access. The "alternating sessions" experiment demonstrated that the adaptive order helps to

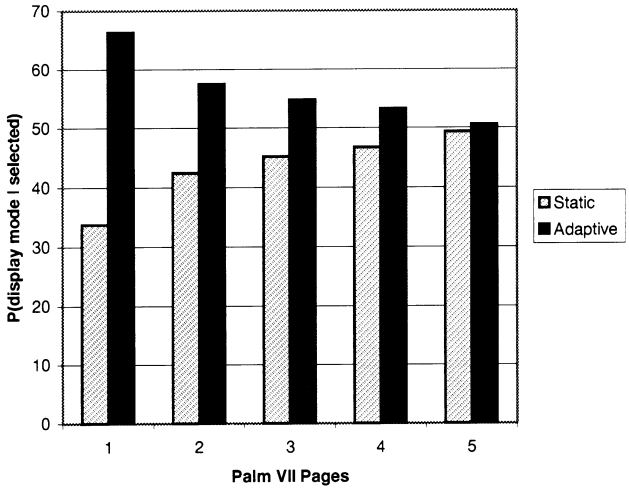

Figure 9. Static vs. adaptive selection probabilities/no top stories.

move interesting items towards the beginning of personalized item lists, simplifying access to relevant content. The ''alternating stories'' experiment showed that the system is capable of ordering content in a way such that the top-ranked stories have a significantly higher chance of being selected than the top-ranked stories obtained from a static order.

6. Related Work

Most work on content-based information filtering casts the automated acquisition of user models as a text classification task (Pazzani and Billsus, 1997; Lang, 1995; Mooney et al., 1998). An underlying assumption often made is that more training data leads to improved predictive performance. However, if we take into account that a user's interests are dynamic and are likely to change over time, this assumption does not hold. A classifier built from a large number of training documents that accurately reflect the user's past interests is of limited practical use and might perform substantially worse than a classifier limited to recent data that reflects the user's current interests. This example illustrates that a good text classification algorithm is not necessarily a useful user modeling algorithm. From a machine learning perspective, this is a challenging problem known as concept drift (Widmer and Kubat, 1996). As researchers have begun to take the importance of concept drift for user modeling applications into account, a few initial solutions have emerged in the literature. A straightforward approach is simply to place less weight on older observations of the user (Webb et al., 1996). However, there is some evidence that the effectiveness of this simple approach is constrained (Webb et al., 1997). Klinkenberg and Renz (1998) explore windowing techniques similar to ideas on handling *concept drift* proposed by Widmer and Kubat (1996) in the context of Information Retrieval. The central idea is to limit training data to an adjustable time window, where the window size depends on observed indicators such as sudden changes in term distributions.

Chiu and Webb (1998) have previously studied the utility of the induction of two separate user models in the context of student modeling. While the studied domain, data representation and learning algorithms differ significantly from the text classification approach presented here, the underlying motivation for the use of a ''dual'' model is similar. In general, user modeling is a task with inherent temporal characteristics. We can assume recently collected user data to reflect the current knowledge, preferences or abilities of a user more accurately than data from previous time periods. However, restricting models to recent data can lead to overly specific models, i.e. models that classify instances that are similar to recently collected data with high precision, but perform poorly on instances that deviate from data used to induce the model. To overcome this problem, Chiu and Webb use a dual model that classifies instances by first consulting a model trained on recent data, and delegating classification to a model trained over a longer time period if the recent model is unable to make an accurate prediction.

A few alternative approaches to personalized news access based on learning algorithms have recently been reported in the literature. Here, we briefly summarize the main characteristics of three systems: PZ (Veltmann, 1998), *Anatagonomy* (Sakagami and Kamba, 1997) and *P-Tango* (Claypool et al., 1999).

The *PZ* system is an agent that generates personalized newspaper digests (Veltmann, 1998). The system is specifically designed to represent multiple topics of interest per user, and takes into account that users' interests may change frequently. To achieve this functionality, *PZ* is based on a multi-agent design, where each agent models a different facet of the user's interest. Three different agent types are supported. The *General Agent* monitors all articles and learns from all user feedback. The *Section Agent* is restricted to individual news categories, such as business or politics. *Topic Agents* represents specific topics of interest to the user. The dynamic nature of user interests is taken into account by charging each agent with an initial amount of ''energy'', which decreases over time if the user does not indicate further interest in a topic. As soon as an agent's energy drops below zero, it is deleted from the system. Likewise, new agents can be generated dynamically. This results in an evolving set of topic agents that model a user's current interests. Each individual agent is represented as a prototype document vector derived with Rocchio's algorithm (Rocchio, 1971). Users can provide explicit feedback for news stories, rating them as either interesting or not interesting. In addition, the system collects implicit feedback, where selecting and reading an article counts as implicit positive feedback.

The *ANATAGONOMY* system is a research prototype of a personalized online newspaper (Sakagami and Kamba, 1997). The system represents a user's interests as a vector of words and associated weights. New stories are scored based on their similarity to this vector. The system's main research purpose is the exploration of the utility of implicit user feedback. Both explicit and implicit feedback mechanisms are supported. A score bar indicates the predicted score for each story that the system recommends to the user. Users can provide explicit feedback by adjusting the predicted score. In addition, the user interface allows for enlarging articles, as well as scrolling through text using scroll bars. These interactions are interpreted as implicit feedback. A story receives a ''score bonus'' if the user enlarges it or scrolls through it. Based on an initial experimental evaluation with 15 system users, the authors conclude that the success of implicit feedback depends on the particular values assigned to user actions such as enlarging and scrolling. Explicit feedback always led to better predictive performance than implicit feedback alone. However, combining explicit and implicit feedback resulted in predictive performance that was only slightly worse than explicit feedback alone, even though the combined approach was trained on only one third of the explicit ratings. These experiments suggest that implicit feedback can help reduce the amount of required user feedback significantly.

P-Tango combines content-based and collaborative techniques to provide personalized access to daily news stories. Recommendations for individual stories are determined using both a keyword-based text classification component and a col-

laborative filtering algorithm. These two models are maintained separately. An overall prediction is derived as a weighted combination of both model predictions. The combination weights are learned on a per-user basis, and are adjusted over time so that the system's past prediction error is minimized.

In summary, all three described news systems were experimentally shown to provide accurate personalized news access. However, none of the described systems is geared towards immediate adaptation to changing interests (in contrast to our hybrid user model), and none of the described systems attempts to prevent presenting information the user is likely to know (in contrast to our knowledge-dependent classification approach).

7. Discussion and Future Work

One of the most noticeable findings of our empirical evaluation is that the *Palm VII*™ version currently attracts many more regular users than the web version. Overall, more than 80% of all users logged into the *Adaptive Information Server* through a *Palm VII*™ device. We attribute this effect to two main reasons. First, the Palm VII(version does not require explicit ratings and is therefore easier to use than the web version. Second, the bandwidth and display size constraints create an even stronger need for adaptive information access on wireless devices than on the web. Our future work will be based on these findings.

Support for additional clients is currently being added to the *Adaptive Information Server*. The University of California has licensed *AIS* to *AdaptiveInfo.com* that plans to make the system available on additional wireless information platforms. A version implemented for the "Wireless Application Protocol (WAP)" that can send content to Internet-enabled cell phones presents a related set of challenges because these devices have smaller screens, but a slightly faster network connection than the Palm VII.

The *Adaptive Information Server* determines which stories to recommend based on estimated relevance to the user. However, when compiling an overall personalized news digest, relevance is not the only factor that should be taken into account. We expect an explicit model of the diversity of a news digest to improve the overall utility and user acceptance of our recommendation algorithm. Generating news digests that contain recommended stories spanning a broad range of different topics will be the subject of future research.

Finally, we plan to extend our user modeling approach with a collaborative filtering component. Content-based techniques alone are unlikely to provide solutions to all the challenges encountered in intelligent information access tasks. Content-based algorithms typically perform well at determining the general topic of a document, but they cannot easily evaluate qualitative attributes such as style, usefulness, or timeliness. Collaborative-filtering avoids this shortcoming through the use of aggregated user ratings. However, dynamic domains, i.e. domains in which new information is becoming available rapidly and ages quickly, are known to be

problematic for collaborative filtering techniques. This is primarily due to the small number of user ratings for individual items. As a result, we expect commonly used correlation-based approaches to collaborative filtering (Shardanand and Maes, 1995) to perform poorly. The computational complexity of these algorithms poses an additional challenge for AIS, as scalability to large numbers of users is one of our primary design goals. In future research, we will explore the integration of efficient collaborative filtering techniques into the *Adaptive Information Server*.

8. Summary and Conclusions

We presented the design, algorithms, deployment and evaluation of the *Daily Learner*, an agent that adapts to its users' individual interests in daily news stories. Driven by requirements and constraints of real-world deployment, the system is based on a client/server architecture that supports multiple different clients, geared towards different usage scenarios. Here, we focused on two versions of the *Daily Learner*. While the web-based version learns from explicit user feedback, the wireless *Palm VII*™ version is based on implicit feedback, obtained by observing the users' browsing patterns. Our system uses a multi-strategy machine learning approach to induce hybrid user models that model a user's short-term and long-term interests separately. Based on data collected from a user base of about 3,000 registered users we empirically evaluated the system's overall performance and analyzed performance contributions of individual model components. In an additional experiment, we assessed the overall utility of our user modeling algorithm by comparing its adaptive characteristics to static, non-personalized news access. Our results provide empirical evidence for the utility of the proposed hybrid user model and suggest that effective personalization can be achieved without requiring any extra effort from the user.

References

Allan, J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y.: 1998, Topic detection and tracking pilot study final report. *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 194–218, Lansdowne, VA.

Balabanovic, M.: 1998, *Learning to surf: multiagent systems for adaptive web page recommendation.* Ph.D. Thesis, Stanford University.

Bauer, M., Gmytrasiewicz, P. and Pohl, W.: 1999, Machine learning for user modeling, *Seventh International Conference on User Modeling*, Banff, Canada.

Belkin, N.: 1997, User modeling in information retrieval. [online]. Available: *http://www.scils.rutgers.edu/~belkin/um97oh/*. (June 7, 2000).

Billsus, D. and Pazzani, M.: 1999a, A personal news agent that talks, learns and explains, *Proceedings of the Third International Conference on Autonomous Agents*, Seattle, WA, pp. 268–275.

Billsus, D. and Pazzani, M.: 1999b, A hybrid user model for news story classification, *User Modeling: Proceedings of the Seventh International Conference (UM99)*, Banff, Canada, pp. 98–108.

Chiu, B. and Webb, G.: 1998, Using decision trees for agent modeling: improving prediction performance. *User Modeling and User-Adapted Interaction*, **8**, 131–152.

Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D. and Sartin, M.: 1999, Combining content-based and collaborative filters in an online newspaper. *ACM SIGIR Workshop on Recommender Systems*, Berkeley, CA.

Cohen, W. and Hirsh, H.: 1998, Joins that generalize: text classification using WHIRL, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, NY, pp. 169–173.

Dietterich, T.: 1998, Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, **10**(7), 1895–1924.

Duda, R. and Hart, P.: 1973, *Pattern Classification and Scene Analysis*, New York, NY: Wiley.

Jameson, A., Paris, C. and Tasso, C. (eds.): 1997, *User Modeling: Proceedings of the Sixth International Conference (UM97)*, New York: Springer.

Joachims, T., McCallum, A., Sahami, M. and Ungar, L. (eds.): 1999, *IJCAI Workshop IRF2: Machine Learning for Information Filtering*, Stockholm, Sweden.

Kay, J. (ed.).: 1999, *User Modeling: Proceedings of the Seventh International Conference (UM99)*, Banff, Canada.

Klinkenberg, R. and Renz, I.: 1998, Adaptive information filtering: learning in the presence of concept drift, *AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05, Madison, WI.

Lang, K.: 1995, NewsWeeder: learning to filter news, *Proceedings of the Twelfth International Machine Learning Conference (ICML '95)*, Lake Tahoe, CA, pp. 331–339.

Lewis, D. and Gale, W.A.: 1994, A sequential algorithm for training text classifiers, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Dublin, Ireland, pp. 3–12.

Lieberman, H.: 1995, Letizia: An agent that assists web browsing, Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, Canada, pp. 924–929.

McCallum, A. and Nigam, K.: 1998, A comparison of event models for naive bayes text classification, *AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05, AAAI Press, pp. 41–48.

Mooney, R., Bennet, P. and Roy, L.: 1998, Book recommending using text categorization with extracted information. *AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05, AAAI Press, pp. 49–54.

Papatheodorou, C. (ed.).: 1999, Machine learning and applications workshop. *Machine Learning in User Modeling*, Chania, Greece.

Pazzani, M. and Billsus, D.: 1997, Learning and revising user profiles: the identification of interesting web sites. *Machine Learning,* **27**, 313–331.

Quinlan, J.: 1986, Induction of decision trees. *Machine Learning*, **1**, 81-106.

Rocchio, J. (1971). Relevance feedback in information retrieval, In: G. Salton (ed.). *The SMART System: Experiments in Automatic Document Processing*, NJ: Prentice Hall, pp. 313–323.

Rudstrom, A., Bauer, M., Iba, W. and Pohl, W. (eds.).: 1999, *IJCAI Workshop ML4: Learning About Users*, Stockholm, Sweden.

Sakagami, H. and Kamba, T.: 1997, Learning personal preferences on online newspaper articles from user behaviors. *Proceedings of the Sixth International World Wide Web Conference (WWW6)*, Santa Clara, CA, pp. 291–300.

Salton, G.: 1989, *Automatic Text Processing*, Addison-Wesley.

Shardanand, U. and Maes, P.: 1995, Social information filtering: algorithms for automating 'word of mouth', *Proceedings of the Conference on Human Factors in Computing Systems (CHI95)*, Denver, CO, pp. 210–217.

Veltman, G.: 1998, A multi-agent system for generating a personalized newspaper digest. *AAAI/ICML-98 Workshop on Learning for Text Categorization*, Technical Report WS-98-05, AAAI Press, pp. 99–102.

Webb, G., Chiu, C. and Kuzmycz, M.: 1997, Comparative evaluation of alternative induction engines for feature based modeling. *International Journal of Artificial Intelligence in Education*, **8**, 97–115.

Webb, G. and Kuzmycz, M.: 1996, Feature based modeling: a methodology for producing coherent, consistent, dynamically changing models of agents' competencies. *User Modeling and User Assisted Interaction,* **5**(2), 117–150.

Widmer, G. and Kubat, M.: 1996, Learning in the presence of concept drift and hidden contexts. Machine Learning, **23**, 69–101.

Yang, Y.: 1999, An evaluation of statistical approaches to text categorization. *Information Retrieval*, **1**(1), 67–88.

Author's Vitae

Daniel Billsus received a diploma in computer science from the Technical University of Berlin, Germany, and MS and PhD degrees from the University of California, Irvine. His research focus has been in the area of intelligent information access. He studied the use of machine learning techniques as part of various information agents, leading to his doctoral dissertation on ''User Model Induction for Intelligent Information Access''.

Michael Pazzani is a professor and the chair of the Information and Computer Science Department at the University of California, Irvine. His research interests include data mining and intelligent agents. He received his BS and MS in computer engineering from the University of Connecticut and his PhD in computer science from UCLA. He is a member of the AAAI and the Cognitive Science Society.