User Modeling and User-Adapted Interaction **6**: 225–261, 1996.

User-Centered Indexing for Adaptive Information Access

NATHALIE MATHÉ and JAMES R. CHEN*
NASA Ames Research Center, Mail Stop 269–2, Moffett Field, CA 94035, U.S.A.
E-mail: mathe@ptolemy.arc.nasa.gov, jchen@ptolemy.arc.nasa.gov

(Received 2 March 1995; in final form 13 March 1996)

Abstract. We are focusing on information access tasks characterized by large volume of hypermedia connected technical documents, a need for rapid and effective access to familiar information, and long-term interaction with evolving information. The problem for technical users is to build and maintain a personalized task-oriented model of the information to quickly access relevant information. We propose a solution which provides user-centered adaptive information retrieval and navigation. This solution supports users in customizing information access over time. It is complementary to information discovery methods which provide access to new information, since it lets users customize future access to previously found information. It relies on a technique, called Adaptive Relevance Network, which creates and maintains a complex indexing structure to represent personal user's information access maps organized by concepts. This technique is integrated within the Adaptive HyperMan system, which helps NASA Space Shuttle flight controllers organize and access large amount of information. It allows users to select and mark any part of a document as interesting, and to index that part with user-defined concepts. Users can then do subsequent retrieval of marked portions of documents. This functionality allows users to define and access personal collections of information, which are dynamically computed. The system also supports collaborative review by letting users share group access maps. The adaptive relevance network provides long-term adaptation based both on usage and on explicit user input. The indexing structure is dynamic and evolves over time. Learning and generalization support flexible retrieval of information under similar concepts. The network is geared towards more recent information access, and automatically manages its size in order to maintain rapid access when scaling up to large hypermedia space. We present results of simulated learning experiments.

Key words: user-centered indexing, long-term adaptation, adaptive information retrieval, adaptive navigation, user feedback, shared information access

1. Introduction

1.1. Human Problem

We are interested in facilitating access to reference information contained in large volume of on-line technical and operational manuals. Technical reference information is used in many fields by professionals like lawyers, doctors, power plant controllers, airplane mechanics, pilots, astronauts, flight controllers, etc. These professionals need to access a large amount of technical information to perform their

* Dr. Mathé and Dr. Chen are contractors with Recom Technologies, Inc.

everyday job. Moreover, technical information is usually highly cross-referenced, creating a huge space of hypermedia connected documents.

For these users, the context of the information access task is different from the one assumed by traditional information retrieval tasks. In general, they already know most of the information they need and where it is stored. However to prevent human errors, they are usually required to quickly access information relevant to specific tasks they are performing. They rarely need to search for new information. When they do find new interesting information, they need to memorize where to find it for future reference. For example, this type of information access is of great importance to Space Shuttle flight controllers at NASA Johnson Space Center (JSC). It takes years of training to become a flight controller in the Space Shuttle Mission Control Center. As part of this training, controllers learn to use a large corpus of documentation to solve problems. They develop a deep knowledge of the organization and content of these manuals in order to access the proper sections as quickly as possible. This knowledge, developed as part of a situational and task-oriented process, is highly context-dependent and user-dependent (Boy, 1991). Contextual factors such as frequency of access, date of last use, relative relevance to a task, and individual user preferences have been shown to be very important to classify, organize and access information from an individual point of view (Barreau, 1995). Therefore there is a need to support information access based on contextual use and individual user preferences.

Another aspect of the problem is the time span of the information need: technical information access is a long term process, in the sense that information pertaining to a particular task needs to be accessed each time this task is performed. However, human memory of information organization and content does not persist very long unless the same information is accessed often under the same conditions. Users therefore develop artifacts to support their memory, like hand-written annotations, Post-It notes, bookmarks; or they create condensed representations of important information, like cue cards, quick access guide, abbreviated checklists. For example, each flight controller develops a personal collection of information selected from existing Shuttle operations documents, which is referred to as a "goody book". A personal "goody book" is used to access critical or often-used information faster. Therefore there is a need to support customization of information access by individual users.

Another problem related to the long term aspect of this information task, is due to frequent updates and revisions of technical manuals. Users have to continuously revise what they know, as well as manually update their personal annotations and quick-access collections. This is time consuming, and creates a potential safety risk. Therefore there is a need to support persistence and incremental modification of customized information access.

To summarize, technical information access task is characterized by a large volume of hypermedia connected documents, the need for rapid and effective access to already known information, and long-term interaction with evolving information.

The problem for technical users is to build and maintain a personalized task-oriented model of the information in order to quickly access information relevant to specific tasks at hand.

1.2. TECHNICAL PROBLEM

Two main ways of accessing on-line information are querying and browsing. Querying consists in providing a description (query) of the information being sought, and having an information retrieval system locates information that matches the description (Salton, 1989). Browsing assumes that the information has already been organized (usually in a hierarchical structure like a table of contents, or with cross-reference hyperlinks), and a hypertext system provides means for the user to navigate within this structure (e.g., World Wide Web browser like Mosaic).

Many useful techniques in traditional information retrieval have been developed in the past few decades (van Rijsbergen, 1979; Salton, 1989). Inverted index has long been used for fast and effective retrieval. Boolean retrieval models compare Boolean queries with term sets used to identify document contents. Extended Boolean systems employ query and document term weights to generate ranked output documents. Vector Space models represent both queries and documents by term sets and computes similarity measures between them (Salton et al., 1975). Latent Semantic Indexing uses singular value decomposition to encode compressed representation of a term-document matrix in vector space, which facilitates both efficient and generalizable retrieval (Dumais et al., 1988). Probabilistic indexing models (Fuhr, 1992) follow the Probability Ranking Principle to achieve better retrieval performance (Robertson, 1977). A Bayesian Inference Network retrieval model regards information retrieval as uncertain inference (Turtle & Croft, 1991). It provides a unified representation framework of different models, and allows the integration of multiple sources of evidence and the combination of different queries and query types. These information retrieval systems require significant preprocessing to extract indexing information from the content of a relatively static set of documents. These information retrieval techniques are very effective, but with huge spaces of documents, they might still retrieve too many relevant documents based on content only. There is a need to further discriminate among relevant documents based on tasks users are engaged in and individual user preferences, which evolve over time.

On the other hand, hypertext systems provide user-driven navigation. Their main strength relies on a fixed structure to support navigation, but this is also their main weakness, since the structure is usually authored by someone else than the user (or automatically generated by a software program), and cannot be easily modified. Knowledge-based hypertext systems add a second-level structure (often referred to as a thesaurus, semantic net, domain knowledge, or index space) on top of the basic document hypertext structure, in order to provide more flexible and intelligent navigation (Agosti et al., 1995; Belkin et al., 1993; Tudhope et al., 1995). Concepts

Concepts in the index space are assigned to hypertext nodes in the document space. Navigation in the document space is performed indirectly through navigation in the index space, and hyperlinks can be dynamically computed. Constructing the index space and associating concepts to hypertext nodes is usually performed manually by domain experts, and/or automatically pre-computed from existing thesaurus. As for the document-level structure, this second-level structure is usually pre-computed and authored by someone else than the user. With large hypermedia systems, users might however "get lost in hyperspace" without intelligent navigation support adapted to each user's needs.

In term of customization at the document level, most hypertext systems let users add their own annotations and sometimes hyperlinks on top of the existing hypertext structure. They usually offer a mechanism to create a quick-access list of user-selected information (Hotlist or Bookmarks list in WWW browsers), hierarchically organized under unique headings. However, these customization techniques do not scale up with large hypermedia spaces. There is a need to automatically filter and adapt pages and links for different users and different tasks. The work presented in this paper focuses primarily on customized information access for individual users and groups of users. Other papers in this issue (Vassileva, 1996; Höök et al., 1996) focus on task-oriented information access, and are discussed in the related work section (Section 2), since the two approaches are related and complementary.

1.3. Proposed Solution

A solution to the problem of quickly accessing information contained in large volume of hypermedia connected documents relies on providing user-centered adaptive information access, both for querying and browsing; as well as on providing tools to continuously let users customize information access over time. We consider three ways of adapting information access (Brusilovsky, 1994): (1) adaptive information retrieval, (2) adaptive hypermedia presentation, and (3) adaptive hypermedia navigation.

Adaptive information retrieval allows users to access information using personal conceptual descriptors, in addition to usual keywords. For example, given a query containing a list of keywords and a list of personal concepts, an adaptive information access system retrieves a ranked list of information units relevant to the query. Depending on particular applications, information units might correspond to entire documents, pages, or hypertext nodes. Conceptual descriptors might correspond to tasks, problems, goals, topics explicitly specified by the user, or automatically deduced by the system.

Adaptive hypermedia presentation changes the set of visible links displayed to the user. For example, given a hypertext node selected by the user, and a set of personal concepts, the system filters existing hyperlinks associated to the selected node, and displays only existing links relevant to the set of personal concepts.

Adaptive hypermedia navigation changes the layout of links (by re-ordering existing links, or by generating new links) to provide intelligent guidance to the user. For example, given a selected hypertext node, and a set of personal concepts, the system dynamically generates a list of hypertext nodes destinations likely to be relevant to the set of personal concepts, given the starting hypertext node. Such adaptive hyperlinks are computed on the fly, and are dynamic instead of static links.

We propose a technique, called Adaptive Relevance Network (ARN), which supports adaptive information retrieval, as well as adaptive hypermedia presentation and navigation, based on a second-level user-centered indexing structure. Information access is customized to individual user' needs. ARN supports users in creating and managing personalized information access maps, which are organized by concepts. Each concept is user-defined and is described by a descriptor, corresponding to a task, a problem, a goal, a topic, etc. Our approach is complementary to information discovery methods which provide access to new information, since it lets users customize future access to previously found information.

The adaptive relevance network creates and maintains a complex indexing structure to store individual user's information access maps (Mathé & Chen, 1994) (see Sections 4.1 and 4.2). Adaptation is based both on usage (automatic adaptation), and on explicit user input (adaptation prompted by user) for assigning user-defined concepts to information units, and for updating relevance weights between concepts and information units (see Section 4.3). Direct adaptation by the user is also supported with tools for editing concept descriptors. Flexible retrieval is provided by the ability to compute relevant information from incomplete relevance knowledge stored in a network (see Section 4.4). The network maintains a balance between memorization and generalization, to support retrieval of information under similar sets of concepts. The indexing structure is dynamic: ARN incorporates incremental user input and dynamically adjusts the indexing structure and relevance weights over time. It is geared towards more recent information access, and can adjust the speed at which it forgets, as well as the number of inputs after which it forgets (see Section 4.5).

To store user's information access maps, we chose an indexing structure rather than a hierarchical structure, since it is generally recognized that multiple information access points are needed (Barreau, 1995). We chose a complex structure (a composite-index network), rather than a flat index structure, to be able to capture non-linear aspect of conceptual knowledge, such as exceptions (Boy, 1991). In the context of this paper, non-linear means that a portion of relevant information units cannot be derived from information units relevant to individual concepts, but only from information units relevant to a specific set of concepts, representing a complex concept. These complex concepts are represented by composites nodes in a network. Relations between single and complex concept nodes denote a sub-set relationship, rather than a hierarchical relationship. Lastly, we implemented the adaptive relevance network so that it provides rapid information access, and automatically manages its size in order to maintain rapid access when scaling up to large hypermedia space (see Section 4.5).

We integrated our adaptive relevance network technique into the HyperMan hypermedia system (Rabinowitz et al., 1995). Space Shuttle flight controllers at NASA/JSC use HyperMan to access operations documents in mission control. Adaptive HyperMan builds upon the hypertext features of HyperMan, and helps users organize and access large amount of information. It lets users select and mark any part of a document as interesting, and index that part with user-defined concepts, corresponding to particular flights, simulations, problems, systems, tasks, goals, or topics. Users can then do subsequent retrieval of marked portions of documents by concepts. This functionality allows users to define and access personal collections of information, which are dynamically computed, thus providing a virtual "goody book" facility to individual flight controllers. Adaptive HyperMan provides user-driven and long-term adaptation: it lets users build their individual information access maps, and automatically keeps them up to date. The system also supports collaborative review by letting users build group information access maps using a shared list of concepts.

Lastly, our adaptive relevance network technique is independent of any particular information system or document format. It is complementary to other information retrieval and hypermedia systems which support querying and browsing, and discovery of new information. In fact it has also been successfully integrated with the Boeing Portable Maintenance Aid system for airplane mechanics, and with the World Wide Web for organizing and sharing personal pages collections. These applications are at the prototype and testing stage, and are not described in this paper.

In the following section, we present an overview of related approaches to adaptive information access. In the third section, we describe the Adaptive HyperMan system and illustrate its use with an example scenario. The fourth section describes in details the structure of the Adaptive Relevance Network, its learning method based on user input, and its adaptive information access method. Results of simulated learning experiments are presented in the fifth section. We propose future directions of research in the sixth section, and then conclude.

2. Related Work

In Section 1.2, we reviewed traditional information retrieval techniques and knowledge-based hypertext systems, and their limitations in adapting to individual users' needs. We now briefly compare our proposed solution to these techniques, and then review related work on adaptive information retrieval, adaptive information filtering, and adaptive hypermedia.

While traditional information retrieval systems are very effective at retrieving information from large set of documents based on content, they are not geared towards building customized personal indices, nor supporting incremental modifications by users over time. The ARN technique we propose supports a customizable and sharable, personal index system, which is complementary to existing information retrieval systems. It is similar to an inverted index with a more complex composite index structure. Simple probabilistic estimates are used to quickly represent and compute relevance measures. Although ARN does employ a network architecture and probabilistic relevance estimates similar to those of a Bayesian Inference Network, it is different. ARN focuses on maintaining a dynamic and effective indexing structure, instead of on optimizing probabilistic estimates. Lastly, some techniques like logical queries and/or queries with weighted terms, which are not currently supported in our application interface, could be integrated to accommodate more complex queries.

Short-term adaptive information retrieval systems provide relevance feedback mechanisms to help users formulate queries, and improve retrieval. User feedback is used to modify the current query, either query terms and/or weights (Harman, 1992; Haines & Croft, 1993; Robertson & Sparc Jones, 1976). Our work primarily uses user feedback to capture user-centered indexing information over time and across several queries, instead of refining the formulation of a single query. Standard relevance feedback method, however, is also applicable to our personal indexing system. Another approach of adaptive information retrieval uses a connectionist network to learn the information space structure over time from users' combined input (Belew, 1989; Rose & Belew, 1991). These systems usually require a long training period before reaching a state of fertility (Chen, 1995).

Adaptive information filtering examines a continuous stream of incoming documents and display only these that are relevant to a user's long term interests (Callan et al., 1992). Most information filtering methods rely on modeling and learning user interests over time. Although we are not modeling users' interests, but rather modeling information access and organization, we face similar issues in modeling changes over time, and in sharing information access maps with other users. Jennings and Higichi have embedded a connectionist model of long-term user interests into a system for reading Usenet news (Jennings & Higichi, 1993). They apply supervised learning to manage large search space without extensive knowledge engineering. Fischer and Stevens apply a rule-based technique to suggest boolean search agents for reading Usenet news (Fischer & Stevens, 1991). Sheth and Maes (1993) use a genetic algorithm to evolve boolean search agents parameters. The main problem is to model changes and persistence in user interests, as well as interaction between interests. This often requires combining machine learning techniques with interest management through user interaction. Collaborative filtering methods go one step further by involving the explicit advice of other users (Pargman, 1994). Collaborative filtering might be based on other users' actions (to search information read by another user (Goldberg et al., 1992)), or based on other users' evaluations (to search information that has received positive feedback from other types of users, or from users with similar interests (Bergstrom & Riedl, 1994)). Maes and Kozierok apply memory-based learning and learning by example from both individual user and similar users' agents (Maes & Kozierok, 1994).

Adaptive hypermedia systems have been developed to tailor information content and hyperlinks to different classes of users with different goals and knowledge, and to provide individual guidance through large hypermedia spaces (Brusilovzky, 1994). Typical application domains are educational and tutoring systems, advanced help and explanations, and on-line documentation. Most adaptive hypermedia techniques in tutoring systems rely on a domain model, implemented as a more or less complex semantic net, and on relations between hypertext nodes and concepts in the domain model. The individual user model is usually an overlay model of what the user knows about the domain, or a simpler stereotypical model. The systems are complex and require extensive knowledge acquisition and modeling, which does not scale up easily.

Simpler approaches have been developed for on-line documentation. Adaptive hypermedia presentation automatically changes the information displayed, or the set of visible links (by hiding irrelevant information or links, or by highlighting relevant ones). Armstrong et al. have proposed a learning by example apprentice, which help users locate desired information by highlighting recommended links in a Web document (Armstrong et al., 1995). Recommendations are based on the user's information goal, and the content of the previously selected hypertext node, its surrounding sentence and its parent headings. The user provides feedback by accessing or not the suggested nodes. This system requires pre-processing a set of training data first, and does not learn in an incremental manner. Vassileva proposes to limit the browsing space according to the current task performed by the user and her level of experience (Vassileva, 1996; Vassileva, 1994). The system utilizes pre-defined tasks hierarchies, which are acquired by empirical analysis of the domain. Hypertext nodes are associated to tasks under which they are relevant (indexing done a priori by a knowledge engineer). Users manually select a current task, which provides a starting point for browsing as well as a filter by providing access only to nodes relevant to the current task. User's level of experience is automatically adjusted by the system based on analysis of previous user navigation actions. The system also supports users in directly modifying the task hierarchy, creating new tasks or changing their level of experience. This approach is similar to ours, the main difference being that the task model is pre-defined by a knowledge engineer, instead of being incrementally learnt from the user. This is appropriate for small domains with relatively stable hierarchical task models.

Adaptive navigation automatically changes the layout of links to provide guidance, by re-ordering links, augmenting links with dynamic comments, or suggesting most relevant links to follow. Kaplan et al. proposed a model to memorize user preferences about relevance between topics in an associative matrix (Kaplan et al., 1993). The system is based only on user input, and does not generalize the information for different usage. Other adaptive navigation methods utilize learning by observation techniques to automatically learn patterns of navigation based on frequency of nodes accesses (Kibby & Mayes, 1989; Monk, 1989). Similar to Vassileva's and our approaches, Höök et al. propose to adapt hypermedia using knowledge about the user's current task (Höök et al., 1996). They developed a help assistant system which provides both adaptive presentation of content, and adaptive navigation with follow-up questions most relevant to user's task. The current task is either directly selected by the user, or automatically selected using plan recognition. This approach assumes relatively stable hierarchical task models, and a priori knowledge acquisition of a task hierarchy might become a problem for large domains.

At the other end of the spectrum, fully user-driven customization systems, which don't rely on any a priori knowledge, have been developed to help users organize their personal information space. The Warmlist system is a personal Internet assistant which lets users build their own collections of interesting WWW documents, organize them hierarchically, index them with a full-text indexing engine, and share them with other users (Klark & Manber, 1995). The Active Notebook system lets users label WWW documents with conceptual classifications, and organize these documents into a semantic taxonomy, which can be shared with other users (Torrance, 1995). Because they are fully manual, these systems do not provide any intelligent support to users in creating hierarchies and taxonomies, and their approach might not scale up to large information spaces.

To summarize, we propose to use a second-level conceptual indexing structure, similar to the one used in knowledge-based and adaptive hypertext systems. But, instead of being manually acquired from experts or pre-computed, this structure is incrementally learnt over time from user input, and therefore customized to individual user's needs. Compared to personal information space organizers, our approach is automated: users assign concepts to information units, and our system automatically builds a conceptual network structure, which is used for adaptive information access. The problem of sharing complex taxonomies among users is tackled by combining input from multiple users into a group network structure.

3. Adaptive HyperMan System

This section illustrates our user-centered indexing approach to adaptive information access with the Adaptive HyperMan system (AHM). AHM lets users select and mark any part of a document as interesting, and index that part with user-defined concepts, corresponding to particular flights, simulations, problems, systems, tasks, goals, or topics. Users can then do subsequent retrieval of marked parts of documents by concepts. This functionality allows users to define and access personal collections of information, which are dynamically computed, thus providing a virtual "goody book" facility to individual flight controllers. The system also supports collaborative review by letting users build group information access maps using a shared list of concepts.

3.1. The HyperMan viewer

HyperMan 2.1 is a software tool for document viewing and parsing, developed as part of the Electronic Documentation Project (EDP) at NASA/JSC (NASA JSC–26679, 1994). The goal of the EDP project is to provide an electronic capability to support authoring, distribution, viewing, and controlled revision of crew and ground controller operations documents, for use in Mission Control Center and in their office environment. EDP integrates the state-of-the-art hypertext document viewer, HyperMan, with JSC flight planning and scheduling tools, and commercial workflow automation tools. Starting with a literal representation of the current paper-based system, HyperMan extends that metaphor with hypertext capabilities. HyperMan is a full blown wysiwyg PDF (Adobe's Portable Document Format) viewer designed for hundreds of simultaneous users. HyperMan is being used by flight controllers in support of Space Shuttle missions since July, 1995.

To answer flight controllers' need for intensive customization of documents, HyperMan provides the ability for end-users to create and store various types of visual markers in a document (Figure 1). User create visual markers on a page by first selecting a tool in the tools palette, then selecting a portion of the page, or a location in the page. Different types of visual markers are available: *color highlight* (changing background color of selected text or graphical region), *colored text* (changing foreground color of selected text), an *anchor icon*, a *bookmark icon*, a *notepad icon* (hidden note), or a *sticky note* (visible note). Users can also create hyperlinks between any markers (both inter- and intra-book, and uni- or bi-directional) by selecting the linking tool in the tools palette and selecting two markers. To support collaborative work, users can publish their markers and links, and subscribe to published markers and links from various user groups.

Other useful features include full text search, automatic hyperlinking at parsing time, version control (to retain user markers and hyperlinks between versions of documents), and transportable hypertext layer (each user's markers and links are stored separately from the documents, so that the document itself is never altered by the creation or deletion of markers or links). We do not have space to describe the HyperMan system in full details, and will focus in the next sections on the adaptive functionality we added to the system.

3.2. Adaptive HyperMan

Adaptive HyperMan (AHM) builds upon the hypertext features of HyperMan by allowing users to create markers on a page to mark any part of a document as interesting, and to assign conceptual descriptors to selected markers in documents. Users can then retrieve markers that are relevant under particular conceptual descriptors. These conceptual descriptors are called *topics* in AHM user interface.* Topics cor-

* This supplements full text search by keyword capability, as markers may have been indexed under topics that never appear as words on that page, or markers may be associated with graphics.

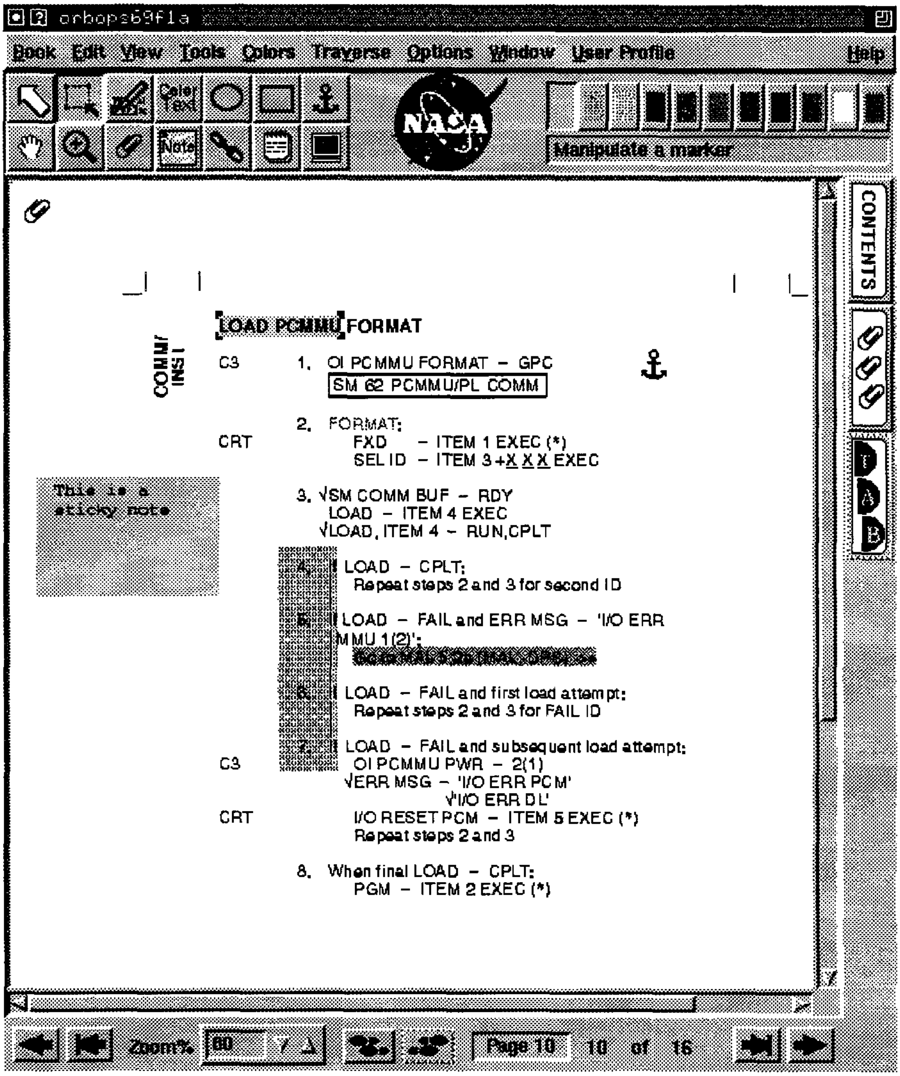


Figure 1. HyperMan book window. The annotation tools palette is shown at the top left, below the menu bar. Various visual markers are shown on the page. The User Profile menu has been added for the adaptive version of HyperMan.

respond to particular flights, simulations, problems, systems, tasks, goals, or any concept relevant to facilitate information access. They are either offered by the user, or chosen from the system-defined list, which is used to facilitate sharing of indexing information among groups of users. AHM also lets users provide feedback over time to update the relevance of markers to topics. The system utilizes the Adaptive Relevance Network mentioned in Section 1.3 (and fully described in Section 4) to store these users' personal information access maps. AHM supports both adaptive retrieval of markers, and adaptive navigation. Adaptive presentation of markers has not been implemented.

For *adaptive retrieval*, given a particular user and a query containing a list of topics, the system computes a ranked list of markers relevant to the set of topics for this user, by accessing his/her personal information access map stored in a relevance network structure. For *adaptive navigation*, navigation in the document space is performed indirectly through navigation in the relevance network structure, in an approach similar to those proposed in (Agosti et al., 1995; Belkin et al., 1993; Tudhope et al., 1995). More specifically, given a particular user and a marker selected by the user, the system dynamically computes a ranked list of destination markers relevant to the selected marker, by using as a query the set of topics previously assigned to this marker by the user. If no topics have been assigned to the selected marker, the user can add topics in the query window, or index the selected marker first. Therefore virtual links are dynamically computed using the same relevance network used for adaptive retrieval. In the rest of the paper we will focus on adaptive retrieval.

3.2.1. *Indexing*

To help users build a personal information access map, AHM provides the ability to index markers by user-defined topics. From the adaptive relevance network point of view, a topic can be any word or sequence of words.* The relevance network memorizes the exact set of topics defined by a user for a selected marker, so as to provide accurate retrieval later on. It also generalizes the indexing to subsets of the topics set, in order to facilitate retrieval with similar sets of topics (i.e., sharing common topics). This indexing information is stored in an adaptive relevance network structure, called user profile database in AHM user interface.

3.2.2. *Retrieval*

To facilitate quick access to information, AHM provides the ability to retrieve markers by topics. The user specifies a set of topics as a conjunctive query. The network retrieves a list of markers by exact match, if the exact same set of topics was previously assigned to some markers (memorization); or by derivation from previous queries (memorized or generalized) which contain subsets of the current query.

3.2.3. *Learning*

To help users maintain an accurate information access map, as the information is updated or the user knowledge evolves, AHM offers users the ability to give

* We do not currently represent semantic relationship between topics, but plan to do it in our future work.

feedback in order to modify the relevance of markers to given topics over time (users can always add new topics through the indexing capability). When users give positive or negative feedback to markers retrieved with given topics, the adaptive network automatically adjusts its relevance measures for these markers and topics, and propagates feedback to all proper topics subsets (generalization). The adaptive network also learns to improve its retrieval performance by usage. It automatically analyzes co-occurrence statistics of topics over a collection of previous queries, then selects and memorizes sets of topics with high co-occurrence value to improve the generalization process.

3.2.4. *Collaborative work*

To facilitate sharing of information among a group of users (e.g., flight controllers with the same console position), and to facilitate training of novice users, AHM combines inputs from all users in a separate network, called system profile database (in addition to each user's profile database). Only system-defined topics are used. This supports the creation of a corporate memory over time, which can be shared by all users, or used as a starting point to their individual indexing.

3.3. A SAMPLE SCENARIO

This section describes how users interact with Adaptive HyperMan. An explicit design goal was to change the HyperMan user interface as little as possible, so as not to confuse novice HyperMan users. The only visible difference we made to the book window is the additional *User Profile* menu (Figure 1), which provides access to four new windows: *Categorize Marker* (Figure 2), *Markers Basket* (Figure 3), *Marker Retrieval* and *Marker Retrieval Results* (Figures 4 and 5).

3.3.1. *Assigning topics to markers*

Initially, a user starts by identifying information of interest and assigning topics to it. This is performed by first creating and selecting a marker in the HyperMan book window (Figure 1), then choosing the Categorize Marker option from the User Profile menu. The Categorize Marker window is displayed with the selected marker in the top line (Figure 2). The selected marker is described by its name ("Go to MAL"), book name ("orbops69f1a"), page number ("Page 13"), and internal id number. The user then selects topics from the list of topics on the right and assembles on the left a list of topics to be assigned to the selected marker. Topics are either User Topics, which are user-defined, or Author Topics, which are system-defined and fixed (defined in advance by groups of users to be shared). To assign topics that are not in any of the topics lists, users type them directly into the Enter User Topic field. These new topics will automatically be added to the User Topics list and saved for future use. When the user clicks on the Categorize button, the selected marker is automatically indexed under the list of topics specified by the user, and

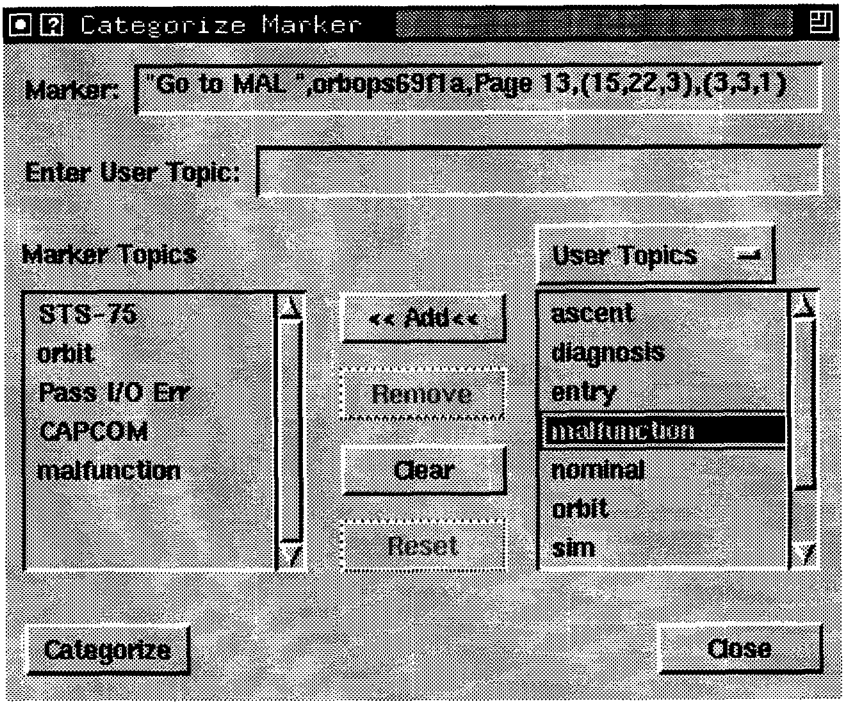


Figure 2. Categorize Marker window. The user selects from the list of topics on the right and assembles on the left a list of topics to be assigned to the marker shown on top.

this customization is stored in both the user profile database and the system profile database.

For example in Figure 2, the highlighted text marker "Go to MAL" is indexed under the user-defined topics "STS–75", "orbit", and "malfunction", corresponding to a mission name, a phase of the mission, and a task of handling a malfunction; and under the system- defined topics "Pass I/O Err" and "CAPCOM", corresponding to a problem description, and a system name.

We will see in Section 3.3.2 that the user can also assign topics to a marker by clicking on the "Success" feedback button (Figure 4b), specifying that the selected marker is relevant to the last topics search.

3.3.2. *Setting aside interesting markers*

After marking interesting information, users might not always have time to categorize markers during a mission, or they might prefer to mark information for a while, then come back and categorize it. For these reasons, AHM offers users the ability to store aside interesting markers, to be indexed or worked on later on. The Markers Basket is a place where users can store a list of markers belonging to any book (Figure 3). This list is personal and saved from one session to the next. To add

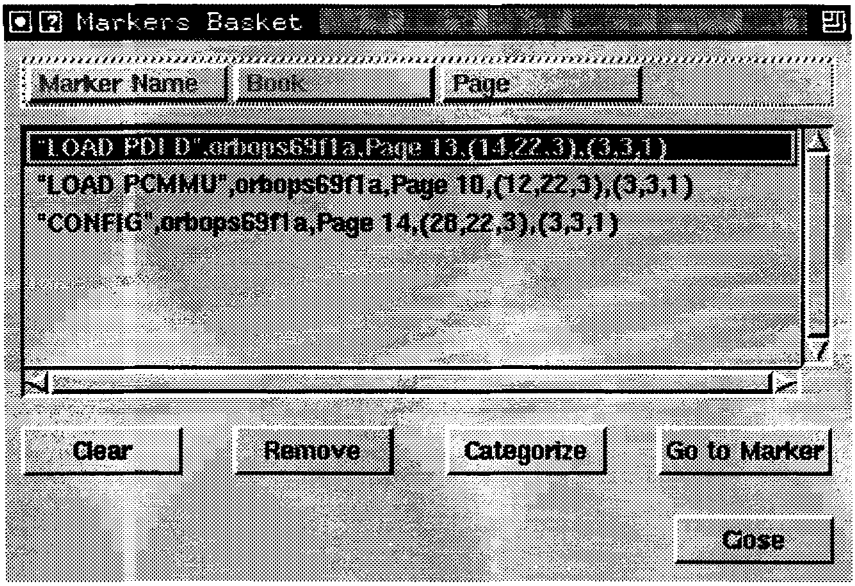


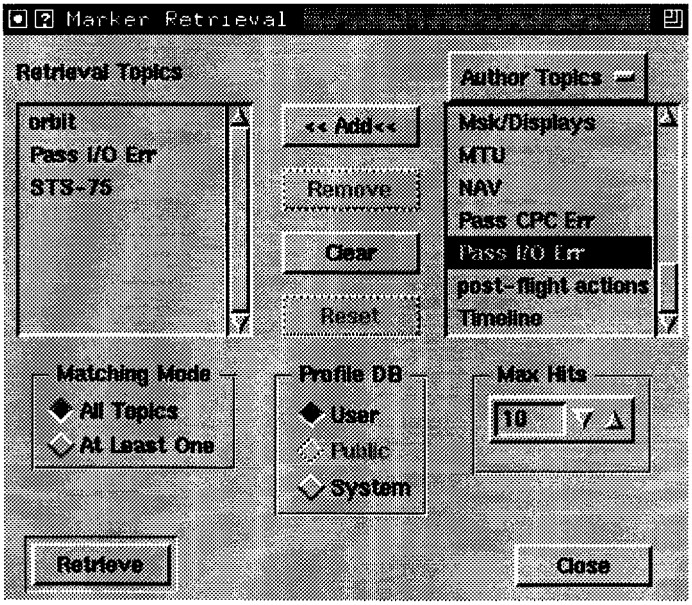
Figure 3. Markers Basket window stores a list of markers for quick access or future indexing.

a marker to this list, a user first selects a marker in the HyperMan book window (Figure 1), then chooses the Markers Basket option from the User Profile menu. To later categorize a marker stored in the list, a user first selects this marker in the list, then clicks on the Categorize button at the bottom (Figure 3): the Categorize Marker window is displayed with the selected marker in the top line (Figure 2).

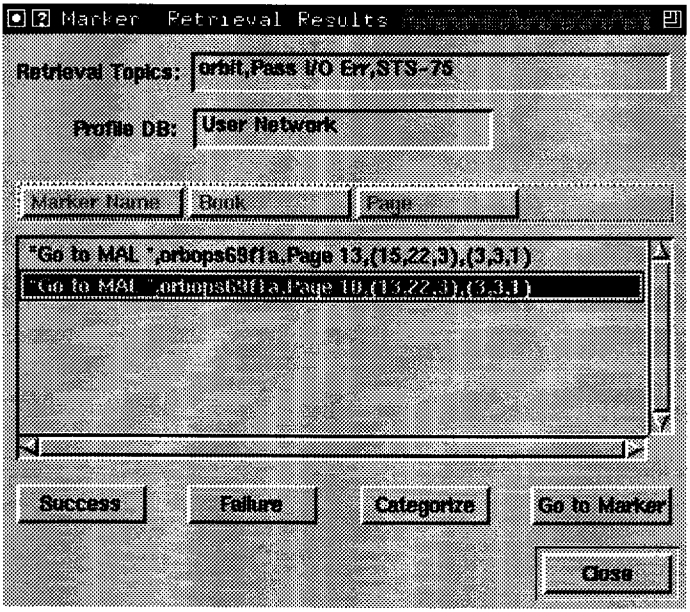
The Markers Basket can also be used as a quick access tool, like Bookmarks in Netscape or the Hotlist in Mosaic. Users can directly go to markers attached to specific pages in books, by simply double-clicking on a marker in the Basket, or selecting it and clicking on the Go to Marker button.

3.3.3. *Retrieving markers by topics*

Once a user has categorized a few markers, he/she can search all books in the current library for markers relevant to specific topics. The user opens the Marker Retrieval window (Figure 4a) from the User Profile menu. The user assembles a list of topics describing the markers they are looking for. This window gives several options specifying how to perform the retrieval. The matching mode option specifies whether to retrieve markers that have been assigned every topic in the list or those that include at least one topics. The profile database option specifies whether to consider input learned from all users relevant to the search (System), or only input which has been learned from that particular user. The user can also specify the maximum number of markers to be retrieved. For example in Figure 4a,



(a)



(b)

Figure 4. Marker Retrieval and Marker Retrieval Results windows. The user assembles a list of topics describing the markers they are looking for (a), and the system displays a ranked list of markers relevant to the retrieval topics (b).

the user specifies to retrieve markers he/she personally indexed under all three topics “orbit”, “Pass I/O Err”, and “STS–75”.

After the user submits the search, the system displays a ranked list of retrieved markers in the Marker Retrieval Results window (Figure 4b). The list of markers is ranked by relative relevance to retrieval topics (most relevant first). In this example, only two markers were indexed under all three topics. However these markers were originally indexed under a more specific set of topics (marker shown in Figure 2), and the system used a derivation algorithm to retrieve these markers under a more general set of topics (see Section 4.4.2).

The user can directly go to a retrieved marker in a book by double-clicking on it, or by selecting it and clicking on the Go to Marker button. Feedback buttons (Success and Failure) are used to adjust the ranking of a particular marker, which will move the marker up/down the list. The system automatically assigns the current retrieval topics to the selected marker, and adjust it relevance measures. The user can also associate additional topics to a marker using the Categorize button, which displays the Categorize Marker with the selected marker.

Indexing and retrieving markers by topics lets flight controllers create personal collections of markers. A personal collection corresponds to a list of markers retrieved under specific topics in the Marker Retrieval Results window. This is why flight controllers call it a “virtual goody book”. This list is dynamically computed, and always kept up to date. Moreover, flight controllers can create markers for multiple purposes: quick access, highlighting critical information, writing down comments during a mission. Therefore being able to index markers under multiple topics is very valuable, as it gives them capabilities to retrieve markers for a given flight or simulation, for a particular problem, for a particular system, and to combine these topics to narrow down the list of markers.

3.3.4. *Providing feedback*

We illustrate more sophisticated capabilities to customize information access in the following example. After categorizing more markers, the user decides to retrieve all markers relevant to “orbit” and “STS–75” (Figure 5a), which is more general than the previous query (Figure 4a). The user has previously categorize the markers “Go to MAL, Page 10”, and “Go to MAL, Page 13” with the set of topics “STS–75”, “orbit”, “malfunction”, “Pass I/O Err” and “CAPCOM”. Because of its generalization algorithm, the relevance network has also propagated the relevance of these two markers to each individual topic. Therefore the system retrieves these two markers, together with five other relevant markers, when the user submits the query “orbit” and “STS–75”. The user however decides that these two markers are not relevant for this general query, since they correspond to very specific malfunctions. The user decides to tell the system about this exception by selecting each marker and clicking on the Failure button. As a result, the system first moves them to the bottom of the retrieval list. The user continues to click on the Failure button until the system lower their relevance weight below the minimal threshold (see Section 4.5.4), and removes these two markers from the retrieval list (Figure 5b). The two markers will still be retrieved by the query "orbit", "Pass I/O Err", and "STS–75" as in Figure 4, but won't be retrieved by the query "orbit" and "STS–75" anymore. This example illustrates the non- linear aspect of the Adaptive Relevance Network.

3.4. Collaborative Review

Most markers are created by flight controllers prior to a mission, but they also write down comments during a mission whenever a problem is encountered. These comments, also called post-flight actions, are later used in collaborative reviews to revise procedures and update documents. With Adaptive HyperMan, flight controllers now have the ability to categorize their comments under system-defined topics, for example "post-flight actions". After a mission and in preparation of a collaborative review, they can then access all annotated pages indexed under "post-flight actions" by other flight controllers, by retrieving markers from the system profile database which integrates input from all users.

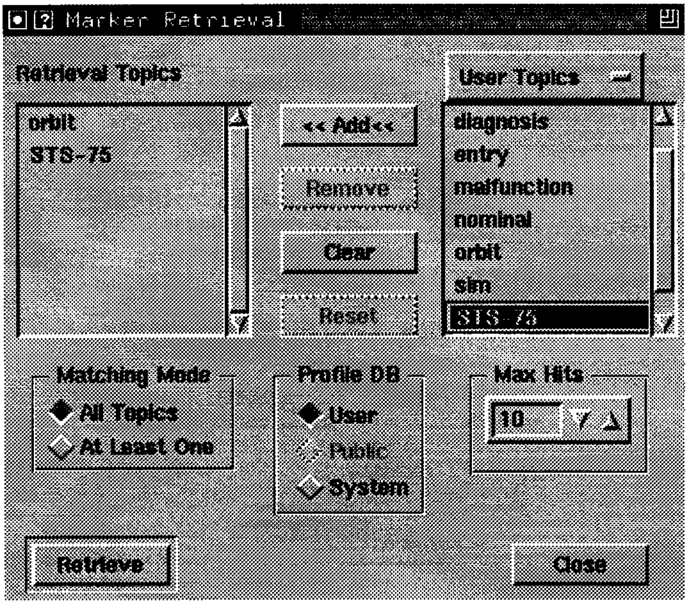
In the following section, we describe the structure of the relevance network, its learning method based on user feedback, and its information retrieval method. We finish the section with a discussion of the efficiency and computational cost of the relevance network.

4. Adaptive Relevance Network

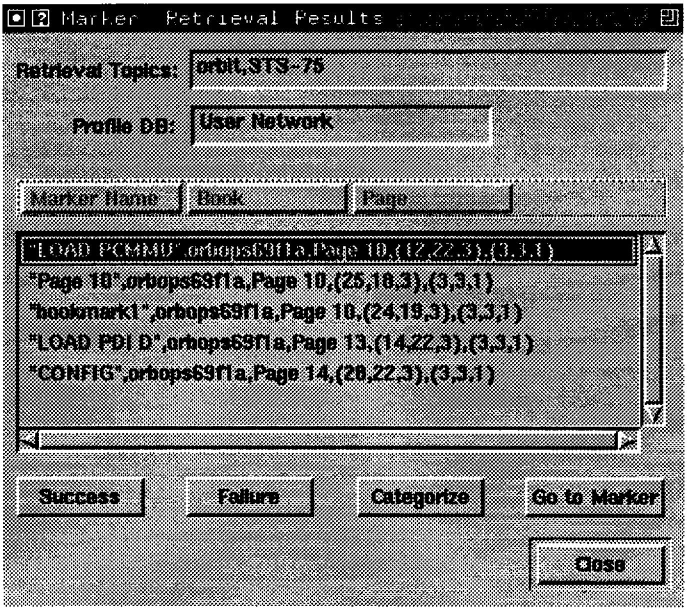
An adaptive relevance network models user preferences on information relevance with respect to given tasks. This network provides a domain independent information architecture which facilitates incremental storage of both relevance information provided by users, and relevance information computed through other traditional retrieval techniques. The network memorizes information on the relevance of references based on user feedback for specific queries. It also aggregates and generalizes such information to facilitate future retrievals with similar queries.

4.1. Relevance Network

A relevance network records measures of relevance of output nodes with respect to input nodes. For information retrieval purposes, an *output node* corresponds to a reference, which can be a document or any marked location within a document. There are two types of input nodes: *basic nodes* and *composite nodes*. A basic input node corresponds to a descriptor. A descriptor can be a keyword in the index, a sequence of words in the text, or a user-defined task or goal. A descriptor can also be a reference, to retrieve other related references. A composite input node corresponds to a combination of query descriptors. Composite nodes are defined in Section 4.2.



(a)



(b)

Figure 5. Providing Feedback. The user issues a fairly general query (a), and gets rid of non-relevant retrieved markers by giving negative feedback with the Failure button, until unwanted markers fall to the bottom of the list, or even disappear from the list (b).

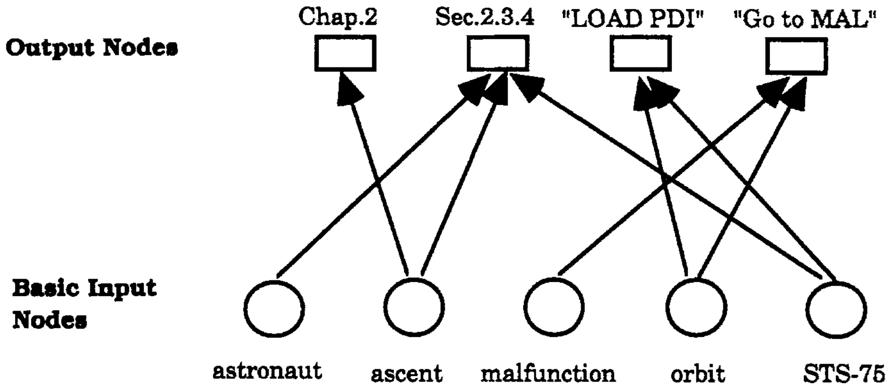


Figure 6. An example of a simple relevance network.

Figure 6 shows an example of a simple relevance network with only basic input nodes. Nodes in the top layer represent output nodes. Nodes in the lower layer represent input nodes. A user query is interpreted as an input activation pattern by the relevance network. A Boolean activation value of an input node denotes whether the corresponding descriptor is a member of the current query. An activation value on an output node denotes the relevance of the corresponding reference, conditioned by the current user query encoded in the input layer.

Associated with each connection from an input node to an output node is a relevance measure between the corresponding descriptor and reference. A network is initially empty.* As a user specifies queries and provides positive or negative feedback on the relevance of retrieved references, input and output nodes that do not exist yet in the network are created, and relevance measures associated with the connections are adjusted accordingly. A relevance measure, in its simplest form, is defined as the relative frequency of positive user feedback for a reference given a descriptor. Each relevance measure is maintained as two parts of a fraction: the number of positive feedback, S, over the number of total feedback, N. That is, a relevance measure R_{ij} of a reference j with respect to a descriptor i is

$$R_{ij} = \frac{S_{ij}}{N_{ij}} = \frac{\text{Number of (Positive Feedback)}}{\text{Number of (Feedback)}} .$$

Maintaining the total number of feedback in the denominator facilitates an accurate recording of both the relevance of the reference and the sampling precision of such relevance.

A relevance network can be built incrementally entirely through user feedback, starting from an empty network without any node and connection. For pragmatic purpose, however, a relevance network is often initialized with known relevance

* When the relevance network is empty, references relevant to a user query can be accessed through other retrieval means provided by the application interface. A relevance network can also be initialized using traditional information retrieval techniques.

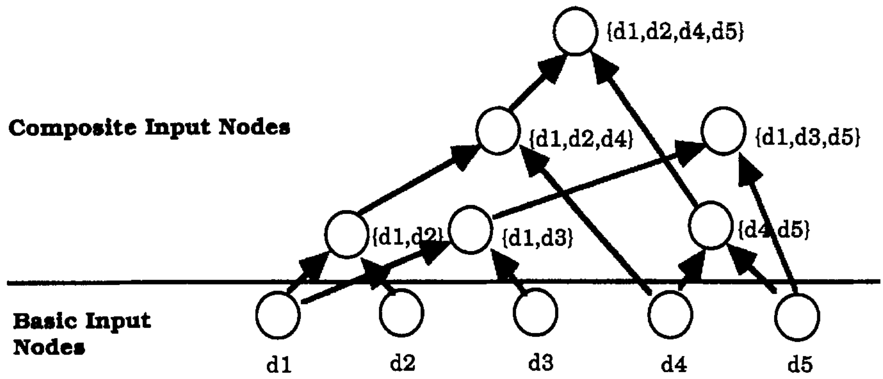


Figure 7. Input layer of a relevance network with composite nodes. Arrows denote subset relations. Output nodes and relevance connections are not displayed.

information. This can be done through various means. A user can borrow a personal relevance network from another user, or a shared network from a group, to create his/her own initial network. Information obtained from simple probabilistic indexing models can be used directly to initialize nodes and connections. For more sophisticated probabilistic or inference models, however, specific conversion algorithm will be needed to assign appropriate relevance measures to initial connections. Another way to initialize a relevance network is to submit a training set of queries with corresponding result references to the system. In this case the training set will be simulated as user feedback to derive appropriate relevance connections.

4.2. COMPOSITE NODES

The relevance model described thus far records only relevance information based on single descriptors. User preferences for particular composite queries cannot be saved, and non-trivial relations* between references and descriptors cannot be encoded. To retain better information from user feedback, the relevance network accommodates composite nodes in the input layer, as illustrated in Figure 7. It is assumed that the relevance information associated with a composite node is more specific than the information associated with its nested subset nodes or its basic input nodes (corresponding to its descriptors). Therefore, during retrieval, a user query is first matched against highest level composite nodes, rather than lower level nodes. The use of composite nodes enables the system to derive more accurate relevance measures learned from previous queries.

Composite nodes are added to the network in two ways. A new composite node, corresponding to a user query, is added to the relevance network when a user provides feedback upon retrieval. A second composite node addition method based on co-occurrence statistics of query descriptors is discussed in Section 4.5.3.

* E.g., a reference relates to {Apple, Computer} but does not relate to {Apple}.

4.3. Learning Relevance Measures from User Feedback

When a user provides positive or negative feedback for a reference given the current query, this relevance information is memorized and generalized. To memorize feedback information on specific queries, relevance of the connection between the reference and the composite node corresponding to the query is updated. If such a connection does not already exist, a new connection is created and the relevance measure initialized.* If a composite node corresponding to the query does not exist, a new composite is created with associated relevance information derived from that of its components. The derivation algorithm is described in Section 4.4.2.

To derive generalized relevance measures for new queries in the future, nodes which are more general than the user query inherit feedback: relevance measures from all proper query subsets including basic input nodes are updated. We describe below how relevance measures are updated or initialized.

4.3.1. *Updating relevance measures*

As mentioned above, relevance measures from an input node can be adjusted either through direct user feedback from a query of the same composition as that node, or through feedback inherited from its superset composite nodes. Direct feedback provides more accurate information pertaining to the node than inherited feedback. To compromise between memorization and generalization, a weight constant integer $C >= 1$ is added to the relevance feedback adjustment: if $C = 1$, inherited feedback is as important as direct feedback; and the relative importance of inherited feedback decreases when C increases. Relevance measures are updated as follows:

$$R_{new} = \frac{S_{new}}{N_{new}} = \frac{S_{old} + \lambda * \delta}{N_{old} + \lambda}, \quad \text{where}$$

$$\delta = \begin{cases} 1 & \text{for positive feedback} \\ 0 & \text{for negative feedback} \end{cases}, \qquad \lambda = \begin{cases} 1 & \text{for inherited feedback} \\ C & \text{for direct feedback} \end{cases}.$$

As mentioned in Section 4.1, maintaining relevance measure as a fraction provides additional information on the precision of the measure. To accommodate more recent changes into the relevance network, a maximal threshold on the denominator is specified. When the number of total feedback exceeds this threshold, the denominator is no longer incremented, instead, a momentum term is used in the calculation:

$$R_{new} = \frac{S_{new}}{N_{new}} = \frac{S_{old}}{N_{max}} * \alpha + \delta * (1 - \alpha),$$

where α is the momentum, $0 \leq \alpha \leq 1$, and N_{max} the maximal threshold

* The user can choose to give feedback on any reference s/he has access to (not only on these references retrieved from the network), thus automatically indexing this reference with the current set of query descriptors.

for number of total feedback. Since N_{new} must equal N_{max} in this case, $S_{new} = S_{old} * \alpha + \delta * (1 - \alpha) * N_{max}$.

For consistency with the relevance adjustment formula where the denominator is smaller than the maximum threshold, the momentum is typically set accordingly as: $\alpha = (N_{max} - \lambda)/N_{max}$.

4.3.2. *Initializing relevance measures*

Relevance measure for a new connection is initialized with $S_{old} = 0$, $N_{old} = K$. K, a positive integer, corresponds to an initial negative bias. With this initial bias, the relevance measure asymptotically approaches one with the increase of the rate of positive feedback. The scale of relevance thereby provides better resolution for positive relevance information, and is biased against relevance measures with lower feedback frequency, which is assumed to indicate lower confidence of relevance accuracy.

4.4. Retrieval of Relevant References

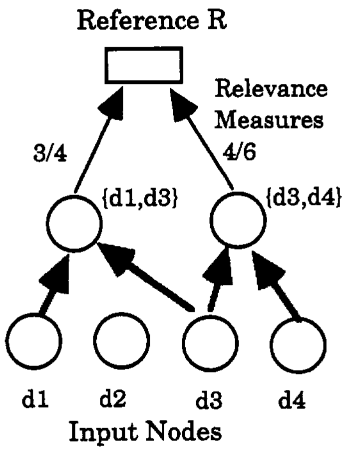
In response to a query, the network retrieves and displays a list of references with the highest relevance measures. A query is assumed to be semantically constructed as a conjunction of member descriptors. Relevance measures are derived from nodes most specifically related to the query. Equal importance is given to all query descriptors. And for each descriptor, relevance measures from composite nodes are pooled together according to their statistical accuracy. This retrieval algorithm is described in the following two subsections.

4.4.1. *Retrieval by exact match*

Upon presentation of a query, if a composite node corresponding to the query already exists in the network, the relevance measures from that node to associated references are directly used to generate a ranked list of relevant references.

4.4.2. *Retrieval by derivation*

If a matching composite node does not exist, relevance measures from other input nodes, corresponding to proper subsets of the query, are used to derive a list of relevant references. For a relevance network with only basic input nodes, a simplistic estimate of relevance of a reference with respect to a query can be taken as the product of the relevance measures of that reference with respect to the descriptors of the query. The use of multiplicative estimation assumes no weighting information among individual query descriptors, and gives higher relevance to references of uniform relevance to all query descriptors.



Query Q = {d1,d3,d4}

Pooled estimates of query descriptors:

$PE_{d1} = 3/4$

$PE_{d3} = (3+4)/(4+6) = 7/10$

$PE_{d4} = 4/6$

Derived relevance of R for query Q:

$Rel_Q = (3/4 * 7/10 * 4/6)^{1/3}$

Figure 8. An example of retrieval by derivation from subset query nodes for one reference.

With the presence of composite nodes corresponding to query subsets, relevance information is derived only from the top-level subsets, i.e., the ones not nested within other subsets. However, relevance measures for a query cannot be appropriately obtained by simply taking the product of top-level subsets' relevance measures. Top-level subsets of a query can be of different sizes, and are not necessarily disjoint. References connected from one composite node may not be connected from another. Multiplicative measures, therefore, can be biased toward certain query components. We propose a heuristic derivation algorithm intended to provide impartial relevance estimations. For simplicity, we describe the relevance derivation algorithm for a single reference R, hence subscript reference indices are neglected in the formulae that follow.

For each descriptor in a query, a pooled estimate (i.e., an average adjusted by the sample sizes of individual estimates) of the relevance between that descriptor and a reference, is obtained from relevance measures associated with the top-level subsets of the query which contain that descriptor. Let $Comp_j$ denote the set of descriptors in a composite index J, and S_{top} the set of top-level subsets of a query Q. The pooled estimate of relevance for a descriptor d_i in Q is

$$PE_i = \frac{\sum_j S_j}{\sum_j N_j}, \quad \text{where the sums are taken over all } j \text{ where } Comp_j \in S_{top}$$
and $d_i \in Comp_j$.

Relevance measure of reference R for the query Q is then computed as

$$Rel_Q = \sqrt[n]{\prod_i PE_i}, \quad \text{where } d_i \in Q, n \text{ is the size of } Q.$$

The derived relevance measures are then used to generate an ordered list of suggested references. An example of relevance derivation from query subsets is shown in Figure 8.

In a query derivation where no subset composite node is present in the network, the algorithm degenerates to simple multiplicative derivation over basic input nodes. When applied to a query with disjoint top-level subsets, the algorithm reduces to taking the root of the product of all top-level subsets, each to the power of its own size.

The use of multiplicative derivation of relevance described above is based on the design choice for our current application, that queries are semantically constructed as conjunction of member descriptors. This choice gives priority of relevance ranking to references more equally related to all components of a query. Other alternative aggregate functions can be used if deemed useful in future study. The relevance ratios associated with connections can be considered as probabilistic estimates, and can be combined with different Boolean operators. If weights associated with individual query descriptors are available, a weighted-sum of probabilities associated with descriptors can be used to estimate the probability of query-reference relevance.

4.4.3. *Partial match derivation*

The current implementation also supports retrieval with partial match of query descriptors, i.e., retrieving references which are indexed by some, but not all of the query descriptors. Users are given control of the level of partial match, i.e., the minimal number of query descriptors a suggested output reference must be related to. Partial match retrieval is similar to retrieval with a query of disjunctive descriptors. However, relevance estimates of partial match output references are still derived multiplicatively, to ensure consistent rank order display relative to full match results. A *missing relevance* value is used in the multiplicative derivation formula, as an estimate of relevance associated with a missing connection from a query descriptor to a reference. This missing relevance has very similar connotation as the default belief used by the inference network-based model in (Turtle & Croft, 1991). However, since the relevance network only deals with personal indexing independent of text information within the documents, our missing value is just a constant parameter which can be adjusted for better performance. In our current implementation, default value of the missing relevance is set at $1/2K$, where K is the constant initial bias discussed in Section 4.3.2. The value of missing relevance is also used in connection-trimming discussed in Section 4.5.4.

4.5. Managing Network Size and Capacity

4.5.1. *Computing cost and the importance of capacity management*

The primary cost of computing time in the use of an adaptive relevance network is associated with the relevance derivation algorithm, which requires a search of composite nodes corresponding to all proper query subsets. In theory, this search can be computationally exponential with respect to the size of the query, due to the combinatorial large number of possible top level subsets. For pragmatic information retrieval purposes, however, the number of descriptors in a query is usually small, and only a very small percentage of all possible combinations of query descriptors is likely to be present in the network as subset composite nodes. Also, this cost of computing time is in the worst case linearly bound by the total number of composite nodes in the network. Thus the computational complexity of the derivation algorithm is not of realistic concern, provided that the number of composite nodes and the distribution of these nodes are well managed.

While the memorization capacity of a network increases with the number of composite nodes it contains, unnecessary composite nodes can potentially inhibit generalization of retrieval. Higher level large composites carry relevance information more specific to particular queries, whereas lower level smaller composites carry more general feedback information propagated from many queries. Relevance measures associated with larger composites, however, may also have less statistical accuracy since these composites receive less feedback from users. Managing the network capacity is therefore not only important in assuring control of the computing cost, but also important in maintaining a balance between the capacity of memorization and that of generalization.

4.5.2. *Cutting composite nodes*

A node cutting procedure is employed to control the size of an adaptive relevance network. A maximal number of composite nodes allowed in a network can be specified. When a new composite node needs to be inserted, and if the network has reached its specified size limit, an existing composite node with the least *frequency of usage* is removed to make room for the new one. The frequency of usage of a composite node is calculated by recording the number of times the node is used in query execution. In addition, it is also incremented when the composite node receives direct user feedback. The purpose of this feedback-based usage update is to give more weight to composite nodes which carry information that cannot be easily derived.

A portion (currently set at 20%) of the composite nodes most recently added to the network are left in a queue, excluded from the candidate list of nodes used by the cutting procedure described above. This ensures that a new composite node will have ample chance to accumulate usage statistics, thereby proving its usefulness. We are also planning to apply a decay formula on the frequency of usage to all composite nodes at regular intervals, so that previously useful composite indices that have become obsolete over time can be replaced.

The frequency of usage is a direct measure of how often a node is used for user queries. More subtly, it also serves as a measure of the *confidence level* of information accuracy associated with a node. The choice of removing nodes with the least frequency of usage ensures that the composites that remain in the network are the ones with most dependable relevance information.

Once a composite index node is removed, the relevance information it carries cannot be recovered. However, since much of the feedback information associated with this node had been propagated down to its lower level sub-composites and basic descriptors, only the information unique to this composite is lost.

4.5.3. *Adding composite nodes*

The query-based composite node creation method described in Section 4.2 is intended to ensure quick learning of user preference by memorizing relevance information. Ideally, these nodes would also become useful components of the relevance network information structure, to derive relevance information for new queries. Unfortunately, smaller composites are less likely to enter the network since the relevance information they are associated with may be too general for them to be used as specific queries by users. Yet these smaller compositions may be of great importance for a network to assure effective encoding of relevance information.

A node creation method based on co-occurrence of query descriptors is devised to extract compositions important to the relevance network information structure. The network maintains a record of recently submitted user queries, and periodically generates statistics on sets of descriptors that often appear together in different queries. A simple formula is currently used to calculate co-occurrence statistics of query descriptors over a collection of queries:

$$C_s = \frac{f_s}{\text{size\_of}\,(s)\sqrt{\prod_{i \in s} f_i}},$$

where C_s denotes the co-occurrence measure of descriptors in set S,

f_s denotes the frequence of set S appearing in queries, and

f_i the frequency of descriptor i in queries.

Composite nodes consisting of query descriptors of high co-occurrence statistics are then automatically added to the relevance network. Composite creation based on co-occurrence across queries facilitates effective encoding of non-trivial relevance information. It also helps generalize relevance information for future retrieval with similar queries.

4.5.4. *Trimming connections and the scale of relevance*

Another measure of the network size is the number of relevance connections from input nodes to output references. Relevance connections are indexed in a database by the nodes they are associated with, and only the ones related to a query need to be retrieved at a time. Unnecessary connections cause wasteful storage space, and can impact the performance of retrieval.

A connection with relevance value less than or equal to the missing relevance (described in Section 4.4.3), as a result of frequent negative feedback, is removed from the network. This connection *trimming* process prevents the network from unlimited addition of connections, and from keeping wasteful information of very low relevance. The relevance measure maintained by the network is therefore on a rational scale between the missing relevance and one, and it is non-linearly proportional to the ratio of positive feedback. The use of a positive scale does not deprive the network of its capability of encoding negative relevance information. By trimming relevance connections from a composite more specific to a query, i.e., a query subset composite node of larger size, positive information carried by more general, smaller subset composites will be ignored. Thus the effect of negation is supported by the dynamic architecture of the adaptive relevance network.

5. Experimental Results

The adaptive relevance network is designed to model subjective indexing based on user preference of information access. It is intended as an information framework which integrates indexing structure provided by users, with indexing information generated by other conventional indexing methods and/or retrieval methods specific to the domain of application. For application purpose, it has been designed to be incorporated into large-scale, complex information systems. It is therefore difficult to test the full functionality of the network independent of the application domain. As a first step, we focused on the validity of the proposed model and report on experimental results of the memorization capacity and generalization ability of the adaptive relevance network, with no attempt to simulate user behavior. Real-world usage study of Adaptive HyperMan by flight controllers is in progress.

5.1. SIMULATION SETUP

We used two test data sets of information retrieval from the SMART archive at the Computer Science Department of Cornell University. The first is a collection of 1963 Time Magazine news articles which consists of 425 articles and 80 queries. The second is the ISI collection of most highly cited articles and manuscripts in information science in the 1969–1977 period, with 981 articles and 76 queries. These experimental data sets were originally devised for the investigation of automatic indexing and document retrieval methods. Queries of the two sets employ large vocabularies, and different queries share very few similarities. These are therefore not ideal for the testing of adaptive indexing, where higher similarities among queries more specific to individual users and/or task domains, as well as non-trivial relevance structures between references and queries are expected. The collections, nevertheless, were used here in our simulation experiments to ensure objective evaluation of the adaptive relevance network.

Queries in these data sets are composed of common English phrases, e.g., "United Nation's efforts to get Portugal to free its African colonies". For our purpose, the queries are edited into sets of keywords with simplistic stemming, and common English words removed. Thus the above query becomes "unite, nation, portugal, africa, colony". The particular sequential order of words in a query is not utilized.

In simulation, a query is presented to the network as a set of descriptors, and the references retrieved by the network are compared with the target references listed in the original data set. Positive feedback is simulated for references that are in the target list but are not suggested by the network. Similarly, negative feedback is given to the network for suggested references not in the original data set. Although the adaptive relevance network is designed to accommodate other means of retrieval, all simulation trials were conducted with initially empty network, to demonstrate clearly the functionality of the adaptive engine.

It should be noted that the feedback simulation method of our study is very different from that of conventional information retrieval experiments. In our simulation, feedback information is not only given for references retrieved from the system, but also given for new references to be added to the system. The relevance network is designed as a personal index system to organize known information, not to retrieve unknown information. The experiments reported here are intended to verify the validity of the proposed indexing architecture and algorithm, not to measure real-world application performance.

We first tested the memorization capacity of the adaptive relevance network, i.e., the amount of relevance information a network can memorize with respect to the number of composite nodes. We then tested the generalization capability, i.e., the ability of a trained network to derive and suggest references for queries not previously presented to the network.

5.2. Memorization capacity

The Time collection was first used to test the memorization capacity. Each query in this set should retrieve from one to 18 references. Query-based insertion of composite nodes was first disabled. Consequently the network did not contain any composite nodes hence could only encode and derive relevance information with the basic descriptors. This network was trained with the complete set of queries in random order. For each query, positive feedback was given for all relevant references not retrieved, and negative feedback given for irrelevant retrieval of references not in the target set. After one complete cycle of training, i.e., each

Table I. Comparison of ISI data set retrieval results with different numbers of co-occurrence based composite nodes. 100% recall of 2655 relevant references were attained in all cases.

Number of composites	Average precision in %		Total number of irrelevant references	
	1 cycle	3 cycles	1 cycle	3 cycles
0	78.1	78.1	2176	2176
23	85.2	86.1	1142	964
41	88.7	88.9	716	694
81	87.7	88.9	664	512
179	90.8	92.6	402	298

of the queries presented once, the relevance network was able to retrieve with 100% recall,* at a precision** of 93.1%. Specifically, all 321 relevant references, along with 37 irrelevant ones were retrieved. This result suggested that the data set is largely linear (i.e., relevance information associated with a query can be derived from relevance associated with its member descriptors), hence the addition of composite nodes could make limited retrieval enhancement. Without enabling the query-based composite node insertion algorithm, five composite nodes of the highest co-occurrence statistics among the queries were added to the initial network. The modified network was able to improve the precision slightly to 94.8%, with 34 irrelevant retrievals, without affecting the 100% recall. The performance could not be improved further with more composites of lower co-occurrences added.

With the query-based composite insertion enabled, the network achieved perfect performance of 100% recall and precision, by memorizing the target references with 80 composites corresponding to the query set. When the composite node cutting procedure is in effect, the network was able to maintain perfect performance with as few as 10 composites.

Similar studies were done with the ISI data collection. To better demonstrate the network's capacity to encode non-trivial information, for this data set we eliminated query descriptors which appear in only one single query. Two of the 76 queries were invalidated consequently, as they became empty. The resulting set consists of queries of sizes ranging from 2 to 15 descriptors. Each query is to retrieve from 3 to 125 relevant references.

In simulation runs using the ISI data set, different numbers of composite nodes based on levels of co-occurrence statistics were added initially. The results are shown in Table I. Precision and recall statistics were collected after 1 and 3 training cycles. 100% recalls were attained for all simulation runs. The network with 41 composite nodes had higher total number of irrelevant references, yet better average precision than the network with 81 composites. This is because an average precision

* Recall is defined as the proportion of relevant materials retrieved.

** Precision is defined as the proportion of retrieved materials that are relevant.

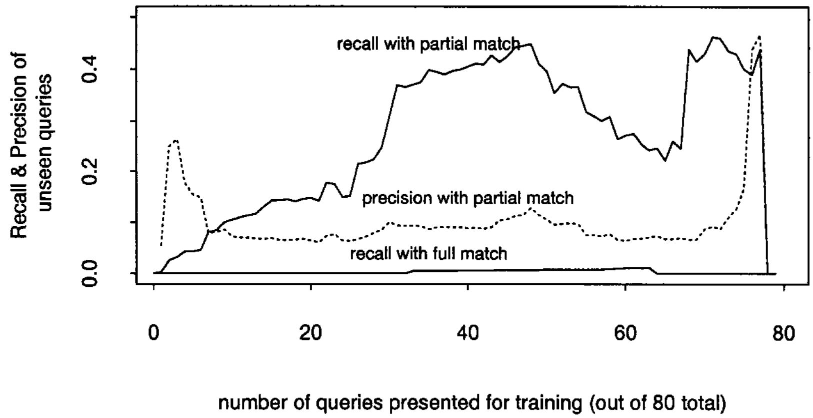


Figure 9. Generalized retrieval results with unseen queries from the Time Magazine data set.

is taken over the precision measures of all queries, which is not the same as a pooled-average calculated directly from the total numbers of relevant and irrelevant references.

5.3. Generalization

We first tested the generalized retrieval capability of the network on the Time collection data. Since this data set is largely linear, no composite nodes were employed. Simulation tests were conducted in both full match mode and partial match mode (described in Section 4.4.3). Queries from the set are presented to the network one at a time for feedback simulation. At the end of each query simulation, the remaining queries in the set not yet presented to the network, were used to test the network's retrieval performance. The results are plotted in Figure 9. With full match only, the average generalized recall was only near 1%. This was not surprising since many queries in this data set contained unique descriptors. The precision was not available since for many queries no reference was retrieved. With partial match, the recall increased to 40% with half of the queries presented. Recall statistics had wider variations at the end of the curve, as the sample size of test queries became smaller. The average precision in partial match retrieval remained mostly stable at around 10%.

We then tested the ISI data set for generalized retrieval in partial match mode. The curve of generalized recall was similar to that of the Time data set, with a peak recall at 57.5%. Precision stayed low at around 10%. To see the effect of composites on generalization, we ran another test with 41 composites of high co-occurrences inserted to the network after 40 of the 74 queries were trained, and the generalization test continued afterward. The generalized recall performance in this case showed consistent improvement as training continued, with a peak recall of 86% at the end. Figure 10 shows the curves of generalized recall with and without

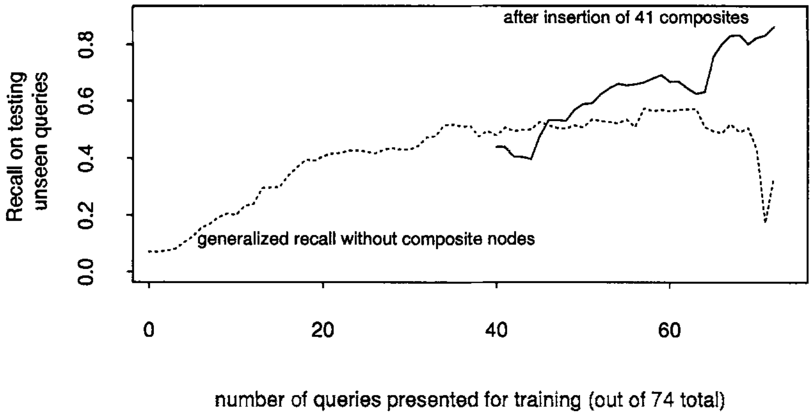


Figure 10. Generalized recall with unseen queries from the ISI data, with and without added composite nodes.

added composite nodes. The low recall rate toward the end of the dashed-curve was partially caused by chance since only few unseen queries were left for testing.

5.4. Discussion

Simulation tests have shown that with query-based composite insertion, an adaptive relevance network is capable of perfectly memorizing relevance information based on user feedback. Test results also suggested that, while a network without composite nodes cannot maintain good precision of retrieval for data sets which contain non-trivial relevance information, the precision can be significantly improved with the addition of only a small collection of composite nodes. Composite nodes help improve precision through the provision of more specific information in relevance derivation. In a real-world application, the query-based composite insertion facilitates the customization of relevance information for specific queries of frequent usage, whereas the co-occurrence-based composite insertion helps the establishment of efficient information structure for long-term usage. These two composition methods, together with the node cutting procedure, work like a genetic algorithm that governs the evolution of the relevance network architecture.

We have shown also in simulation that the network is capable of generalizing retrieval of relevant references for queries not previously seen, through its feedback propagation algorithm. Generalized recall is further enhanced with the addition of composite nodes, which helps direct feedback information to appropriate composite structures, thereby releasing capacity of other parts of the network to encode more accurate relevance information.

Simulation with partial match also incurred low precision for generalized retrieval. This is partially due to the wide variation of relevance information of the test queries. In addition, a complete list of references retrieved in partial match mode carries much additional relevance information, hence inhibits high precision. In practice, users are given flexible control of the amount of information displayed. Lastly, for simulation purpose, relevance information in a network was not initialized. The adaptive relevance network, for application purpose, should be initialized with traditional or other domain specific indexing retrieval information.

To summarize, the adaptive relevance network provides an information structure that facilitates integration of domain specific document indexing information and subjective user preferences. The adaptive architecture of a network, with associated relevance connections, supports a balance between customization and generalization. The control of balance between precision and recall is given to the users.

6. Future Work

A more sophisticated kind of adaptive hyperlinks based on previous path of traversal has been proposed in (Chen & Mathé, 1994). Instead of taking concept descriptors as input, the hyperlink traversal path to a current hypernode is defined as input to the system, and the filtered hyperlinks available from the current node is the suggested output. Like in the adaptive relevance network, composite nodes are used to construct relevance structure of adapted hyperlink information.

The current approach of network capacity management, in particular the composite node cutting procedure, is entirely usage based. An interesting direction of future research is to explore dependency among composite nodes in terms of their associated relevance information, and to utilize such information in the addition and deletion of composite nodes.

The employment of composite nodes to encode non-linear relevance information, like the use of hidden units in artificial neural networks, has the advantage of learning without assuming specific knowledge structure. On the other hand, given the vast complexity of information retrieval, it is likely that the incorporation of knowledge-based components into the network can greatly enhance pragmatic retrieval performance. One possible approach is to translate user queries with a knowledge-based system, into a set of internal input descriptors used exclusively by the relevance network. Another approach described in (Katsumoto et al., 1995) uses a user model to convert a user query to an "average user" query, which is then used by a knowledge base on textile design to retrieve relevant images.

In order to enable sharing of adaptations among a large number of users, it would be useful to have a common vocabulary for describing user-defined concepts. This could be done by acquiring a domain model and enforcing a fixed vocabulary on users; or we could learn this common vocabulary by applying machine learning techniques to the set of concepts defined by users after a period of time. Another extension to promote collaboration among users is to automatically find all users with similar concepts, in order to suggest them new documents found relevant by others with similar interests (Lashkari et al., 1994).

We plan to enhance the Adaptive HyperMan system in two major directions. First we will support collaborative work by providing a publish/subscribe mechanism to develop users group indices for markers. Second, we are studying the idea of a virtual goody book being displayed as a book (with pages to flip) instead of a list of markers, and of providing an authoring facility to let users edit this virtual book (adding their own new pages, or reordering pages).

Finally, we have been collaborating with Boeing on integrating the adaptive relevance network into the Portable Maintenance Aid prototype for airplane mechanics (Bradshaw et al., 1993; Bradshaw et al., 1996). We also integrated the adaptive relevance network into the World Wide Web for organizing and sharing personal pages collections. These applications are currently at the prototyping and testing stage.

7. Conclusion

We have presented a user-centered indexing approach to adaptive hypermedia and information retrieval. The adaptive relevance network technique supports users in creating and managing personalized information access maps, which are organized by concepts. This approach is extremely useful for technical information access tasks characterized by a large volume of hypermedia connected documents, the need for rapid and effective access to already known information, and long-term interaction with evolving information. We demonstrated adaptive retrieval and navigation with the Adaptive HyperMan system. The system provides sophisticated marking and hyperlinking capabilities to end-users, and allows them to assign user-defined and shared topics to markers, retrieve markers by topics, and provide feedback over time. We described the Adaptive HyperMan system, its user interface, and showed how it provides a virtual "goody book" facility for Space Shuttle flight controllers. The system also supports collaborative review by letting users share group access maps.

We then presented the adaptive relevance network which creates and maintains a complex indexing structure to represent personal user's information access maps. The model employs a simple adaptive algorithm embedded in a dynamic indexing architecture based on user feedback. It does not require any a priori specialized index structure, nor any a priori statistical knowledge or computation. We have shown that with query-based composite insertion, a relevance network with a limited capacity is capable of perfect memorization of relevance information based on user feedback. We have shown also that through its feedback propagation algorithm, a network is capable of generalizing retrieval of relevant references for queries not previously seen. While the query-based composite insertion facilitates the customization of relevance information for specific queries of frequent usage, the co-occurrence-based composite insertion helps the establishment of efficient information structure for long-term usage. The indexing structure is dynamic and evolves over time. A relevance network can adapt to specific user needs, or it can generalize over multi-user information requirements, supporting sharing and collaborative work. The network can easily be integrated with other information retrieval and hypermedia systems to provide user-centered, and rapid information access.

Acknowledgments

Many thanks to Peter Brusilovsky and the anonymous reviewers for very pertinent comments on this paper. Many thanks to Josh Rabinowitz and Shawn Wolfe for their help in implementing the Adaptive HyperMan system. Thanks to all members of the Electronic Documentation Project team at NASA/JSC and to Shuttle flight controllers for their support.

References

Agosti, M., Melucci, M., and Crestani, F.: 1995, 'Automatic Authoring and Construction of Hypermedia for Information Retrieval'. *Multimedia Systems* **3**, 15–24.

Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T.: 1995, 'WebWatcher: A Learning Apprentice for the World Wide Web'. *Proc. AAAI Spring Symposium on Information Gathering from Distributed, Heterogeneous Environments*, Stanford, California.

Barreau, D. K.: 1995, 'Context as a Factor in Personal Information Management Systems'. *Journal of the American Society for Information Science* **46**(5), 327–339.

Belew, R. K.: 1989, 'Adaptive Information Retrieval: Using a Connectionist Representation to Retrieve and Learn about Documents'. *Proc. Twelfth SIGIR Conference*, Cambridge, Massachusetts, pp. 11–20.

Belkin, N. J., Marchetti, P. G., and Cool, C.: 1993, 'Braque: Design of an Interface to Support User Interaction in Information Retrieval'. *Information Processing and Management* **29**(3), 325–344.

Bergstrom, P. and Riedl, J.: 1994, 'Group Lens: A Collaborative Filtering System for Usenet News'. MSc thesis.

Bradshaw, J. M., Richards, T., Fairweather, P., Buchanan, C., Guay, R., Madigan, D., and Boy, G. A.: 1993, 'New Directions for Comupter-Based Training and Performance Support in Aerospace'. *Proc. Fourth Int. Conference on Human-Machine Interaction and Artificial Intelligence in Aerospace*, Toulouse, France.

Bradshaw, J. M., Dutfield, S., Benoit, P., and Wooley, J. D.: 1996, 'KAoS: Toward an Industrial-Strength Open Agent Architecture'. To appear in: J. M. Bradshaw (ed.): *Software Agents*. AAAI/MIT Press.

Boy, G. A..: 1991, 'Indexing Hypertext Documents in Context'. *Proc. Third ACM Conference on Hypertext*, San Antonio, Texas, pp. 51–61.

Brusilovsky, P.: 1994, 'Adaptive Hypermedia: An Attempt to Analyze and Generalize'. *Proc. Workshop on Adaptive Hypertext and Hypermedia, Fourth Int. Conference on User Modeling*, Hyannis, Ma. URL:http://www.cs.bgsu.edu/hypertext/adaptive/Brusilovsky.html

Callan, J. P., Croft, W. B., and Harding, S. M.; 1992, 'The Inquiry Retrieval System'. *Proc. Third Int. Conference on Database and Expert System Applications*, pp. 78–83.

Chen, H.: 1995, 'Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms'. *Journal of the American Society for Information Science* **46**(3), 194–216.

Chen, J. and Mathé, N.: 1994, 'Adaptive Dynamic Hypertext based on Paths of Traversal'. *Proc. Workshop on Adaptive Hypertext and Hypermedia, Fourth Int. Conference on User Modeling*, Hyannis, Ma. URL:http://www.cs.bgsu.edu/hypertext/adaptive/Mathe.html

Dumais, S. T., Furnas, G. W., and Landauer, T. K.: 1988, 'Using Latent Semantic Analysis to Improve Access to Textual Information'. *Proc. Sixth Conference on Human Factors in Computing Systems* (CHI'88), pp. 281–285.

Fischer, G. and Stevens, C.: 1991, 'Information Access in Complex, Poorly Structured Information Spaces'. *Proc. Eighth ACM Conference on Human Factors in Computing Systems* (CHI'91), New Orleans, Louisiana, pp. 63–70.

Fuhr, N.: 1992, 'Probabilistic Models in Information Retrieval'. *The Computer Journal* **35**(3), 243–255.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D.: 1992, 'Using Collaborative Filtering to Weave an Information Tapestry'. *Communications of the ACM* **35**(12), 61–70.

Haines, D. and Croft, W. B.: 1993, 'Relevance Feedback and Inference Networks'. *Proc. Sixteenth Int. ACM SIGIR Conference*, Pittsburgh, Pennsylvania, pp. 2–10.

Höök, K., Karlgren, J., Waern, A., Dahlback, N., Jansson, C. G., Karlgren, K., and Lemaire, B.: 1996: 'A Glass Box Approach to Adaptive Hypermedia'. *User Modeling and User Adapted Interaction* **6**, this issue.

Jennings, A, and Higichi, H.: 1993, 'A Personal News Service Based on a User Model Neural Network'. *IEICE Transactions on Information* Systems **75**(2), 198–209.

Kaplan, C., Fenwick, J., and Chen, J.: 1993, 'Adaptive Hypertext Navigation Based on User Goals and Context'. *User Modeling and User Adapted Interaction* **3**, 193–220.

Katsumoto, M., Fukuda, M., and Shibata, Y.: 1995, 'Distributed Design Image Database based on Perceptional Link Method'. *Proc. of the 1995 Pacific Workshop on Distributed Multimedia Systems* (DMS'95), pp. 76–83.

Kibby, M. R. and Mayes, J. T.: 1989, 'Towards Intelligent Hypertext'. In: R. McAleese (ed.): *Hypertext Theory into Practice*. Albex, pp. 164–172.

Klark, P., and Manber, U.: 1995, 'Developing a Personal Internet Assistant'. *Proc. World Conference on Educational Multimedia and Hypermedia* (ED-MEDIA'95), Graz, Austria, pp. 372–377.

Lashkari, L., Metral, M., and Maes, P.: 1994, 'Collaborative Interface Agents'. *Proc. Twelfth National Conference on Artificial Intelligence*, Seattle, Washington, pp. 444–449.

Maes, P. and Kozierok, R.: 1993, 'Learning Interface Agents'. *Proc. Eleventh National Conference of Artificial Intelligence*, Washington, D.C., pp. 459–465.

Mathé, N. and Chen, J.: 1994, 'A User-Centered Approach to Adaptive Hypertext based on an Information Relevance Model'. *Proc. Fourth Int. Conference on User Modeling*, Hyannis, Massachusetts, pp. 107–114.

Monk, A.: 1989, 'The Personal Browser: A Tool for Directed Navigation in Hypertext Systems'. *Interacting with Computers* **1**(2), 190–196.

Pargman, D.: 1994, no title. *Proc. Workshop on User Modeling in Information Retrieval, Fourth Int. Conference on User Modeling*, Hyannis, Massachusetts.

Rabinowitz, J., Mathé, N., and Chen, R. J.: 1995, 'Adaptive HyperMan: A Customizable Hypertext System for Reference Manuals'. *Proc. AAAI Fall Symposium on Artificial Intelligence Applications in Knowledge Navigation*, Cambridge, Massachusetts. pp. 110–115.

Robertson, S. E.: 1977, 'The Probability Ranking Principle in IR'. *Journal of Documentation* **33**(4), 294–304.

Robertson, S. E. and Spark Jones, K.: 1976, 'Relevance Weighting of Search terms'. *Journal of the American Society for Information Science* **27**, 129–146.

Rose, D. E. and Belew, R. K.: 1991, 'A Connectionist and Symbolic Hybrid for Improving Legal Research'. *Int. Journal of Man-Machine Studies* **35**(1), 1–33.

Salton, G.: 1989, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computers*. Addison Wesley, Reading, MA.

Salton, G., Wong, A., and Yang, C.S.: 1975, 'A Vector Space Model for Automatic Indexing'. *ACM Communications* **18**(11), 613–620.

Sheth, B. and Maes, P.: 1993: 'Evolving Agents for Personalized Information Filtering'. *Proc. Ninth IEEE Conference on Artificial Intelligence for Applications*, pp. 345–352.

Torrance, M.: 1995, 'Active Notebook: A Personal and Group Productivity Tool for Managing Information'. *Proc. AAAI Fall Symposium on Artificial Intelligence Applications in Knowledge Navigation and Retrieval*, Cambridge, Massachusetts, pp. 131–135.

Tudhope, D., Taylor, C., and Benyon-Davies, P.: 1995, 'Navigation via Similarity in Hypermedia and Information Retrieval'. In: R. Kuhlen and M. Rittberger (eds.): *Proc. Conference on Hypertext,*

Information Retrieval, Multimedia (HIM'95), Universitätsverlag Konstanz, Konstanz, pp. 203–218.
Turtle, H. and Croft, W.B.: 1991, 'Evaluation of an Inference Network-Based Retrieval Model'. *ACM Transactions on Information Systems* 9(3), 187–222.
van Rijsbergen, C.: 1979, *Information Retrieval.* 2nd Edition, Butterworths, London.
Vassileva, J.: 1994, 'A Practical Architecture for User Modeling in a Hypermedia-Based Information System'. *Proc. Fourth Int. Conference on User Modeling*, Hyannis, Massachusetts, pp. 115–120.
Vassileva, J.: 1996, 'A Task-Centered Approach for User Modeling in a Hypermedia Office Documentation System'. *User Modeling and User Adapted Interaction* **6**, this issue.

Authors' Vitae

Dr. N. Mathé is a research scientist at NASA Ames Research Center, employed through Recom Technologies, Inc. Dr. Mathé received her MS. degree in Physics from Paris VI University, France in 1986, and her Ph.D. in Computer Science from the National School of Aeronautics and Space Technology, France, in 1990. Since 1992, Dr. Mathé leads the Advanced Interaction Media group in the Computational Sciences Division at NASA Ames. Her research objective is to facilitate access to electronic information via the development of intelligent tools integrating hypertext, multimedia, collaborative learning, and knowledge-based systems capabilities.

Dr. J. R. Chen is a research scientist at NASA Ames Research Center, employed through Recom Technologies, Inc. Dr. Chen received his M.S. degree in Systems Analysis from the University of Wisconsin, Madison, and his Ph.D. in Computer Science from the University of California, San Diego. Dr. Chen is a member of the Advanced Interaction Media group at NASA Ames since 1993. His research interests include adaptive information retrieval, collaborative intelligent agents, and connectionist knowledge representation.