CAR-TR-830 CLIS-TR-96-02 CS-TR-3643 EE-TR-96-25 May, 1996

A Conceptual Framework for Text Filtering*

Douglas W. Oard
Medical Informatics and Computational Intelligence Laboratory
Electrical Engineering Department

and

Gary Marchionini
Human-Computer Interaction Laboratory
Center for Automation Research and
Digital Library Research Group
College of Library and Information Services

University of Maryland, College Park, MD 20742
{oard|march}@glue.umd.edu

Abstract

This report develops a conceptual framework for text filtering practice and research, and reviews present practice in the field. Text filtering is an information seeking process in which documents are selected from a dynamic text stream to satisfy a relatively stable and specific information need. A model of the information seeking process is introduced and specialized to define information filtering. The historical development of text filtering is then reviewed and case studies of recent work are used to highlight important design characteristics of modern text filtering systems. Specific techniques drawn from information retrieval, user modeling, machine learning and other related fields are described, and the report concludes with observations on the present state of the art and implications for future research on text filtering.

*The research reported herein was supported in part by NSF grant IRI-9357731, DOD grant MDA9043C7217, the Medical Informatics Network project of the Pathology Department, a Department of Education technology challenge grant, and the Logos Corporation

Process	Information Need	Information Sources
Information Filtering	Stable and Specific	Dynamic and Unstructured
Information Retrieval	Dynamic and Specific	Stable and Unstructured
Database Access	Dynamic and Specific	Stable and Structured
Information Extraction	Specific	Unstructured
Alerting	Stable and Specific	Dynamic
Browsing	Broad	Unspecified
Entertainment	Unspecified	Unspecified

Table 1: Examples of information seeking processes.

1 Introduction

With the growth of the Internet and other networked information, research in automatic mediation of access to networked information has exploded in recent years. This report reviews existing work on text filtering, a type of "information seeking." Here we use "information seeking" as an overarching term to describe any processes by which users seek to obtain information from automated information systems [27]. Table 1 shows common types of information seeking processes. In the "information filtering" process the user is assumed to be seeking information which addresses a specific long-term interest. In this report we will describe general approaches to the information filtering problem and specific techniques that are tailored for "text filtering," the case in which the information sought is in text form.

Information filtering systems are typically designed to sort through large volumes of dynamically generated information and present the user with sources of information that are likely to satisfy his or her information requirement. By "information sources" we mean entities which contain information in a form that can be interpreted by a user. We commonly refer to information sources which contain text as "documents," but in other contexts these sources may be audio, still or moving images, or even people. The information filtering system may either provide these entities directly (which is practical when the entities are easily replicated), or it may provide the user with references to the entities.

This description of information filtering leads immediately to three subtasks: *collecting* the information sources, *selecting* the information sources, and *displaying* the information sources. Figure 1 depicts this subdivision, one which is applicable to a wide variety of information seeking processes. The same three tasks are also fundamental to a process commonly referred to as "information retrieval" in which the system is presented with a query by the user and expected to produce information sources which the user finds useful. "Text retrieval," the specialization of information retrieval to retrieve text, has an extensive research heritage. In one of the classic works on information filtering, this observation led Belkin and Croft to suggest that the information filtering process would be an attractive application for techniques that had already developed for information retrieval systems [2].

The distinction between process and system is fundamental to understanding the

Figure 1: Information seeking task diagram.

difference between information filtering and information retrieval. By "process" we mean an activity conducted by humans, perhaps with the assistance of a machine. When we refer to a type of "system" we mean an automated system (i.e., a machine) that is designed to *support* humans who are engaged in that process. So an information filtering system is a system that is *intended* by its designers to support an information filtering process. Much of the confusion that arises on this issue can be traced back to creative applications of techniques that were designed originally to support one type of information seeking process (e.g., information retrieval) to another (e.g., information filtering).

Any information seeking process begins with the users' goals. The distinguishing features of the information filtering process are that the users' information needs (or "interests") are relatively specific (a point we shall come back to when we define browsing), and that those interests change relatively slowly with respect to the rate at which information sources become available. Although the information retrieval process is also restricted to specific information needs, historically information retrieval research has sought to develop systems which use relatively stable information sources to respond to collections of (possibly) unrelated queries. So a traditional information retrieval *system* can be used to perform an information filtering *process* by repeatedly accumulating newly arrived documents for a short period, issuing an unchanging query against those documents, and then flushing the unselected documents. But the information filtering *process* is distinguished from the information retrieval *process* by the nature of the user's goal. Figure 2 depicts this distinction graphically. While the grand challenge for information seeking systems is to match rapidly changing information with highly variable interests, information retrieval and information filtering both explore important areas of this problem space for which a number of practical applications exist.

It is useful to highlight the distinction between information filtering and information retrieval because systems designed to support the information filtering process can exploit evidence about relatively stable interests to develop sophisticated models of the users' information needs. Information filtering can be viewed as an application of user modeling techniques to facilitate information seeking in dynamic environments. In summary, the design of information filtering *systems* can be based on two established lines of research, information retrieval and user modeling.

1.1 Collection and Display

This report describes the design of systems to support the text filtering process with particular emphasis on the information selection component. Because such an emphasis might leave the reader with the mistaken impression that collection and display are lesser challenges, we pause briefly to describe the relationship between selection and the other two components depicted in figure 1.

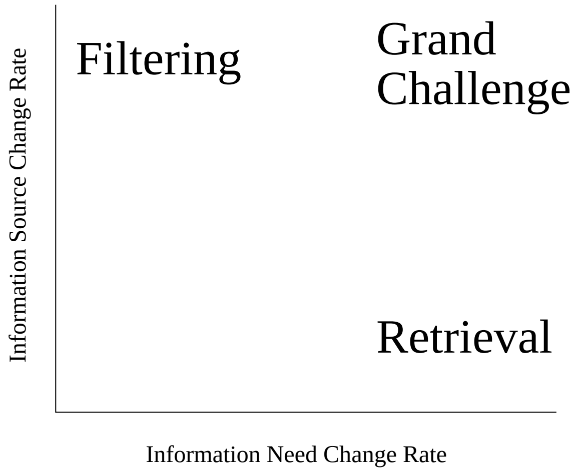

Figure 2: Information seeking processes for relatively specific information needs.

Dynamic information can be collected actively (e.g., with autonomous agents over the World Wide Web), collected passively (e.g., from a newswire feed) or some combination of the two. Early descriptions of the information filtering problem implicitly assumed passive collection (c.f. [7, 18]). As the amount of electronically accessible information has exploded, active collection has become increasingly important (c.f. [44]). Active collection techniques can benefit from a close coupling between the collection and selection modules because they exploit both user and network models to perform information seeking actions in a network on behalf of the user. In a fully integrated information filtering system, some aspects of user model design are likely to be common to the two modules. That commonality would provide a basis for sharing information about user needs across the inter-module interface. But because the purpose of the collection module is to choose whether to *obtain* information before that information is known while the purpose of the selection module is to choose information to *retain* for display to the user once that information has been collected, the user model for the selection module is not likely to be identical to the user model for the collection module. In the succeeding sections we will generally limit the discussion to systems which use passive collection techniques, both because this choice allows us to concentrate on the selection component and because there has been little reported on how the two components can be integrated.

Such a clean division is not possible for the interface between the selection and the display components, however. The goal of an information filtering system is to enhance the user's ability to identify useful information sources. While this can be accomplished by automatically choosing which sources of information to display, experience has shown that user satisfaction can be enhanced in interactive applications by using techniques which exploit the strengths of both humans and machines.

A personalized electronic conference system that lists submissions in order of decreasing likelihood of user interest is one example of such an approach. The automatic system can use computationally efficient techniques to place documents which are likely to be interesting near the top of the list, and then users can rapidly apply sophisticated heuristics (such as word sense interpretation and source authority evaluation) to select those documents most likely to meet their information need. If the system has produced a good rank ordering, the density of useful documents should be greatest near the top of the list. As the user proceeds down the list, selecting interesting documents to review, he or she should thus observe that the number of useful documents is decreasing. By allowing the human to adaptively choose to terminate their information seeking activity based in part on the observed density of useful documents, human and machine synergistically achieve better performance than either could achieve alone.

In other words, in interactive applications an imperfectly ranked list (referred to as "ranked output") can be superior to an imperfectly selected set of documents (referred to as "exact match" selection) because humans are able to adaptively choose the set size based on the same heuristics that they use to choose which documents to read. The choice of a ranked output display design imposes requirements on the selection module, however. Because the display module must rank the documents, the selection module must provide some basis (e.g., a numeric "status value") from which the ranking can be constructed. Display design is a rich research area in its own right, but our discussion of the issue is focused solely on aspects of the display design that impose requirements on the selection module.

1.2 Other Information Seeking Processes

We have already mentioned information retrieval, but there are other information seeking processes for which the decomposition in figure 1 is appropriate. One of the most familiar is the process of retrieving information from a database. The distinguishing feature of the database retrieval process is that the output will *be* information,[1] while in information filtering (or retrieval), the output is a set of entities (e.g., documents) which *contain* the information which is sought [3]. For example, using an library catalog to find the *title* of a book would be a database access process. Using the same system to discover whether any new books *about* a particular topic have been added to the collection would be an information filtering process. As this example shows, database *systems* can be applied to information filtering processes, and we will present examples of this in section 4.

Another process that can be described using figure 1 is information extraction. The information extraction process is similar to database access in that the goal is to provide information to the user, rather than entities which *contain* information. It is distinguished from the database access process by the nature of the sources from which that information is obtained. In the database access process information is obtained from some type of database (e.g., a relational database), while in information extraction the information is less well structured (e.g., the body of an electronic mail message). Information extraction *techniques* are sometimes found in the selection module of a text filtering process, helping to represent texts in a way that facilitates selection.

One interesting variation on the information extraction and database access processes is what is commonly referred to as "alerting." In the alerting process the information need is assumed to be relatively stable with respect to the rate at which the information itself is changing.[2] Monitoring an electronic mailbox and alerting the user

[1]While it is common to draw a distinction between information and data in which the concept of "information" includes some basis for its interpretation, our focus on selection makes it possible to combine the two concepts and refer to both as "information."

[2]Recall that in an information filtering process it is the information *sources*, rather than the information

whenever mail from a specific user arrives is one example of an information alerting process. Presenting mail from that user first in a sorted list would be an example of information filtering.

Database retrieval, information extraction, and alerting techniques all inform text filtering practice, and three benefit from advances in text filtering research. We do not intend to comprehensively review those research areas, but we do occasionally mention how relevant technologies developed to support those processes can be applied to support the information filtering process.

Finally, "browsing" is another information seeking process for which the decomposition shown in figure 1 is appropriate. Since browsing can be performed on either static or dynamic information sources, browsing has aspects similar to both information filtering and information retrieval. "Surfing the World Wide Web" is an example of browsing relatively static information, while reading an online newspaper would be an example of browsing dynamic information. The distinguishing feature of browsing is that the users' interests are assumed to be broader than in the information filtering or retrieval processes. Precisely what is meant by "broader" is difficult to define, however, and the distinction is often simply a matter of judgement. In order to sharpen the distinction for the purpose of this report, we propose an operational definition of browsing. When an interest is so broad that it cannot be represented effectively in an information filtering (or retrieval) system, we will refer to the information seeking process as browsing rather than as filtering or retrieval. In other words, we propose that researchers seek to characterize the broadest interests for which their information filtering systems are useful, and then refer to the limitations they discover in that way as the dividing line between filtering and browsing for their system.

2 Terminology

In a field as diverse as information filtering it is inevitable that a rich and sometimes conflicting set of terminology would emerge. Sometimes this is simply the result of differing perspectives, other times new terminology is needed to convey subtly different meanings. For example, "information retrieval" is sometimes used expansively to include information filtering. But it is also commonly used in the more restricted sense that we have defined. Information filtering is alternatively referred to as "routing" (with a heritage in message processing) as "Selective Dissemination of Information" or "SDI" (with a heritage in library science), as "current awareness," and as "data mining." Sometimes routing is used to indicate that every document goes to some (and perhaps exactly one) user. Information filtering is sometimes associated with passive collection of information, and is sometimes meant to imply that an all-or-nothing (i.e., unranked) selection is required. SDI is sometimes used to imply that the profiles which describe the information need are constructed manually. The use of "current awareness" is sometimes meant to imply selection of new information based solely on the title of a journal, magazine, or other serial publication. And "data mining" is sometimes taken to imply that vast quantities of information are available simultaneously. All of those interpretations have a historical basis, but it is not uncommon to find these terms used to describe systems which lack the distinguishing characteristics of

itself, which change.

their historical antecedents. We shall avoid this problem by referring to all of these variations as "information filtering."

Taylor defined four types of information need (visceral, conscious, formalized, and compromised) that reflected the process of moving from the actual (but perhaps unrecognized) need for information to an expression of the need which could be represented in an information system [39]. In common use, however, application of the terminology is unfortunately not nearly so precise. The visceral information need is often referred to as an "interest" or simply as an "information need." But it is occasionally referred to as a topic, a term that is sometimes (e.g., in the TREC evaluation we describe in section 4) used to describe the formalized (i.e., the human expression of) the information need. And in some experimental work, the visceral information need is referred to as a "query" even though "query" is the traditional term for Perry's concept of a compromised information need that could be submitted to an information retrieval system. In this report, we use "interest" and "information need" interchangeably to refer to the visceral information need, and reserve the use of the terms "topic" and "query" for their more specific meanings.

In an information filtering system, the system's *representation* of the information need (i.e., the compromised information need) is commonly referred to as a "profile." Because the profile fills the same role as what is commonly called a "query" in information retrieval and database systems, sometimes the term "query" is used instead of "profile" in information filtering as well. It would not be technically correct to call the profile a "user model" because a user model consists of both a *representation* of the users interests and a method for *interpreting* that representation to make predictions. But that usage occasionally appears as well. We shall avoid confusion on this subject by using only the term "profile" when referring to the compromised information need in the context of information filtering.

3 Historical Development

Luhn introduced the idea of a "Business Intelligence System" in 1958 [25]. In Luhn's concept, library workers would create profiles for individual users, and then those profiles would be used in an exact-match text selection system to produce lists of new documents for each user. Orders for specific documents would be recorded and used to automatically update the requester's profile. Foreshadowing later concerns about privacy, he also observed that a set of profiles could be used to identify which users had expertise in specific areas.

Luhn's early work identifies every aspect of a modern information filtering system, although the microfilm and printer technology of the day resulted in significantly different implementation details. In describing the function of the selection module as "selective dissemination of new information" he coined the term which described this field for nearly a quarter century.

A decade later, widespread interest in Selective Dissemination of Information (SDI) resulted in creation of the Special Interest Group on SDI (SIG-SDI) of the American Society for Information Science. Houseman's 1969 survey for that organization identified 60 operational systems, nine of which served over 1,000 users each [18]. These systems generally followed Luhn's model, although only four of the 60 implemented automatic profile updating, with the rest about evenly split between manual maintenance of the profiles by professional support staff or by the users themselves. Two factors had led organizations to make this investment in SDI: the availability of timely information in electronic form, and the affordability of sufficient computing capability to match those documents with user profiles. These are the same factors motivating information filtering today, although distribution of scientific abstracts on magnetic tape (the dominant source of external information at the time) has been replaced by nearly instantaneous communications across large networks of interconnected computers.

Denning coined the term "information filtering" in his ACM President's Letter that appeared in the Communications of the ACM in March of 1982 [7]. Introducing the new ACM Transactions on Office Information Systems, Denning's objective was to broaden a discussion which had traditionally focused on *generation* of information to include *reception* of information as well. He described a need to filter information arriving by electronic mail in order to separate urgent messages from routine ones, and to restrict the display of routine messages in a way that matches the personal mental bandwidth of the user. Among the possible approaches he identified was a "content filter." The remaining six techniques (hierarchical organization of mailboxes, separate private mailboxes, special forms of delivery, importance numbers, threshold reception, and quality certification) all required the cooperation of the other users, and hence would better be studied from a more global perspective the receiver's local scope of action represented by the information seeking model in figure 1. We shall have more to say on Denning's other approaches in section 5.3.2.

Over the subsequent decade, occasional papers on information filtering applications appeared in the literature. While electronic mail was the original domain about which Denning had written, subsequent papers have addressed newswire articles, Internet "News" articles,[3] and broader network resources [9, 19, 30, 43]. The most influential paper of this period was published in the Communications of the ACM by Malone and others in 1987 [26]. There they introduced three paradigms for information selection, *cognitive*, *economic*, and *social*, based on their work with a system they called the "Information Lens." Their definition of cognitive filtering, the approach actually implemented by the Information Lens, is equivalent to the "content filter" defined earlier by Denning, and this approach is now commonly referred to as "content-based" filtering. They also described an economic approach to information filtering, a generalization of Denning's "threshold reception" idea, that had implications beyond the scope of the information seeking system model in figure 1. We describe the economic issues related to information filtering briefly in section 5.3.3.

The most important contribution of Malone and his colleagues was to introduce an alternative approach which they called social (now also called "collaborative") filtering. In social filtering, the representation of a document is based on annotations to that document made by prior readers of the document. They speculated that by exchanging this sort of information, communities of shared interest could be automatically identified.[4] If practical, social filtering would provide a basis for selection of

[3]Internet "News" (more properly USENET News) is not a news source in the traditional sense, but rather a form of distributed electronic conference support system in which submissions (referred to as articles) are propagated to central repositories at participating institutions.

[4]The principal difference between social filtering and Denning's more limited concept of "quality certification" is that annotations can be combined more flexibly in social filtering.

information items, regardless of whether their content could be represented in a way that was useful for selection. The balance between content-based and collaborative filtering is an important unresolved issue, and we will have much more to say on the relative merits of the two approaches in the sections that follow.

Large-scale government-sponsored research on information filtering also began in this period. In 1989 the United States Defense Advanced Research Projects Agency (DARPA) sponsored the first of an ongoing series of Message Understanding Conferences (MUC) [23, 17].[5] The principal thrust of those conferences has been use of information extraction techniques to support the selection of messages. In 1990, DARPA launched the TIPSTER project to fund the research efforts of several of the MUC participants [12]. TIPSTER added an emphasis on the use of statistical techniques to preselect messages that could then be subjected to more sophisticated natural language processing. In TIPSTER, this the preselection process is known as "document detection." In 1992 The National Institute of Standards and Technology (NIST) capitalized on this research by co-sponsoring (with DARPA) an annual Text REtrieval Conference (TREC) focused specifically on text filtering and retrieval [13].

So for the first decade after Denning identified networked information as an important application for filtering technology, information filtering was either addressed episodically or included as part of a broader research effort. Finally, in November of 1991, Bellcore and the ACM Special Interest Group on Office Information Systems (SIGOIS) jointly sponsored a workshop on "High Performance Information Filtering" that brought together a substantial quantity of research to establish a basis for the explosive growth the field has experienced in the past five years. Forty contributors examined the area from a wide variety of perspectives, including user modeling, information selection, application domains, hardware and software architectures, privacy, and case studies. A year later, in December of 1992, expanded versions of nine papers from that workshop appeared in a special issue of the Communications of the ACM [1, 2, 4, 10, 11, 24, 31, 36, 37].

4 Case Studies

The recent surge of interest in information filtering has actually contributed to the flood of information, since there is now more being published in the field than any single individual could hope to read. In part this results from the coincident adoption of the World Wide Web as a rapid means for the dissemination of academic work. Presently there are literally hundreds of documents about information filtering accessible through that medium.[6] In this section we describe the two dominant research paradigms, content-based and social filtering, and examine issues related to each. We have selected systems to discuss which highlight the most important approaches that have been used and the significant issues which have been raised.

[5]The first two Message Understanding Conferences were known as "MUCK-I" and "MUCK-II." Subsequent conferences adopted the shorter acronyms "MUC-3," etc.

[6]Network-accessible resources on information filtering that are known to the authors are collected at http://www.ee.umd.edu/medlab/filter

4.1 Content-Based Filtering

With a research heritage extending back to Luhn's original work, the content-based filtering paradigm is the better developed of the two. In content-based filtering, each user is assumed to operate independently. As a result, document representations in content-based filtering systems can exploit only information that can be derived from document contents. Yan implemented a simple content-based text filtering system for Internet News articles in a system he called SIFT [46].[7] Profiles for SIFT are constructed manually by specifying words to prefer or avoid, and must be updated manually if the user desires to change them. For each profile, twenty articles are made available each day in a ranked output format. Articles can be selected interactively using a World Wide Web browser. For users lacking interactive access, clippings (the first few lines of each article) can instead be sent by electronic mail. In that case selections must be done without user interaction, so users are offered the option of defining a profile for an exact match text selection technique.

SIFT offers two facilities to assist users with profile construction. Users are initially offered an opportunity to apply candidate profiles against the present day's articles to determine whether appropriate sets of articles are accepted and rejected. If a substantial amount of information on that interest is present on Internet News that day, iterative refinement allows the user to construct a profile which will move the appropriate articles to the top of the list. To facilitate maintenance of profiles over time, words which contributed to the position of each article in the ranked list are highlighted (a technique known as "Keyword in Context" or "KWIC") when using a World Wide Web browser to access the articles. By examining the context of words which occur with meanings that were unforeseen at the time the profile was constructed, users can select additional words which appear in the same context to add to the list of words to be avoided.

Yan developed SIFT to study efficient algorithms for information filtering. In his implementation, large collections of profiles are compared to every article arriving on Internet News by a central server. Efficiencies are obtained by grouping profiles in ways that permit parts of the filtering process to be performed on groups of profiles rather than individually. SIFT makes no distinction among the words appearing in an article, so words appearing in the newsgroup name (i.e., the specific conference), the author's electronic mail address, the article title, the body of the article, included text, or the "signature" information that is routinely added to every document by some users are all equally likely to result in a high rank for a document.

Stevens developed a system called InfoScope which used automatic profile learning to minimize the complexity of exploiting information about the context in which words were used [38]. Also designed to filter Internet News, InfoScope deduced exact-match rules and offered them for approval (possibly with modifications) by the user. These suggestions were based on simple observable actions such as the time spent reading a newsgroup or whether an individual message was saved for future reference. By avoiding the requirement for explicit user feedback about individual articles, InfoScope was designed to minimize the cognitive load of managing the information filtering system.

[7] At the time of this writing, free interactive access to SIFT is available at http://sift.stanford.edu, but relocation to http://www.reference.com has been announced.

While SIFT treats Internet News as a monolithic collection of articles, InfoScope was able to make fine-grained distinctions between newsgroups, subjects, and even individual authors. Implementation of such extensive deconstruction led Stevens to introduce a facility to reconstruct levels of abstraction in a way that was meaningful to the user. InfoScope implemented this abstraction at the newsgroup level, suggesting to combine related sets of newsgroups that were regularly examined by the user to form a single "virtual newsgroup." By defining filters for virtual newsgroups with possibly overlapping sources, users were thus provided with a powerful facility to reorganize the information space in accordance with their personal cognitive model of the interesting parts of the discussions they wished to observe.

InfoScope was not without its limitations, however. The experimental system Stevens developed was able to process only information in the header of each article (e.g., subject, author, or newsgroup), a restriction imposed by the limited personal computer processing power available in 1991. In addition, his goal of exploring the potential for synergy between user and machine for profile management led him to choose a rule-based exact match text selection technique. Since users are often able to verbalize the selection rules they apply, Stevens reasoned that users would have less difficulty visualizing the effect of changing rules than the effect of changing the types of profiles commonly found in ranked output systems. InfoScope's key contributions, machine-assisted profile learning, the addition of user-controlled levels of abstraction, and implicit feedback, make it an excellent example of a complete content-based information filtering system intended for interactive use.

Because of their low cost, large volume, and ease of recognizing new information, Internet News and electronic mail have been popular domains for information filtering research. Unfortunately, these domains are poorly suited to formal experiments because reproducible results are difficult to obtain. For this reason, very little is known about the effectiveness of either SIFT or InfoScope. Stevens reported that eight of ten experienced Internet News readers preferred InfoScope to their prior software in his initial study, and that all five users in the second evaluation reported that fewer uninteresting articles were presented and more interesting articles were read in a second half of a 10 week evaluation than in the first. Because SIFT was developed to study efficiency rather than effectiveness issues, even less information is available about its effectiveness. Yan does report, however, that in early 1995 SIFT routinely processed over 13,000 profiles and was adding approximately 1,400 profiles each month [46]. Even though one user may create several profiles, this level of user acceptance makes a powerful statement about the utility of even the simple approach used by SIFT.

Learning more about the effectiveness of a text filtering technique requires that the technique be evaluated under controlled experimental conditions. And because the performance of text filtering techniques varies markedly when different information needs and document collections are used, comparison of results across systems is facilitated when those factors are held constant. The TREC evaluation has provided an unprecedented venue for exactly this type of performance evaluation. Conducted annually since 1992, the most recent conference (TREC-4) attracted participation from 24 universities and 12 corporations [14].

NIST provides each participant with fifty topics and a large set (typically thousands) of training documents and relevance assessments on those documents[8] for each

[8] Relevance assessments for the TREC "routing" (text filtering) training documents generally are derived

information need. Participants train their text filtering systems, using this data as if it represented explicit feedback on the utility of each training document to a user, and then must register their profiles with NIST before receiving the evaluation documents. The profiles are then used by the text filtering systems which generated them to rank order a previously unseen set of evaluation documents, and the top several thousand documents are submitted to NIST for evaluation.

In order to achieve reproducible results, it is necessary to make some very strong assumptions about the nature of the information filtering task. In TREC it is assumed that human judgements about whether an information need is satisfied by a document are binary valued (i.e., a document is relevant to an information need or it is not) and constant (i.e., it does not matter who makes that judgement or when they make it). Relevance, the fundamental concept on which this methodology is based, actually fails to satisfy both of those assumptions. Human relevance judgments exhibit significant variability across evaluators, and for the same evaluator across time. Furthermore, evaluators sometimes find it difficult to render a binary relevance judgment on a specific combination of a document and an information need. Nevertheless, performance measures based on a common set of relevance judgements provide a principled basis for for comparing the *relative* performance of different text filtering techniques.

The TREC filtering evaluation is based on effectiveness measures that are commonly used for text retrieval systems. The effectiveness of exact match text retrieval systems is typically characterized by three statistics: "precision," "recall," and "fallout." Precision is the fraction of the selected documents which are actually relevant to the user's information need, while recall is the fraction of the actual set of relevant documents that are correctly classified as relevant by the text filtering system. When used together, precision and recall measure selection effectiveness. Because both precision and recall are insensitive to the total size of the collection, fallout (the fraction of the non-relevant documents that are selected) is used to measure rejection effectiveness. Table 2 illustrates these relationships.

In TREC, almost all of the filtering systems produce ranked output. Accordingly, precision and fallout at several values of recall are reported, and "average precision" (the area under the precision-recall curve) is reported for use when a single measure of effectiveness is needed [34]. Average precision is computed by choosing successively larger sets of documents from the top of the ranked list that result in evenly spaced values of recall between zero and one. Precision is then computed for each set, and the mean of those values is reported as the average precision for an individual information need. The process is repeated for several information needs, and the mean of the values obtained is reported as the average precision for the system on that test collection. Clearly, larger values of average precision are better.

Only the selected documents must be scored to evaluate precision, but it would be impractical to evaluate recall and fallout by scoring every document in the TREC collection. The solution is to estimate recall and fallout by scoring a sample of the document collection. The approach chosen for TREC, known as "pooled relevance evaluation" is to evaluate every document chosen by *any* participating system and then assume that all unchosen documents are not relevant. Since documents are chosen using a wide variety of text filtering and retrieval techniques in TREC, it is felt that the pooled relevance methodology produces a fairly tight upper bound on recall and

from TREC text *retrieval* evaluations conducted in prior years.

Selected as	Actually is	
	Relevant	Not Relevant
Relevant	Found	False Alarm
Not Relevant	Missed	Correctly Rejected

$$\text{Precision} = \frac{\text{Found}}{\text{Found} + \text{False Alarm}}$$

$$\text{Recall} = \frac{\text{Found}}{\text{Found} + \text{Miss}}$$

$$\text{Fallout} = \frac{\text{False Alarm}}{\text{False Alarm} + \text{Correctly Rejected}}$$

Table 2: Measures of text selection effectiveness.

an extremely tight lower bound on fallout.

Although TREC investigates only the performance of the selection module, and that evaluation is necessarily based on a somewhat artificial set of assumptions, the resulting data provides a useful basis for choosing between alternative selection techniques. In the TREC-3 evaluation, for example, 25 text filtering systems were evaluated and average precision was observed to vary between 0.25 and 0.41.

4.2 Social Filtering

The Tapestry text filtering system, developed by Nichols and others at the Xerox Palo Alto Research Center (PARC), was the first to include social filtering [11, 40]. Designed to filter personal electronic mail, messages received from mailing lists, Internet News articles, and newswire stories, Tapestry allowed users to manually construct profiles based both on document content and on annotations made regarding those documents by other users. Those annotations were explicit binary judgements ("like it" or "hate it") that could optionally be made by each user on any message they read.

Like InfoScope, Tapestry profiles consisted of rules that specified the conditions under which a document should be selected. One important difference was that Tapestry allowed users to associate a score with each rule. Tapestry then generated ranked output by comparing the scores assigned by multiple rules. Tapestry implemented this sophisticated processing efficiently by dividing the filtering process into two stages using a client-server model. In the first stage, a central server with access to all of the documents applies a set of simple rules, similar to those used by SIFT, to determine whether each document may be of interest to each user. The more sophisticated rules in each profile are then executed in each users workstation (the client) to develop the ranked list.

Experience with several small scale trials of social filtering suggests that a critical mass of users with overlapping interests is needed for social filtering to be effective. Tapestry was restricted to a single site because both the content and the software were subject to proprietary restrictions, so only limited anecdotal evidence of the social filtering aspects of Tapestry's performance are available. The GroupLens project of Miller and others at the University of Minnesota is presently the most ambitious attempt to reach a critical mass on a dynamic information source [32].

GroupLens is designed to filter Internet News, a freely redistributable text source. Like Tapestry, GroupLens is built on a client-server model. GroupLens uses two types of servers, content servers (which are simply standard Internet News servers) and annotation servers (which have been developed for the project). The design permits both the content and annotation servers to be replicated so that each server can efficiently service a limited user population. Modified versions of some popular (and freely redistributable) Internet News client software are made available in order to encourage the development of a large user population, and implementers of other client software are permitted to incorporate the GroupLens protocol in their products.[9]

GroupLens annotations are explicit judgements on a five-valued integer scale. Unlike Tapestry, however, the annotations need not be assigned an *a priori* interpretation. Users may register annotations with their annotation server using whatever semantics for the five values they wish. The annotation servers collect annotations from their user population, use correlation information to predict their user evaluations of unseen articles, and provide those predictions to client programs on request. The initial GroupLens trial began in 1996 using a limited number of newsgroups and a single annotation server. Results are not yet available, but the project's important contributions, distributed annotation servers, profile learning for social filtering, and a design which encourages development of a large user base, provide an excellent prototype for future work on social filtering.

One limitation of the existing experimental work on social filtering is user motivation. In GroupLens, users annotate documents in order to improve the performance of their filter's ability to learn from other clients who have annotated the same documents. This creates a bit of a "chicken and the egg" problem, though, since there is no incentive for the first user to annotate anything. If content-based and social filtering are integrated in the same system, however, then a synergy between the two techniques can develop. Tapestry demonstrated one way in which the two approaches can be combined when manually constructed profiles are used. The URN system, developed by Brewer at the University of Hawaii, illustrates a more automatic method by which such synergy can be achieved.

URN was an Internet News filtering system in which users could provide two types of information to support profile learning. The first was by making explicit binary judgements about the utility of the document. Those judgements were then used as a basis for a typical content-based ranked output system. What makes URN unique is that users can also collaboratively improve the system's initial *representation* of the document by adding or deleting words which they feel represent (or, for deletions, misrepresent) the content of the document. In URN these changes are propagated to all other users, allowing the user community to collaboratively define the structure of the information space. Since user-specified words are given preference by URN when developing representations for new documents, users have an incentive to improve the set of words which describe existing documents.

[9]The GroupLens protocol and GroupLens client software can be obtained from http://www.cs.umn.edu/Research/GroupLens

In URN each user maintains a separate content-based user model, while the annotation server effectively maintains a single collaboratively-developed model of the document space. This approach lacks the sophistication of the separate user models based on shared annotations found in GroupLens, but URN's integration of content-based and social filtering techniques illustrates one way in which these two paradigms can be combined.

5 Text Filtering Technology

In this section we identify techniques which can be synthesized to produce effective and efficient text filtering systems. These techniques are drawn from a large number of fields, and our presentation will consider each field in turn. The essence of text filtering practice, however, is not the techniques themselves, but rather the way in which the techniques drawn from these fields are integrated to support a text filtering process.

5.1 Information Retrieval

As Belkin and Croft observed, content-based text selection techniques have been extensively evaluated in the context of information retrieval [2]. Every approach to text selection has four basic components:

- Some technique for representing the documents
- Some technique for representing the information need (i.e., profile construction)
- Some way of comparing the profiles with the document representations
- Some way of using the results of that comparison

The objective is to automate the process of examining documents by computing comparisons between the representation of the information need (the profile) and the representations of the documents. This automated process is successful when it produces results similar to those produced by human comparison of the the documents themselves with the actual information need. The fourth component, using the results of the comparison, is actually the role of the display module in figure 1. We include it here to emphasize the close coupling between selection and display.

In each of the text filtering systems we describe in this report, the selection module assigns one or more values to each document, and the display module then uses those values to organize the display. Figure 3 illustrates the representation and comparison process implemented by those systems. The domain of the profile acquisition function p is I, the collection of possible information needs and its range is R, the unified space of profile and document representations. The domain of the document representation function d is D, the collection of documents, and its range is also R. The domain of the comparison function c is $R \times R$ and its range is $[0,1]^n$, the set of n-tuples of real numbers between zero and one. In an ideal text filtering system,

$$c(p(\text{info need}), d(\text{doc})) = j(\text{info need}, \text{doc}), \forall \text{info need} \in I, \forall \text{doc} \in D,$$

where $j : I \times D \mapsto [0,1]^n$ represents the user's judgement of some relationships between an interest and a document, measured on n ordinal scales (e.g., topical similarity or degree of constraint satisfaction).

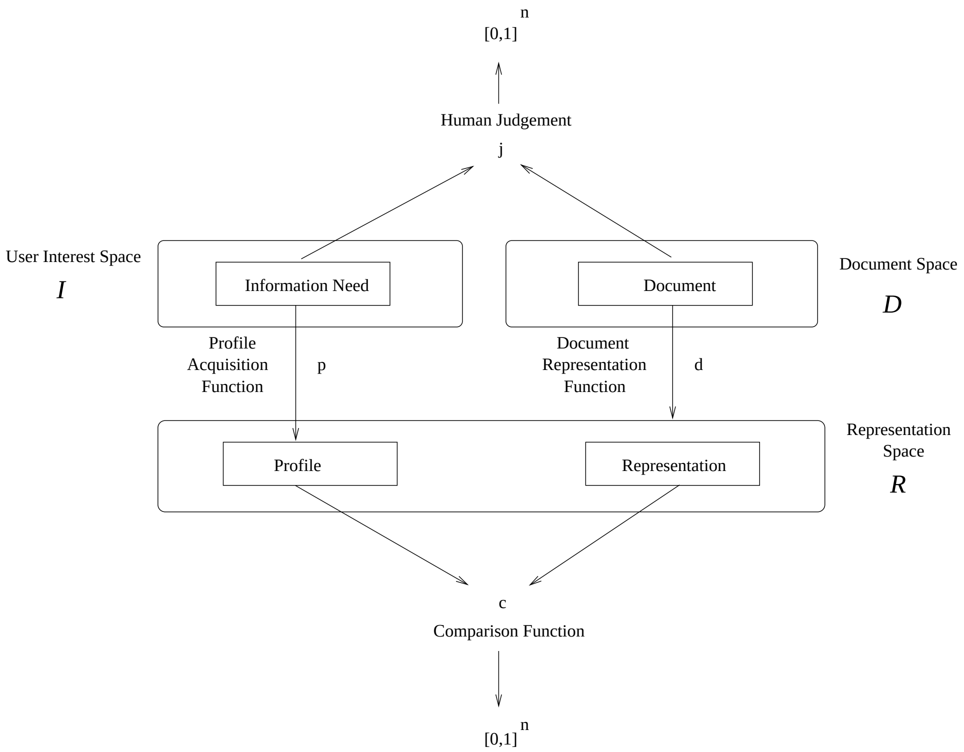

Figure 3: Text filtering system model.

As we saw in section 4, the representation can exploit information derived from the content of the document, annotations made by others, or some combination of the two. Although syntactic and semantic analysis of documents is possible, content-based text filtering systems typically use representations based on the frequency with which terms occur in each document.[10] One reason for this choice is that it lends itself to efficient implementation. But a more compelling reason is that because no domain-specific information is needed to form the representation, a demonstration of acceptable performance in one application is easily translated into similar performance in another.

Although content-based text filtering systems typically start with this term-frequency representation, they generally apply some type of transformation to that representation before invoking the comparison function c in figure 3 The nature of the transformation depends strongly on which characteristics of that representation the comparison function c is designed to exploit, however. For this reason, we describe the transformations together with thir associated comparison functions in the following paragraphs.

For an exact match text filtering system the range of the comparison function c is restricted to be either zero or one, and it is interpreted as a *binary judgement* about whether a document *satisfies* the profile. In this case, a step function that detects term presence is applied to the term-frequency representation when that representation is

[10]We use "terms" rather than "words" because the "terms" which are considered may be parts of words (e.g., overlapping three letter subsequences known as trigrams), single words, or combinations of words (e.g., idiomatic phrases). Common "stopwords" that have little use in subsequent processing are typically eliminated during term selection.

constructed so that the resulting boolean vector can be easily compared to the boolean expression specified by the profile. Exact match text filtering systems typically provide an unranked set of documents which will (hopefully) satisfy the information need. The exact match approach is well suited to autonomous systems which must take actions (such as storage decisions) without user interaction.

Two common approaches to ranked output generation are the vector space method and the probabilistic method, although variations abound. In the vector space method the range of c is [0,1], and the value is interpreted as the degree to which the content of two documents is similar. Both the profile and the documents are represented as vectors in a vector space, and a comparison technique based on the assumption that documents whose representations are similar to the profile will be likely to satisfy the associated information need is used. The angle between two vectors has been found to be a useful measure of content similarity, so the square of the cosine of that angle (easily computed as the normalized inner product of the two vectors) is used used to rank order the documents.

Experience has shown that the vector space method's effectiveness can be improved substantially by transforming the raw term-frequency vector in ways which amplify the influence of words which occur often in a document but relatively rarely in the whole collection. One common scheme, known as "term-frequency—inverse document frequency" weighting, assigns term i in document k a value computed as:

$$tfidf_{ik} = \text{Occurrences of term } i \text{ in document } k * \log_2\left(\frac{\text{Number of documents}}{\text{Number of documents with term } i}\right)$$

In a text filtering system, advance knowledge of the inverse document frequency portion of that equation is clearly not possible. Estimates of that information based on sampling earlier documents can, however, produce useful inverse document frequency values for domains in which term usage patterns are relatively stable.

Rather than estimate similarity, the probabilistic method seeks to estimate the *probability* that a document *satisfies* the information need represented by the profile. The probabilistic method is thus a generalization of the exact match technique in which we seek to rank order documents by the probability that they satisfy the information need rather than by making a sharp decision. To develop this probability, term frequency information (weighted to emphasize within document frequency and to deemphasize across-document frequency) is treated as an observation, and the distribution of the binary event "document matches profile" conditioned by that observation is computed. Bayesian inference networks have proven to be a useful technique for computing this conditional probability [41]. Since it is possible to construct a Bayesian inference net which computes the cosine of the angle between two vectors, the vector space method can be interpreted as a special case of the probabilistic method [42].

Since the comparison function can produce a multiple-valued result, the display module can be designed to exploit the results of both exact match and ranked output techniques. For example, an electronic mail system could reject documents sent by specific users and then rank the remaining documents in order of decreasing content similarity to a prototype document provided by the user. The combination of the profile and the comparison technique in a ranked output text filtering system can be though of as specifying a *point of view* in the document space. Multiple rank orderings can be combined to produce richer displays that combine multiple points of view, a research area often referred to as "document visualization" or "visual information retrieval interfaces."

Although only the vector space method actually uses vector operations such as the inner product, all three of these approaches exploit "feature vectors" in which the features are based on the frequency with which terms appear within documents and across the collection. The annotations provided by social filtering techniques are an additional source of features that can be exploited by a comparison function. Because annotations can be used even when useful content-based features are difficult to construct, information retrieval systems designed for information that is not in text form have explored matching techniques for feature vectors composed of annotations.

One such application which appears to have reached the critical mass necessary for effective use of annotations is a home video recommendation service developed by Hill and his colleagues at Bellcore in which users' tastes in movies were matched using techniques similar to those implemented in GroupLens [16]. Populated with a large and relatively stable set of movie titles, stable interests could be matched against that database for some time before exhausting the set of movies that might be of interest to a user. This is an interesting case in which the unlabeled corner of the graph in figure 1 is worth exploring.

Hill's system allowed users to provide numeric evaluations (on a scale of one to ten) for movies they had already seen, and then matched those ratings with evaluations of the same movies that had previously been provided by other users. Movies were sorted by category (e.g., drama or comedy), and within a category correlation coefficients between the feature vectors were computed. A set of users with the largest correlations was then selected and regression was performed based on evaluations from those users to predict scores for unseen movies in each category. In this case the profile was the set of annotations provided by the user, the "document" features were the annotations provided by others, and the comparison function was a two-step process of feature selection followed by regression.

In addition to showing how annotations can be viewed as features, this example illustrates an important limitation of the information retrieval techniques we have described. In information filtering applications, profiles based on multiple documents (such as the multi-movie evaluation within a category used in Hill's system) are common. But information retrieval research has explored only relatively simple ways of combining this information to form profiles. Relevance feedback, an information retrieval technique in which feature vectors are formed from the content of multiple documents, has shown good results. But the "one query at a time" model which underlies much information retrieval research precludes consideration of techniques such as the regression used by Hill and his colleagues.

5.2 User Modeling

Machine learning, the study of algorithms that improve their performance with experience, offers a source of techniques that are designed to exploit multiple training instances to improve selection effectiveness [22]. Machine learning is one component of "user modeling," a discipline which is concerned with both how information about users can be *acquired* and *used* by automated systems.[11] The models we consider in this report are what Rich has called "individual user, long-term user models" [33].

5.2.1 Sources of Information About the User

Before describing how machine learning techniques have been applied to text filtering it is useful to consider more carefully how information about the user can be acquired. Rich defined a distinction between "explicit" models which are "constructed explicitly by the user" and "implicit" models which are "abstracted by the system on the basis of the user's behavior" [33]. Both implicit and explicit user models are found in text filtering systems (SIFT, for example, uses an explicit model). The machine learning techniques we describe in section can be used to create what Rich called implicit models.

In order to construct an implicit user model we must be able to observe both the user's behavior and the salient features of the environment in which that behavior is exhibited. In the case of text filtering, the salient elements of the environment are the documents which have been examined by the user. Section 5.1 described how information about those documents can be acquired, either from contents or from annotations made by others.

In section 4 we presented several examples of how representations of previously seen documents can be combined with evidence of the user's interest in those documents to predict interest in future documents. With the exception of InfoScope, every system we have described requires the user to explicitly evaluate documents, a technique we refer to as "explicit *feedback*."[12] Explicit feedback has the advantage of simplicity. Furthermore, in experimental systems explicit feedback has the added advantage of minimizing one potential source of experimental error, inference of the user's true reaction. But in practical applications explicit feedback has two serious drawbacks. The first is that a requirement to provide explicit feedback increases the cognitive load on the user. This added effort works against one of the principal benefits of a text filtering system, the reduced cognitive load that results from an information space more closely aligned with the user's perspective. This problem is compounded by the observation that numeric scales may not be well suited to describing the reactions humans have to documents. For example, is a document which address the information need well but contains little expository text better or worse than a document that is easily understood but less complete? These difficulties motivate the study of implicit feedback mechanisms.

In his InfoScope system, Stevens observed three sources of implicit evidence about the user's interest in each message: whether the message was read or ignored, whether it was saved or deleted, and whether it was replied to or not. Because the users decision to read or ignore the message was necessarily based on a summary of the same message header information that InfoScope used to construct feature vectors, it would be reasonable to assume that the "read or ignore" decision would be nearly as useful

[11] As Karlgren and his colleagues have observed, it is also important to construct systems whose operation conforms with the user's mental model of the information filtering process [21]. The *user* models we refer to in this report, however, are models constructed by the system which describe some aspect of the user.

[12] There is some potential for confusion here because we are describing the use of explicit *feedback* to construct what Rich has called an implicit *user model*. In order to minimize confusion, we avoid using the terms "implicit" and "explicit" in isolation.

as explicit feedback. InfoScope did, however, allow explicit feedback as well.

Morita and Shinoda also investigated implicit feedback for filtering Internet News articles, using both save and reply evidence but substituting reading duration for InfoScope's "read or ignore" evidence [29]. In a six week study of eight users, they found a strong positive correlation between reading time and explicit feedback provided by the user on a four-level scale. Furthermore, they discovered that interpreting as "interesting" articles which the reader spent more than 20 seconds reading produced *better* recall and precision in a text filtering experiment than using documents explicitly rated by the user as interesting. This surprising result reinforces our observation that users sometimes have difficulty expressing their interest explicitly on a single numeric scale.

Since the experimental subjects were asked to read articles without interruption, it is not clear whether such useful relationships can be found in environments where reading behavior is more episodic. But Morita and Shinoda's results, coupled with the anecdotal evidence reported by Stevens, suggest that implicit feedback may be a practical source of features to which machine learning algorithms can be applied. Both implicit and explicit feedback produce features that are associated with documents. But unlike the feature vectors which describe the document's contents, feature vectors based on implicit or explicit feedback describe the user's *reaction* to the document.

5.2.2 Machine Learning

Complete feature vectors describing both the document and the user's reaction to it can be constructed for documents which have been read by adjoining the features that represent the document (e.g., term frequency values) with the vector that represents the user's reaction to it (e.g., explicit feedback). For new documents, only those features that represent the document will be known, and it would clearly be useful to be able to estimate the missing information (the user's anticipated reaction to the document). In the field known as "machine learning" this is known as the "supervised learning" problem.

In the canonical supervised learning problem, the machine is presented with a sequence of feature vectors (training instances), and then it is required to predict one or more missing elements in another set of feature vectors.[13] Predicting these missing values is an *induction* process, so induction forms the basis for machine learning. No induction technique can be justified without reference to domain knowledge, however. Because it would be possible to explain any set of observations after the fact, in the absence of some bias in the induction technique, any values could reasonably be predicted.[14] Langley identifies three ways in which this necessary bias can be introduced in a machine learning system: in the representation, in the search technique, and as explicit domain knowledge. [22] The vector space method, in which profiles are represented as a single vector and documents are ranked based on the angular similarity of their representation with that vector, combines both representation bias and search bias. InfoScope's learning heuristics (e.g., suggest filters for newsgroups that are read in at least 2 of the most recent 6 sessions) is an example of domain knowledge

[13]What we describe here is actually a restricted case of the supervised learning problem that is specialized to vector representations.

[14]One possible "after the fact explanation" would simply be that the formerly unknown parameters are random variables with some (still unknown) distribution that included the observed values.

bias.

Supervised learning is particularly well suited to exact match filtering systems which use explicit binary feedback, because in that case the training data contains exactly the same information (whether or not to select a document) that must be estimated for newly arrived documents. This is a special case of the "classification" problem, in which we wish to sort newly arrived documents into two or more categories (in this case, retained and rejected). Supervised learning can also be applied in ranked output filtering systems that use explicit feedback, assigning as a score for each document the system's estimate of the score that the user would assign. When implicit feedback is used, the ranking can be based on the predicted value of some observed parameter (e.g., reading duration). Alternatively, a manually constructed user model can be used to combine several observed parameters to produce an estimate of utility and then that estimate can be used to augment the training data.

Six classic machine learning approaches have been applied to text filtering: rule induction, instance based learning, statistical classification, regression, neural networks, and genetic algorithms. Stevens' work on InfoScope is an example of rule induction. InfoScope's filter suggestions were implemented as a decision list of parameters (newsgroup, field and word) which, if present in an article, would result in either selection or rejection of that article. These rules (e.g., select if newsgroup is rec.sewing and "bobbin" appears in the subject field) are learned using heuristics which can be modified by the user.

Foltz applied an instance based learning technique to selection of Internet News articles [9]. He retained representations of about 100 articles from a training collection which the user designated as interesting, and then ranked new articles by the cosine between their representation and the nearest retained representation. In other words, articles were ranked most highly if they were the most similar (using the cosine measure) to *some* positive example. In a small (four user) study, he found that this technique produced an average precision of 0.55 (43% above that achieved by random selection), and that a further improvement to 0.61 (11%) could be achieved using a dimensionality reduction technique known as Latent Semantic Indexing (LSI).

This dimensionality reduction is an example of "feature selection." Feature selection can be an important issue when applying machine learning techniques to vector representations. Langley has observed that "many algorithms scale poorly to domains with large numbers of irrelevant features," [22] and it is not uncommon to have thousands of terms in the vocabulary of a text filtering system. Schütze and others at Xerox PARC applied two rank reduction techniques, one using the best 200 terms found with a χ^2 measure of dependence between terms and relevant documents, and the other using a variation of the LSI dimension-reduction technique used by Foltz [35]. Four each of these feature selection techniques they evaluated four machine learning techniques, linear discriminant analysis (a statistical decision theory technique), logistic regression, a two-layer (linear) neural network, and three-layer (nonlinear) neural network using training and evaluation collections from TREC.

Schütze and his colleagues found that using only the LSI feature vectors provided the best filtering effectiveness with linear discriminant analysis and with logistic regression, and that their implementation of linear discriminant analysis was the better of the two techniques. They also found that both the linear and nonlinear networks were able to equal the effectiveness of linear discriminant analysis on the LSI feature vectors, but that both types of networks performed slightly (but not statistically significantly) better when presented with both sets of selected features simultaneously. Finally, they found that a nonlinear neural network resulted in no improvement over their simpler linear network.

Exploring another machine learning technique, Sheth implemented a genetic algorithm to filter Internet News in a system called "Newt." A genetic algorithm uses algorithmic analogues to the genetic crossover and mutation operations to generate candidate profiles that inherit useful features from their ancestors, and uses competition to identify and retain the best ones. Candidate profiles in Newt were vectors of term weights.[15] Relevance Feedback based on explicit binary evaluations of articles was used to improve candidate profiles, moving them closer in the vector space to the representation of desirable articles and further from the representation of undesirable ones. In machine learning this approach is referred to as "hill climbing." The crossover operator was periodically applied to combine segments of two candidate profiles which were among those that had produced the highest ranks (using a cosine similarity measure) for articles that the user later identified as desirable. A mutation operator was sometimes applied to the newsgroup name to explore whether existing candidate profiles would perform well on newsgroups with similar names. All of the candidate profiles contributed to the ranking of the documents shown to the user, although those which consistently performed well contributed more strongly to the ranking. Hence, the profile itself was determined by the *population* of candidate profiles, rather than by any individual candidate.

Sheth evaluated Newt using a technique referred to in machine learning as a "synthetic user." By generating (rather than assessing) user preferences, the synthetic user technique allows specific aspects of a machine learning algorithm's performance (e.g., learning rate) to be assessed. Sheth created synthetic users whose interests were deemed to be satisfied whenever at least one word from a list associated with that simulated user appeared in an article. Using this technique he found that although individual candidate profiles were able to learn to satisfy a simulated user quickly, when the simulated user's interest shifted abruptly (simulated by changing the list of words associated with the simulated user) individual candidate profiles were slower to adapt. When evaluating complete profiles made up of populations of individual candidates, Sheth demonstrated the ability to control the adaptation rate by adjusting parameters of the genetic algorithm. Simulated users lack the sophistication of human relevance judgements, but the technique is both economical and reproducible, so it is useful for certain types of evaluations.

5.3 Other Fields

This completes our description of the two major sources of technology for text filtering systems: information retrieval and user modeling. Humans pursue the information filtering process in a social context, though, and the machines that they use must operate in some physical context. In this section we briefly identify the issues raised by the interaction between the information filtering process and these larger contexts.

[15]In Newt, terms were segregated by the field of the article in which they occurred, so "talk" in the subject field could be assigned a different weight than "talk" in the body of a message.

5.3.1 Networked Computing Infrastructure

The physical context for the information filtering process is the existing networked computing infrastructure. The relevant portion of the physical context may consist of, for example, isolated workstations monitoring a common newsfeed, a workgroup computing environment supported by an intranet, or the entire Internet. With a few notable exceptions (SIFT and Tapestry), in our descriptions we have placed more emphasis on effectiveness than efficiency when describing design features and performance evaluations. This is not surprising, since most experimental work on text filtering has sought to demonstrate effectiveness and a small user population suffices for that purpose. Even the TREC evaluation, which requires filtering hundreds of thousands of pages of text, specifies only 50 topics each year.

Once adequate effectiveness has been demonstrated for small user populations, the task of engineering efficient implementations for widespread use of such systems remains. One alternative is to simply replicate the filtering system and then provide all of the content to each filtering system. Tapestry implemented a more sophisticated approach, demonstrating that an appropriate division of effort between server-side and client-side computing can improve overall efficiency.

In general, he goal of distributed computation is to optimize the tradeoff between distributing the workload and minimizing communication requirements. Yan studied this issue rigorously in conjunction with his work on SIFT, developing optimal assignments of computational tasks among a group of cooperating servers [45]. The GroupLens project has chosen an alternative approach that exploits an existing infrastructure for document distribution. By augmenting this infrastructure with distributed annotation servers, GroupLens expects to achieve acceptable efficiency in a manner compatible with the existing physical and social structure for Internet News. Thus, one of the key issues to be addressed as the number of users scales up is which constraints to accept and which to attempt to change.

5.3.2 Computer Supported Cooperative Work

The same type of tension between constrained and unconstrained system design occurs at many levels. Adopting an even broader perspective, it is apparent that users operate within a social system, and that system imposes social constraints on what is possible. Organizational aspects of networked communications are studied the field of Computer Supported Cooperative Work (CSCW), so text filtering is an issue for which the CSCW perspective can be informative.

Consider, for example, Denning's suggestion that users set up separate mailboxes for specific purposes and that senders direct electronic mail to the appropriate mailbox. In order to be effective, this approach would require that the user address messages correctly, that receivers organize their mailboxes in a useful manner, and that all of the software systems between the sender and the receiver support this addressing scheme. Standards development processes and competitive market mechanisms are two techniques for addressing such issues, and there are numerous examples of the practicality of such schemes (e.g., Lotus Notes and Internet News). Because many of the constraints on such efforts are social rather than technical, the breadth offered by the CSCW perspective is essential to the success of such endeavors.

Once such social conventions are created to add the necessary structure to the documents, text filtering techniques provide a way to exploit that information. For example, the current interest in assigning "ratings" to World Wide Web pages to facilitate parental control of the information available to their children presumes the availability of technology to exploit that information. The design a system for creating, distributing, and using these ratings is an issue best studied from the perspective of CSCW because a common task motivates multiple participants. Ratings are, however, simply one type of annotation. So an understanding of how annotations are used in information filtering systems can provide useful insight into how those annotations could be integrated with other sources of information about the contents of a document.

5.3.3 Market Formation

For applications which lack a shared objective, economic theory provides a more useful perspective than CSCW. In a market economy, "cost" or "price" (the value discovered by a market) serves as a basis for allocating scarce resources. In the emerging information-based economy, both information itself and the tools which manage that information have economic value. This will result in the development of a market for not merely information and tools, but also for metainformation such as the annotations on which social filtering is based. The CSCW perspective will certainly be helpful when designing common standards for the exchange of price information and monetary instruments because all participants in a market benefit from such social structures. But when participants do not share common goals with respect to the *use* they make of the information they obtain, market dynamics provide a more effective way of allocating scarce information resources such as intellectual property and expert annotations.

The vast majority of experimental work on text filtering has exploited freely available information such as Internet News and messages sent to electronic mailing lists, so little reference to the cost of intellectual property can be found in that literature. On the other hand, users of commercial text filtering systems have developed profile construction techniques which which recognize differing costs for different aspects of access to intellectual property (e.g., selective purchase of limited redistribution rights) [8]. Commercial text filtering systems typically require explicit profiles, however, and we are not aware of any research on implicit user models for text filtering which exploit cost information. Like the ratings we described in section 5.3.2, prices are a type of annotation, and hence they can be exploited by a social filtering system. The difference between prices and other annotations on which social filtering can be based is that there may be a firmer *a priori* basis for using cost information than for using other types of annotations, and that fact may prove useful when designing user models for text filtering.

In addition to these technical considerations, market formation also raises broad social issues. The creation of markets for information, for annotations, and even for the filtering systems themselves *restricts* information access to users for whom the value of the information justifies the cost of obtaining it. Such unrestrained market operation is rarely allowed, however. Governments and other social structures are often charged with regulation of economic activity in order to limit the effect of inequities that can result from market economics. The establishment of public libraries, the imposition of disclosure requirements for securities transactions, and the regulations which subsidize universal access to the telephone network with revenue generated from other sources provide instructive examples of how market forces can be adjusted to accomplish social goals. If information truly has value then such issues of equity will undoubtedly arise in information filtering as well.

5.3.4 Privacy

Privacy becomes an issue when a system collects information about its user, so important social issues arise on an individual scale as well. In commercial applications, for example, it may be desirable to restrict access to profile information in order to protect a competitive advantage. And users with personal applications may demand that their profile remain private simply on moral grounds.

For content-based filtering systems, the privacy issue has two aspects: preventing unauthorized access to the profile and preventing reconstruction of useful information about the profile. The first issue is a straightforward security problem for which a variety of techniques such as password protection and encryption may be appropriate depending on the nature of the anticipated threat. But preventing reconstruction of useful information about the profile is a much more subtle problem. In Tapestry, for example, it would be possible to infer a good deal of information about the profile registered at the server by simply noting which documents were forwarded. An unauthorized observer who can detect which documents are being forwarded to specific users could conceivably build a second text filtering system (e.g., a social filter with an implicit user model) and then train it using the observed document forwarding decisions. Preventing such an attack would require that unauthorized observers be denied access to information about the sources and destinations of individual messages. In the computer security field, this is known as the "traffic analysis problem," and cryptographic techniques which address it have been devised (c.f., [5, 6]).

In the case of collaborative filtering, the situation is further complicated by the imperative to share document annotations. A simple approach (which is used by GroupLens) is to allow each user to adopt a pseudonym. While use of pseudonyms makes it more difficult to associate annotations with users, traffic analysis can still be used to determine which users would read a document. Unfortunately, information about who is reading specific documents is exactly what other authorized users must know to perform social filtering. Furthermore, Hill has observed that users choosing which information to examine may find it useful to know the identity (not merely the pseudonym) of the users who made the annotations [16]. While encrypted transmission of annotations to other authorized users is a possibility in such cases, significantly limiting the user group in that way may prevent a social filtering system from reaching the necessary critical mass. This tension between a desire for privacy and the benefit of free exchange of information may ultimately limit the applications to which social filtering can be applied.

The level of protection which must be afforded to privacy varies widely across applications. By common agreement, many details of our private lives (e.g., birth, marriage and death) are a matter of public record. On the other hand, in the state of Maryland it is a crime to divulge the borrowing history of a library patron without a court order. One can even envision applications in which a user might prefer not to know information represented in their own profile. Where these lines should be drawn is a matter of judgement that must ultimately be resolved by those who control the information resources that are being used.

6 Observations on the State of the Art

Early information filtering systems (then known as SDI) were developed to exploit the availability information in electronic form to manage the process of disseminating scientific information. When the printed page was the dominant information paradigm for text transmission, high production costs led to the development of extensive social structures (e.g., the peer review process) for selecting information worthy of publication. As long as this situation persisted, the dissemination process managed admirably, and SDI improved its performance. With the introduction of personal computing and ubiquitous networking, each participant is now able to also be both a consumer and a producer of information. The drastic reduction in publishing costs has greatly increased the importance of filtering the resulting flood of information, but the resulting variability document quality has also made that filtering task more difficult. Automatic techniques are needed to make this wealth of information accessible, since information that cannot be found is no better than information which does not exist.

Rather than simply removing unwanted information, information filtering actually gives consumers the ability to reorganize the information space [38]. For economic reasons, information spaces have traditionally been organized by producers and, in some cases, reorganized by intermediaries. In book publishing, for example, authors and publishers work together to assign titles to books and to announce their availability. Intermediaries such as libraries, book clubs and book stores obtain those announcements, select items which are likely to be of interest to their customers, and organize information about their selections in ways that serve the needs of those customers. Because such intermediaries typically serve substantial numbers of customers, economic factors usually limit them to providing a few (sometimes only one) perspectives on the information space.

Information filtering is essentially a personal intermediation service. Like a library, a text filtering system can collect information from multiple sources and produce an organization that is useful to the patrons. But by automating the process of organizing the information space it becomes economically feasible to personalize this organization. Of course, automating this intermediation process eliminates the value that could be added by human intermediaries who can apply their judgement to improve the organization of the information space.

Social filtering offers a way of integrating human and automated intermediation. Human intermediaries have traditionally organized the information space through selection and annotation. Selection, however, is simply a special type of annotation (i.e., a document is marked as "selected by the intermediary"). As with price annotations, the user may find it useful to assign expert annotations an *a priori* degree of confidence because they come from a source with well understood characteristics. Tapestry's profile specification language provides an example of how such functionality could be incorporated.

Social filtering alone is unlikely to provide a complete solution to users' information filtering needs. Expert annotations require effort and have economic value, so the marketplace will undoubtedly assign them a price. With continued reductions in the cost of computing and communications resources, content-based filtering will offer a competitive source of information on which to base selections. Furthermore, because humans and machines base their evaluations on different features, systems which incorporate both social and content-based filtering will likely be more effective than those which use either technique in isolation. In this light, the work of Schütze and his colleagues suggests that machine learning techniques which effectively exploit multiple sources of evidence can be found [35].

Content-based and social filtering will almost certainly prove to be complementary in other, less easily measured ways as well. A perfect content-based technique would never find anything novel, limiting the range of applications for which it would be useful. Social filtering techniques excel at identifying novelty (because they are guided by humans), but only when the humans who guide them are not overloaded with information. Content-based systems can help to reduce this volume of information to manageable levels. Thus, both content-based and collaborative filtering contribute to the other's effectiveness, allowing an integrated system to achieve both reliability and serendipity.

Social filtering has yet to realize this potential, however. The difficulty of achieving a critical mass of participants makes social filtering experiments expensive. One clear disincentive in present experiments is the additional cognitive load imposed on the user by the requirement to provide explicit feedback. We are not aware of any research in which implicit feedback has been applied to social filtering, but there is some evidence that such an approach could be successful. Hill and his colleagues have reported that readers find it useful to know which portions of a document receive the most attention from other readers. In an analogy to the tendency of well-used paper documents to acquire characteristics which convey similar information, they call this concept "read wear" [15]. Coarser measurements such as Morita and Shinoda's reading time metric, or the save and reply decisions explored by Stevens, may also prove to be useful bases for social filtering in some applications. If useful annotations can be acquired without requiring explicit feedback, lesser inducements (such as the improvement that could result from application of a simple content-based filtering technique) may be sufficient to assemble the critical mass of users needed to evaluate social filtering techniques.

Another serious impediment to the large scale evaluation of social filtering techniques is the difficulty of constructing suitable measures of effectiveness. Recall, precision and fallout are of some use when comparing content-based filtering techniques, but their reliance on normative judgements of document relevance suppresses exactly the individual variations that social filtering seeks to exploit. One feasible evaluation technique would be to apply simulated users like those used by Sheth to investigate specific aspects of collaborative behavior. Important issues such as the learning rates and variability in learning behavior across large heterogeneous populations could be investigated with large collections of simulated users whose design was tailored to explore those issues.

Another alternative is to study situated users (i.e., human users performing self-directed tasks), attempt to provide them with desirable documents, and then measure something related to their satisfaction. Those "dependent variables" could certainly be the sort of explicit feedback commonly required in present social filtering experiments, but insisting on explicit feedback increases the difficulty of assembling a sufficiently large user population. In suitable sources of implicit feedback can be identified, those same measures would would be a far better choice for the set of dependent variables. Such an experiment design requires that separate training and evaluation document collections be used, a feature easily introduced by withholding implicit feedback from the filtering algorithm during the evaluation period. This approach can be used to evaluate both content-based and social filtering systems, so it would be a natural choice when evaluating systems which applied both types of techniques. It can only be applied, however, after suitable sources of implicit feedback are found. Since implicit feedback has the potential for a high payoff in performance evaluation, filtering effectiveness, and user satisfaction, research on that topic should be accorded a high priority.

7 Conclusion

Designers of text filtering systems can benefit from research in text retrieval, user modeling and a number of other fields. Text filtering is, however, a unique information seeking process that is distinguished by a focus on satisfying relatively stable interests in documents containing text. This report has reviewed progress in the field with particular emphasis on the selection component of the filtering process. Other useful perspectives are offered by Jiang [20], Mock [28], Stevens [38], and Wyle [44].

Text filtering systems must develop representations of both documents and user interests, they must be endowed with some way of comparing documents with interests, and they must possess some way of using the results of those comparisons to assist the user with document selection. Text retrieval research has produced a number of content-based representations that use the frequency with which terms appear in documents, and social filtering research has produced a complementary set of features based on shared annotations from other users. When combined with implicit or explicit feedback from the user about the documents they have examined, those representations provide a basis for construction of profiles which represent the user's interests. Both text retrieval and machine learning offer techniques for comparing document representations with profiles, and this is an area of active research. Document visualization is another dynamic research area, but ranked output presently offers a simple way of synergistically exploiting the strengths of human and machine to facilitate the filtering process.

The text filtering techniques described in this report offer a range of solutions that can help users achieve their information seeking goals. With technology presently in hand, designers can produce effective and efficient systems that will be useful in a number of applications. Furthermore, the present research on applications of user modeling, implicit feedback, shared annotations and document visualization to text filtering suggests that text filtering technology will have even greater impact in the future. As the quantity of online information continues to increase, text filtering will provide an increasingly important technique for bringing together producers and consumers of information.

Acknowledgement

The authors would like to express their appreciation to Bonnie Dorr, Stuart Stubblebine, Vigil Gligor and John Riedl for their useful comments.

References

[1] Paul E. Baclace. Competitive agents for information filtering. *Communications of the ACM*, 35(12):50, December 1992.

[2] Nicholas J. Belkin and W. Bruce Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, December 1992.

[3] D. C. Blair. *Language and Representation in Information Retrieval.* Elsevier, Amsterdam, 1990.

[4] T. F. Bowen, G. Gopal, G. Herman, T.Hickey, K.C. Lee, W. H. Mansfield, J. Raitz, and A. Weiribnrib. The datacycle architecture. *Communications of the ACM*, 35(12):71–80, December 1992.

[5] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[6] David A. Cooper and Kenneth P. Birman. Preserving provacy in a network of mobile computers. In *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pages 26–38. IEEE Computer Society, May 1995. http://cs-tr.cs.cornell.edu.

[7] Peter J. Denning. Electronic junk. *Communications of the ACM*, 25(3):163–165, March 1992.

[8] Barbara Denton. Ten ways to control dialog alert costs. *Online*, 19(2):47–48, March 1995.

[9] Peter W. Foltz. Using latent semantic indexing for information filtering. In Frederick H. Lochovsky and Robert B. Allen, editors, *Conference on Office Information Systems*, pages 40–47. ACM, April 1990. http://www-psych.nmsu.edu/~pfoltz/cois/filtering-cois.html.

[10] Peter W. Foltz and Susan T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, December 1992. http://www-psych.nmsu.edu/~pfoltz/cacm/cacm.html.

[11] David Goldberg, David Nicholas, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.

[12] Donna Harman. The DARPA TIPSTER project. *ACM SIGIR Forum*, 26(2):26–28, Fall 1992.

[13] Donna Harman. Overview of the first TREC conference. In Robert Korfhage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 36–47. ACM, June 1993.

[14] Donna Harman. Overview of the third Text REtrieval Conference (TREC-3). In D. K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 1–19. NIST, U.S. Department of Commerce, 1994. NIST Special Publication 500-225. http://potomac.ncsl.nist.gov/TREC.

[15] W. C. Hill, J. D. Hollan, D. Wroblewski, and T. McCandless. Read wear and edit wear. In *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI '92*, pages 3–9. ACM Press, 1992.

[16] Will Hill, Mark Rosenstein, and Larry Stead. Community and history-of-use navigation. In *Electronic Proceedings of the Second World Wide Web Conference '94*. National Center For Supercomputer Applications, Software Development Group, October 1994. Not available in print. http://community.bellcore.com/navigation/home-page.html.

[17] Lynette Hirschman. Comparing MUCK-II and MUC-3: Assessing the difficulty of different tasks. In *Proceedings, Third Message Understanding Conference (MUC-3)*, pages 25–30. DARPA, Morgan Kaufmann, May 1991.

[18] Edward M. Housman. Survey of current systems for selective dissemination of information. Technical Report SIG/SDI-1, American Society for Information Science Special Interest Group on SDI, Washington, DC, June 1969.

[19] Paul S. Jacobs and Lisa F. Rau. SCISOR: Extracting information from on-line news. *Communications of the ACM*, 33(11):88–97, November 1990.

[20] Zhenglian Jiang. Understanding information filtering and providing and information filtering system model. Master's thesis, University of Missouri, Kansas City, December 1993.

[21] Jussi Karlgren, Kristina Hook, Ann Lantz, Jacob Palme, and Daniel Pargman. The glass box user model for filtering. Technical Report T94:09, Swedish Institute of Computer Science, July 1994. http://mars.dsv.su.se/~fk/if_Doc/JPfilter-filer/Glassbox1.1.ps.Z.

[22] Pat Langley. *Elements of Machine Learning*. Morgan Kaufmann, San Francisco, 1996.

[23] Wendy Lehnert and Beth Sundheim. A performance evaluation of text analysis technologies. *AI Magazine*, 12(3):81–94, Fall 1991.

[24] Shoshana Loeb. Architecting personalized delivery of multimedia information. *Communications of the ACM*, 35(12):39–48, December 1992.

[25] H. P. Luhn. A business intelligence system. *IBM Journal of Research and Development*, 2(4):314–319, October 1958.

[26] Thomas W. Malone, Kenneth R. Grant, Franklyn A. Turbak, Steven A. Brobst, and Michael D. Cohen. Intelligent information sharing systems. *Communications of the ACM*, 30(5):390–402, May 1987.

[27] Gary Marchionini. *Information Seeking in Electronic Environments*. Cambridge University Press, Cambridge, 1995.

[28] Kenrick Jefferson Mock. *Intelligent Information Filtering via Hybrid Techniques: Hill Climbing, Case-Based Reasoning, Index Patterns, and Genetic Algorithms*. PhD thesis, Univeristy of California Davis, 1996. http://phobos.cs.ucdavis.edu:8001/~mock/infos/infos.html.

[29] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In W. Bruce Croft and C.J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 272–281. Springer-Verlag, July 1994. http://www.jaist.ac.jp/jaist/is/labs/shinoda-lab/papers/1994/sigir-94.ps.

[30] Stephen Pollock. A rule-based message filtering system. *ACM Transactions on Office Information Systems*, 6(3):232–254, July 1988.

[31] Ashwin Ram. Natural language understanding for information filtering systems. *Communications of the ACM*, 35(12):80–81, December 1992. ftp://ftp.cc.gatech.edu/ai/ram/er-92-08.ps.Z.

[32] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In Richard K. Faruta and Christine M. Neuwirth, editors, *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, October 1994. http://www.cs.umn.edu/Research/GroupLens/cscwpaper/paper.html.

[33] E. A. Rich. User modeling via stereotypes. *Cognitive Science*, 3:329–354, 1979.

[34] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, New York, 1983.

[35] Hinrich Schütze, David A. Hull, and Jan O. Pedersen. A comparison of classifiers and document representations for the routing problem. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 229–237, July 1995.

[36] Irene Stadnyk and Robers Kass. Modeling users' interests in information filters. *Communications of the ACM*, 35(12):49–50, December 1992.

[37] Curt Stevens. Automating the creation of information filters. *Communications of the ACM*, 35(12):48, December 1992. http://www.holodeck.com/curt/mypapers/CACM-12-92.ps.

[38] Curt Stevens. *Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces.* PhD thesis, University of Colorado, Department of Computer Science, Boulder, 1992. http://www.cs.colorado.edu/homes/stevens/public_html/mypapers/Thesis-tech-report.ps.

[39] Robert S. Taylor. The process of asking questions. *American Documentation*, 13(4):391–396, October 1962.

[40] Douglas B. Terry. A tour through tapestry. In *Proceedings of the ACM Conference on Organizational Computing Systems (COOCS)*, pages 21–30, November 1993.

[41] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In Jean-Luc Vidick, editor, *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*, pages 1–24. ACM SIGIR, September 1990.

[42] Howard R. Turtle and W. Bruce Croft. A comparison of text retrieval models. *The Computer Journal*, 35(3):279–290, 1992.

[43] M.F. Wyle and H.P. Frei. Retrieving highly dynamic, widely distributed information. In N. J. Belkin and C.J. van Rijsbergen, editors, *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, pages 108–115. ACM, June 1989.

[44] Mitchell F. Wyle. *Effective Dissemination of WAN Information.* PhD thesis, LaSalle University, Mandeville, LA, 1995. http://vhdl.org/~wyle/diss/diss.html.

[45] Tak W. Yan and Hector Garcia-Molina. Distributed selective dissemination of information. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, pages 89–98. IEEE Computer Society, September 1994. ftp://db.stanford.edu/pub/yan/1994/dsdi.ps.

[46] Tak W. Yan and Hector Garcia-Molina. SIFT — A tool for wide-area information dissemination. In *Proceesings of the 1995 UNSENIX Technical Conference*, pages 177–186, 1995. ftp://db.stanford.edu/pub/yan/1994/sift.ps.

A note on the references

Where Uniform Resource Locators (URL) are included in the citation, they were believed to be correct at the time of publication but may have changed since. Current links to to every online information filtering reference of which we are aware (including those filtering types of information other than text) can be found on the World Wide Web at http://www.ee.umd.edu/medlab/filter/. The first author would appreciate being notified of additional online resources or changed URL's by electronic mail.