The State of the Art in Text Filtering

DOUGLAS W. OARD *

University of Maryland, College Park, MD, U.S.A.

Abstract.

This paper 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 text 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. User modeling techniques drawn from information retrieval, recommender systems, machine learning and other fields are described. The paper concludes with observations on the present state of the art and implications for future research on text filtering.

Key words: Information filtering, Text retrieval, Social filtering, Collaborative, Content-based, Selective Dissemination of Information, Current awareness, Recommender systems

* Digital Library Research Group, College of Library and Information Services, University of Maryland, College Park, MD 20742, oard@glue.umd.edu

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 paper reviews existing work on text filtering, a type of "information seeking." We use "information seeking" as an overarching term to describe any processes by which users seek to obtain information from automated information systems (Marchionini, 1995). In the "information filtering" process the user is assumed to be seeking information which addresses a specific long-term interest. In this paper we introduce the information filtering problem in some detail and describe the specific techniques used 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, detecting useful information sources, and displaying the useful 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. As Belkin and Croft have observed, this makes the text filtering process an attractive application for techniques that were originally developed to support the text retrieval process (Belkin and Croft, 1992). In recognition of that commonality, we use the term "detection" when we wish to refer generally to techniques which could be used for either filtering or retrieval.

Figure 1. Information seeking task diagram.

1.1. THE PROCESS PERSPECTIVE

The distinction between process and system is fundamental to understanding the 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 the creative application of techniques that were designed originally to support one type of information seeking process (e.g., information retrieval) to support a different type of information seeking process(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 exploration), 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 sequences 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 detection 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. Thus, information filtering can be viewed as an application of user modeling techniques to facilitate information detection in dynamic environments.

1.2. COLLECTION AND DISPLAY

This paper describes the design of systems to support the text filtering process, with particular emphasis on the text detection 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 detection and the other two components depicted in figure 1. The paper by Winiwarter,

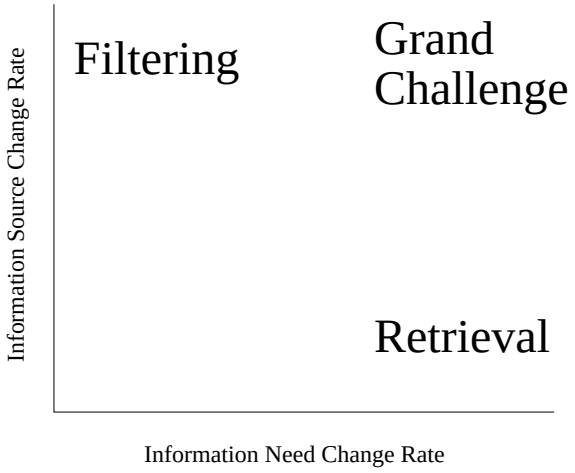

Figure 2. Information detection processes.

et al. in this issue provides an excellent example of how all three components can be integrated into an effective and usable system (Winiwarter et al., 1997).

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 through some combination of the two. Early descriptions of the information filtering problem implicitly assumed passive collection (Housman, 1969; Denning, 1982). As the amount of electronically accessible information has exploded, active collection has become increasingly important (c.f. (Wyle, 1995; Pizzani et al., 1996)). Active collection techniques can benefit from a close coupling between the collection and detection modules because they exploit models of both the user and the networked information resources 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 an active collection module must choose whether to obtain information before that information is known, the user model for the detection module which seeks to choose the information to retain will likely differ from the collection module's user model in important ways. For example, the source of the information is one of a small number of useful facts to consider when building the user model for the collection module, but that is only one of many factors which could be considered when building the user model for the detection component. 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 detection component and because there has been little reported on how the two components can be integrated.

Such a clean division is not as easy to construct for the interface between the detection 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(c.f. (Winiwarter et al., 1997)). 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 (called "exact match" detection) 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 detection module, however. Because the display module must rank the documents, the detection module must provide some basis (e.g., a numeric "status value") from which the ranking can be constructed. Human-computer interaction is a rich research area in its own right, but our discussion of the issue will be limited to those aspects of display design that impose requirements on the detection module.

1.3. Other Information Seeking Processes

We have already mentioned information filtering and retrieval, but there are other information seeking processes for which the decomposition in Figure 1 is appropriate. Table I lists some examples of information seeking processes. One familiar type of information seeking is the process of retrieving information from a database. A distinguishing feature of the database access process is that the output will *be* information, while in information filtering (or retrieval), the output is a set of entities (e.g., documents) which *contain* the information that is sought (Blair, 1990). For example, searching book titles in an online library catalog (a type of database) to find the author of a specific book would be a database access process. On the other hand, occasionally using the same library catalog to discover whether any new

books on a certain topic have been added to the collection by searching for keywords in the title field would be an information filtering process. As this example shows, database systems can be used to support an information filtering processes, and we will present examples of such an approach in section 3.

Table I. Examples of information seeking processes.

Process	Information Need	Information Sources
Information Filtering	Stable & Specific	Dynamic & Unstructured
Alerting	Stable & Specific	Dynamic & Structured
Data Mining	Stable & Specific	Stable
Information Retrieval	Dynamic & Specific	Stable & Unstructured
Database Access	Dynamic & Specific	Stable & Structured
Exploration	Broad	Varied

An interesting variation on the database access processes is what is sometimes referred to as "alerting." For an alerting process the information need is assumed to be relatively stable with respect to the rate at which the information itself is changing. The classic example is an alarm which informs the operator of a complex machine whenever a parameter exceeds prespecified limits. Alerting is thus the database analogue of information filtering, since the only difference between the two is that in an information filtering process it is the information sources (e.g., documents) rather than the information itself which are arriving rapidly. Alerting systems are useful for information filtering because an activity such as monitoring an electronic mailbox and informing the user whenever mail from a specific user arrives is easily cast as an alerting process.

Another closely related research area is what is becoming known as "data mining," the search for useful information in large collections of data. Text retrieval systems seek to organize collections of documents in ways that facilitate retrieval, so in a sense text retrieval systems all exploit simple data mining techniques. Text filtering systems also share some common techniques with data mining. In section 4.2 we describe how annotations assigned by early readers of individual documents can be used as a basis for predicting the interest of later readers in those documents. The same techniques have been applied by Hill, *et al*. to relatively static collections with some success (Hill et al., 1994).

Finally, "exploration" is a somewhat less well structured information seeking process for which the decomposition shown in figure 1 may be appropriate. Since users might choose to explore either static or dynamic information sources, exploration has aspects similar to both information filtering and information retrieval. "Surfing the World Wide Web" is an example of exploring relatively static information, while reading an online newspaper would be an example of exploring relatively dynamic information. When comparing exploration to information filtering and retrieval, the important distinguishing feature of the exploration process is

that the users' interests are assumed to be broader than for an information filtering or retrieval processes (Marchionini, 1996). Precisely what is meant by "broader" is difficult to define, however, and the distinction is often simply a matter of judgment. In order to sharpen the distinction for the purpose of this paper, we propose an operational definition of exploration. 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 exploration. In other words, we propose that developers of information filtering systems 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 the filtering and exploration processes for users of their system.

1.4. 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 "recommendation." "Recommendation" has recently been proposed as a general term to describe both filtering and retrieval processes that exploit the opinions of other users (Resnick and Varian, 1997). 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 it is sometimes meant to imply that an all-or-nothing (i.e., unranked) detection 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 detection of new information based solely on the title of a journal, magazine, or other serial publication. Most of these interpretations have some historical basis, but it is not uncommon to find the terms used to describe systems which lack the distinguishing characteristics of their historical antecedents. We shall avoid this problem by referring to all of these variations as "information filtering."

The term "filtering" is also sometimes used differently, even in closely related fields. For example, in machine learning, the feature selection step that we describe in section 4.4 is sometimes referred to as "information filtering." Perhaps the greatest potential for confusion results from the use of "filtering" to describe techniques for customizing the user's view of a large quantity of relatively stable information. Such usage can be found in the literature on social filtering and on adaptive hypertext. The potential for confusion in these cases is particularly acute

because such applications often rely on text detection techniques that are similar to those used to support the text filtering process. But it is the process perspective that makes it possible to sort out what sense of "filtering" is intended. In this survey we use the term "filtering" only to describe the process identified in section 1.1 and systems which are intended to support that process.

Taylor defined four types of information need (visceral, conscious, formalized, and compromised) that reflect 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 (Taylor, 1962). In common use, however, application of the terminology is rarely 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 also used to describe the formalized (i.e., the human expression of) the information need. "Query" is the traditional term for Taylor's concept of a compromised information need that could be submitted to an information retrieval system, but in some experimental work it is the visceral information need that is referred to as the "query." In this paper, 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 an information filtering context as well. Sometimes a collection of profiles is referred to as a "user model." In our view that usage is somewhat imprecise because we prefer to think of the user model as including both the representation of the user's interests and some means for interpreting that representation to make predictions. 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. Table II summarizes the terminology that we have adopted for this survey.

Table II. Information filtering terminology.

Preferred Term	Other Commonly Used Terms
Filtering	Routing, SDI, Current awareness, Recommendation
Information need, Interest	Visceral information need
Topic	Formalized information need
Profile	Compromised information need (filtering)
Query	Compromised information need (retrieval)

2. Historical Development

Luhn introduced the idea of a "Business Intelligence System" in 1958 (Luhn, 1958). In Luhn's concept, library workers would create profiles for individual users, and then those profiles would be used in an exact-match text detection 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, Luhn 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 detection module as "selective dissemination of new information" Luhn 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 (Housman, 1969). These systems generally followed Luhn's model, although only four of the 60 implemented any type of 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 the further development of 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 an ACM President's Letter that appeared in the Communications of the ACM in March of 1982 (Denning, 1982). 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 the 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." Five of the remaining six ideas (hierarchical organization of mailboxes, separate private mailboxes, special forms of delivery, threshold reception, and quality certification) all would have required the cooperation of the sender, and thus must properly be studied from a perspective more global than the receiver's scope of action represented by the information

seeking model in figure 1. We shall have more to say on such perspectives in section 4.5.

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,* and broader network resources (Pollock, 1988; Wyle and Frei, 1989; Foltz, 1990; Jacobs and Rau, 1990). The most influential paper of this period was published in the Communications of the ACM by Malone, *et al*. in 1987 (Malone et al., 1987). There they introduced three paradigms for information detection, "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 also had implications beyond the scope of the information seeking task model depicted in figure 1. We describe the economic issues related to information filtering briefly in section 4.5.3.

The most important contribution of Malone, *et al.* was to introduce an alternative approach which they called social (and which is 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. 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. If it proves to be practical, social filtering could provide a basis for detection of information items, even if their content can not be represented in a way that is useful for detection. The balance between content-based and social 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) (Lehnert and Sundheim, 1991; Hirschman, 1991). One principal thrust of MUC has been use of information extraction techniques to support the detection of messages. In 1990, DARPA launched the TIPSTER project to fund research on this topic (Harman, 1992). TIPSTER also added a second task, the application of statistical techniques to preselect messages which could then be subjected to more sophisticated natural language processing. In 1992 The National

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

Institute of Standards and Technology (NIST) capitalized on the TIPSTER test collection by co-sponsoring (with DARPA) an annual Text REtrieval Conference (TREC) focused specifically on text filtering and retrieval (Harman, 1993).

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. But 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 number of researchers 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 detection, application domains, hardware and software architectures, privacy, and case studies. A year later, in December of 1992, nine papers that resulted from that workshop appeared in a special issue of the Communications of the ACM (Baclace, 1992; Belkin and Croft, 1992; Bowen et al., 1992; Foltz and Dumais, 1992; Goldberg et al., 1992; Loeb, 1992; Ram, 1992; Stadnyk and Kass, 1992; Stevens, 1992a).

3. 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.* 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 approaches that operate on behalf of the receiver (rather than the sender) and which illuminate what we feel are the most important issues to guide system implementation decisions and further research.

3.1. 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 called SIFT (Yan and Garcia-Molina, 1995). Profiles for SIFT were constructed manually by specifying words to prefer

* Network-accessible resources on information filtering that are known to the author are collected at http://www.clis.umd.edu/dlrg/filter

or avoid, and had to be be updated manually if the user desired to change them. For each profile, twenty articles were made available each day in a ranked output format. Articles could be selected interactively using a World Wide Web browser. For users lacking interactive access, clippings (the first few lines of each article) could instead be sent by electronic mail. In that case detection was done without user interaction, so users were offered the option of defining a profile for an exact match text detection technique.

SIFT offered two facilities to assist users with profile construction. Users were 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 was present in Stanford's Internet News server that day, iterative refinement allowed the user to construct a profile which would 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 were 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 occurred with meanings that were unforeseen at the time the profile was constructed, users could select additional words which appeared 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 SIFT, large collections of profiles were compared to every article arriving on Internet News by a central server. Efficiencies were obtained by grouping profiles in ways that permit parts of the filtering process to be performed on groups of profiles rather than individually. SIFT made no distinction among the words appearing in an article, so words appearing in the article title, the body of the article, included text, or the "signature" information that is routinely added to every document by some users were all equally likely to result in a high rank for a document.

Some commercial text filtering systems also rely on the manual profile construction technique. The "Fast Data Finder," a product of Paracel, Inc., uses thousands of custom processing units that are optimized for operations such as term weighting, proximity constraints, and exact and fuzzy matching (Mettler, 1993).[⋆] Each processing unit is programmed for a single task, and separate pipelines of processing units are formed for each profile. Simultaneous searching with multiple profiles is supported, so multilingual profiles can be implemented as a set of monolingual profiles, one for each language. Automated tools are provided to assist users with profile translation, so the effort expended to construct a profile in the first language can be leveraged to quickly produce profiles which will recognize the same concepts in other languages. But like SIFT, no provisions are made to automatically update Fast Data Finder profiles in response to the user's behavior.

[⋆] Additional information on the Fast Data Finder is available from Paracel Inc., 80 South Lake Avenue, Suite 650, Pasadena, CA 91101-2616.

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 (Stevens, 1992b). Like the electronic mail version of SIFT, InfoScope was also designed to filter Internet News using exact-match rules. InfoScope, however, implemented adaptive filtering, suggesting rules based on observations of user behavior and offering 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.

While SIFT treated 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 ways that were 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 to accommodate the limited personal computer processing power available in 1991. In addition, Stevens' goal of exploring the potential for synergy between user and machine for profile management led him to choose a rule-based exact match text detection technique. Since users are sometimes 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, the availability of a large volume of messages, and the 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 an 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 (Yan and Garcia-Molina, 1995). Even though one user could create several profiles, this level of user acceptance provides some evidence for the utility of even the simple manual profile construction 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 may vary 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 routing evaluation has provided an unprecedented venue for that type of performance evaluation. Conducted annually since 1992, the most recent routing evaluation (at TREC-5) attracted participation from 14 research groups (Harman, 1997).

NIST provides each participant with fifty topics and a large set (typically hundreds) of training documents and relevance assessments for each topic. 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 judgments 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 judgment 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 judgments do provide a principled basis for comparing the relative performance of different content-based text filtering techniques.

The TREC "routing" 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 detected (and thus hopefully relevant) 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 detection effectiveness. Neither precision nor recall calculations incorporate a factor that depends on the total size of the collection, so fallout (the fraction of the non-relevant documents that are classified by the system as potentially relevant) is used to measure rejection effectiveness. Table III illustrates these relationships.

Table III. Measures of text detection effectiveness.

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

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

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

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

In TREC, almost all of the text 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 (Salton and McGill, 1983). Average precision is computed by choosing successively larger sets of documents from the top of the ranked list that result in increasingly greater recall. Precision is then computed for each set, an interpolation technique is used to estimate the precision between the observed recall points, and the area under the interpolated curve 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 in this manner is reported as the average precision for the system on that test collection. Since both precision and recall vary between zero and one, larger values of average precision are better and the ideal value would be one.

Precision is relatively inexpensive to evaluate near the top of a ranked list because only a relatively small number of documents must be "scored" as relevant or not relevant. But it would be impossible to exhaustively evaluate recall and fallout because every document in the collection would have to be scored for relevance because the size of the document collection is not fixed. The obvious solution is to estimate recall and fallout by scoring a sample of the document collection. The sampling approach chosen for TREC, known as a "pooled relevance" assessment methodology, is to evaluate only the documents from a fixed size set that are chosen

by at least participating system. For purposes of evaluation, documents which are not selected by any system are treated as if it were known that they are not relevant. Since documents are chosen using a wide variety of techniques, it is felt that the pooled relevance assessment methodology produces a fairly tight upper bound on recall and an extremely tight lower bound on fallout. And because the collections used are fairly large (typically around 500,000 pages of text), the TREC routing evaluations provide a useful degree of insight into the performance of participating text filtering techniques in high-volume applications.

Although TREC investigates only the performance of the detection 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 detection techniques. In the TREC-5 routing evaluation, for example, 23 text filtering systems were evaluated and average precision was observed to vary between 0.25 and 0.03 (Harman, 1997).

3.2. SOCIAL FILTERING

The Tapestry text filtering system, developed by Nichols, *et al.* at the Xerox Palo Alto Research Center (PARC), was the first to include social filtering (Goldberg et al., 1992; Terry, 1993). 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 judgments ("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 as potentially relevant. 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 applied a set of simple rules, similar to those used by SIFT, to determine whether each document might be of interest to each user. The more sophisticated rules in each profile were 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. From this experience and others (c.f., (Brewer and Johnson, 1994; Hill et al., 1994; Sheth, 1994)) it appears that social filtering systems must assemble a fairly large critical mass of users before it would be possible to demonstrate their effectiveness.

The ongoing GroupLens project of Konstan, *et al.* at the University of Minnesota is presently the most ambitious attempt to reach such a critical mass using an information filtering system that is designed to manage a dynamic information source (Konstan et al., 1997).

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 rating servers (which have been developed for the project). The design permits both the content and rating 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.*

GroupLens annotations are explicit judgments 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 rating server using whatever semantics for the five values they wish. The rating 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 was conducted in 1996 using a limited number of newsgroups and a single rating server, and more comprehensive evaluations are planned.

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 (Balabanović and Shoham, 1997). 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, illustrated 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 (Brewer and Johnson, 1994). As in other experimental content-based text filtering systems, users could provide explicit binary judgments about the utility of the document. Those judgments were then used as a basis for a typical content-based ranked output system. But what made URN unique was that users could also collaboratively improve the system's initial representation of the document by adding or deleting words which they felt

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

represented (or, for deletions, misrepresented) the content of the document. In URN those changes were propagated to all other users, allowing the user community to collaboratively define the structure of the information space. Since user-specified words were given preference by URN when developing representations for new documents, users had an incentive to improve the set of words which described existing documents.

In URN each user maintained a separate content-based user model, while the shared annotation server effectively maintained 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.

4. Constructing the User Model

"User modeling" is a broad discipline that is generally concerned with how information about users can be acquired by automated systems and with how that information can be used to improve the system performance.* Many types of user models are possible; for information filtering what Rich has called "individual user, long-term user models" are needed (Rich, 1979). In this section we identify the genesis of the techniques that have been synthesized to produce effective and efficient text filtering systems. These techniques are drawn from the fields of information retrieval, recommender systems, and machine learning, and a number of related fields. Our presentation considers each field in turn.

4.1. Techniques from Information Retrieval

As Belkin and Croft observed, content-based text detection techniques have been extensively evaluated in the context of information retrieval (Belkin and Croft, 1992). Every approach to text detection has four basic components:

- Some technique for representing the documents
- Some technique for representing the information need
- Some way of comparing the information need representations with the document representations
- Some way of using the results of that comparison

When specialized to information filtering, 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 like those obtained through human comparison of the the documents themselves with the actual information

* As Karlgren, *et al.* have observed, it is also important to construct systems whose operation conforms with the user's mental model of the information filtering process (Karlgren et al., 1994). The user models we refer to in this paper, however, are models constructed by the system which describe some aspect of the user.

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 detection and display.

In each of the text filtering systems we describe in this paper, the detection 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 judgment of some relationships between an interest and a document, measured on n numeric scales (e.g., topical similarity or degree of constraint satisfaction).

As we saw in section 3, 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.* 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 shown 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 their 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 judgment 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 constructed so that the resulting boolean vector can be easily compared to the boolean expression specified by the profile. Exact match text filtering systems

* These "terms" 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.

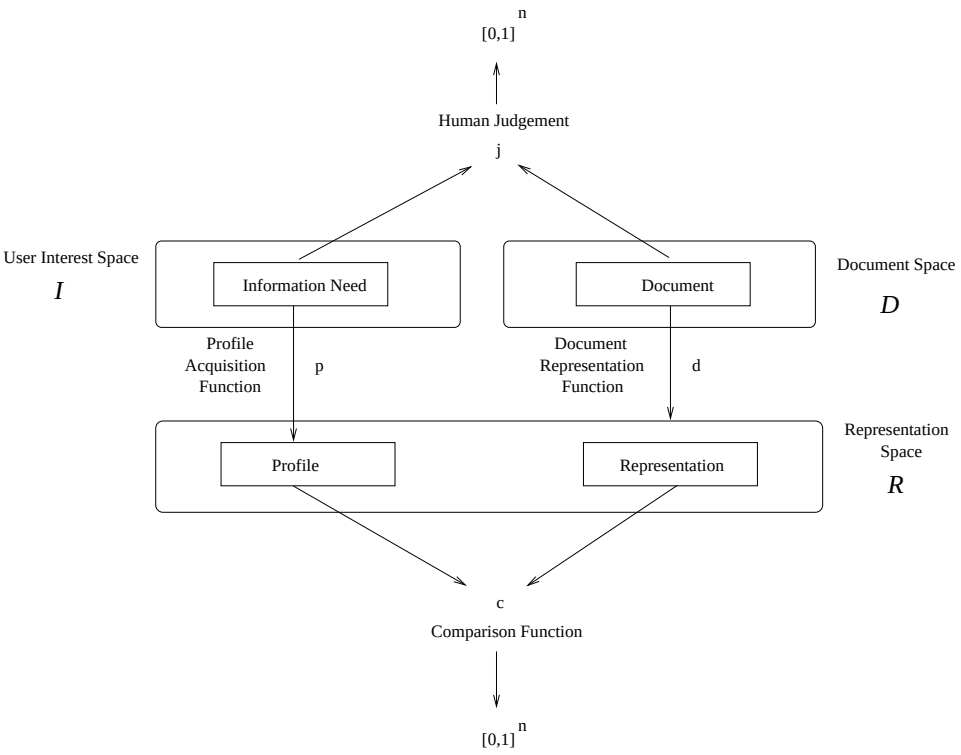

Figure 3. Text filtering system model.

typically provide an unranked set of documents which will (hopefully) satisfy the information need. This 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 the square of the cosine of that angle (easily computed as the normalized inner product of the two vectors) is often used to rank order the documents.

$$\cos^2(v_1, v_2) = \frac{v_1^{\mathrm{T}} v_2}{\sqrt{v_1^{\mathrm{T}} v_1}\sqrt{v_2^{\mathrm{T}} v_2}}$$

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 (Frakes and Baeza-Yates, 1992). One common scheme, known as "term-frequency—inverse document frequency" weighting, assigns term i in document k a value computed as:

$$tfidf_{ik} = \text{occurrences of term } i \text{ in doc } k * \ln\left(\frac{\text{number of docs}}{\text{number of docs with term } i}\right)$$

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 (Allan, 1996). Early vector space systems were commonly applied to abstracts of a fairly consistent length and topic density; the widespread availability of full text documents in electronic form has motivated the development of techniques which perform equally well on documents of any length (Hearst, 1994; Singhal et al., 1996)

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 on that observation is computed. Bayesian inference networks have proven to be a useful technique for computing such conditional probabilities (Turtle and Croft, 1990). Since it is possible to construct a Bayesian inference network 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 (Turtle and Croft, 1992).

Because more sophisticated comparison functions can be designed that produce multiple-valued results, the display module can exploit 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. A profile represents what Olsen, *et al.* have called a "point of interest" and together the profile and the comparison technique in a ranked output text filtering system can be thought of as specifying a "point of view" in the document space (Olsen et al., 1993). Multiple rank orderings can be combined to produce richer displays that combine multiple points of interest, a research area often referred to as "document visualization" or "visual information retrieval interfaces."

4.2. TECHNIQUES FROM RECOMMENDER SYSTEMS

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, *et al*. at Bellcore in which users' tastes in movies were matched using techniques similar to those implemented in GroupLens (Hill et al., 1994). 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 what is essentially "collaborative data mining" is used to explore the unlabeled corner of the problem space depicted in figure 2. The term "recommender system" has recently been proposed to describe applications of such techniques to retrieval and filtering problems (Resnick and Varian, 1997).

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 the 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 techniques used effectively by Hill, *et al.*

4.3. SOURCES OF INFORMATION ABOUT THE USER

Rich defined a distinction between "explicit" user models which are "constructed explicitly by the user" and "implicit" user models which are "abstracted by the system on the basis of the user's behavior" (Rich, 1979). Both implicit and explicit user models are found in text filtering systems. SIFT, for example, used an explicit user model, while the machine learning techniques we describe in section 4.4 can be used to create what Rich called implicit user models. Sections 4.1 and 4.2 have described how information about those documents can be acquired, either directly from their contents or from annotations made by others. In order to construct an implicit user model the system must also have some way of observing the user's behavior.

In section 3 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."* Explicit feedback has the advantage of simplicity, both for system design and system implementation. 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 single-valued numeric scales may not be well suited to describing the reactions humans have to documents. For example, is a terse document which addresses every aspect of the information need but contains little explanatory text better or worse than a verbose document which is easily understood but which provides only part of the required information? These two deficiencies of explicit feedback motivate the study of implicit feedback mechanisms.

In Stevens' InfoScope system, three sources of implicit evidence about the user's interest in each message were observed: whether the message was read or ignored, whether it was saved or deleted, and whether it was replied to or not. Because the user's 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 as explicit feedback. InfoScope did, however, allow explicit feedback as well.

* 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.

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 (Morita and Shinoda, 1994). In a six week study of eight users, they found a strong positive correlation between reading time and explicit feedback provided by those users on a four-level scale. Furthermore, they discovered that interpreting as "interesting" articles which the reader spent more than 20 seconds reading actually 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 can be associated with individual documents that describe the user's reaction to those documents.

4.4. Techniques from Machine Learning

Machine learning, the study of algorithms that improve their performance with experience, offers a rich source of techniques that are designed to exploit multiple training instances to improve detection effectiveness (Langley, 1996). By simply adjoining the features that represent a document (e.g., term frequency values) with the features that represent the user's reaction to that document (e.g., explicit feedback), the system can form a complete feature vector for each previously seen document. For new documents, only the first features (the ones 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.* 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 is possible to explain any set of observations after the fact. Langley identifies three ways in which the necessary "induction bias" can be introduced in a machine learning system: in the representation, in the search technique, and as explicit domain knowledge (Langley, 1996). 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,

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

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 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 identify a document as potentially relevant) 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 status value 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 preprocessing step that combines several observed parameters to produce an estimate of utility can be manually constructed and then the resulting estimates 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 detection or rejection of that article. These rules (e.g., detect 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 detection of Internet News articles (Foltz, 1990). Representations of about 100 articles from a training collection which the user designated as interesting were retained, and then new articles were ranked 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, Foltz found that this technique produced an average precision 43% above that achieved by random selection, and that a further 11% improvement 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," (Langley, 1996) and it is not uncommon to have thousands of terms in the vocabulary of a text filtering system. Schütze, *et al.* 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 applied by Foltz (Schütze et al., 1995). For 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, *et al.* found that using only the LSI feature vectors provided the best filtering effectiveness when applied with linear discriminant analysis and with logistic regression, and that their implementation of linear discriminant analysis was the better of those 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. Jennings, *et al.* provide another perspective on the application of neural networks to text filtering, demonstrating how an exact match detection technique can be incorporated (Jennings and Higuchi, 1993).

Exploring another machine learning technique, Sheth implemented a genetic algorithm to filter Internet News in a system called "Newt" (Sheth, 1994). 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. 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. Because the technique is both economical and reproducible, evaluation using simulated users can be useful when answers to specific questions about learning performance are sought.

4.5. Relationship to Other Fields

This completes our description of three significant sources of technology for text filtering systems: information retrieval, data mining 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.

4.5.1. *Networked Computing Infrastructure*

The physical context for the information filtering process is the existing networked computing infrastructure. The relevant portion of this infrastructure may consist of, for example, isolated workstations monitoring a common newsfeed, a workgroup computing environment supported by an local area network, or the entire Internet. With a few notable exceptions (SIFT and Tapestry), we have placed more emphasis on effectiveness than efficiency when describing design features and performance evaluations. This should not be 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. But 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, the goal of distributed computation is to optimize the tradeoff between distributing the workload and minimizing communication requirements. Yan studied this issue rigorously in conjunction his with work on SIFT, developing optimal assignments of computational tasks among a group of cooperating servers (Yan and Garcia-Molina, 1994). The GroupLens project has chosen an alternative approach that exploits an existing infrastructure for document distribution. By augmenting this infrastructure with distributed rating servers, GroupLens seeks to achieve acceptable efficiency in a manner compatible with the existing physical and social structure for Internet

News. One of the key issues to be addressed as the number of users scales up is which constraints of the existing infrastructure to accept and which will be worth the additional implementation effort to change.

4.5.2. *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 social 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 sender 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 that are developed by consensus or through competitive market mechanisms often address 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 will likely prove essential to the success of such endeavors.

Once such social conventions are created to add the necessary structure to the documents, the text filtering techniques we have described 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 (Resnick and Miller, 1996). The design of a system for creating, distributing, and using these ratings can productively be studied from the perspective of CSCW because a common task must motivate multiple participants. But because the resulting ratings are simply another type of annotation, an understanding of how annotations are used in text filtering systems would provide system designers with useful insight into how such annotations could be integrated with other sources of information about user preferences and the contents of a document to construct a comprehensive system that effectively addresses user needs.

4.5.3. *Market Formation*

For applications in which the participants lack the shared objective that is central to the CSCW perspective, economic theory can provide a useful alternative. In a market economy, "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 can 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 can be based (Avery and Zeckhauser, 1997). Common standards for the exchange of price information and monetary instruments are needed because all participants in a market benefit from such social structures, and a CSCW perspective can certainly be helpful when developing such standards. But when participants do not share common goals with respect to the use they will make of the information they seek, market dynamics provide a useful way of allocating scarce information resources such as intellectual property and expert annotations.

The vast majority of reported 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 recognize differing prices for different aspects of access to intellectual property (e.g., selective purchase of limited redistribution rights) (Denton, 1995). Commercial text filtering systems typically require explicit profiles, however, and we are not aware of any experimental results for implicit user models for text filtering which exploit price information. Like the ratings described above, prices are a type of annotation, and hence they could in principle 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 prices 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 serves to restrict information access to users for whom the value of the information justifies the cost of obtaining it (Wresch, 1996). Such unrestrained market operation is rarely allowed to persist, however, once undesirable consequences emerge. 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 unconstrained 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.

4.5.4. *Privacy*

Privacy can become an issue whenever 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 that was registered at the server by simply noting which documents were forwarded to the user's computer. An unauthorized observer who could detect which documents were 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., (Chaum, 1981; Cooper and Birman, 1995)).

In the case of social 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 in order to perform social filtering. Furthermore, Hill has observed that users may want to know the identity (not merely the pseudonym) of the people who made each annotation because humans have a remarkable ability to construct sophisticated heuristics that rely on personal knowledge of the characteristics of specific individuals (Hill et al., 1994). 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. Many details of our private lives (e.g., birth, marriage and death) are a matter of public record. On the other hand, in the United States federal law prohibits the disclosure of video rental records without a court order and 46 states extend similar protection to the borrowing history of library patrons (Bielefield and Cheeseman, 1994). 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 judgment that must ultimately be resolved by those who control the information resources that are being used and by those who are making use of those resources.

5. Observations on the State of the Art

With this background we can now identify some issues that will be important for further progress in the development of text filtering systems. In this section we discuss in some detail the relationship between the content-based and social filtering approaches to text filtering, and then conclude by identifying a set of important issues for further research.

Rather than simply removing unwanted information, information filtering actually gives consumers the ability to reorganize the information space (Stevens, 1992b). For economic reasons, information spaces have traditionally been organized by producers such as book publishers and additional organization has been added by intermediaries such as libraries. Because such intermediaries typically must serve many customers with limited resources, economic factors usually limit them to providing a small number of perspectives on the information space. Information filtering is essentially a personal intermediation service. By automating the process it becomes economically feasible to personalize the resulting organization, but the risk of using only content for this purpose is that the value added by human intermediaries may be lost. Human intermediaries organize the information space using selection and annotation, precisely the information that is exchanged in social filtering systems.

But social filtering is unlikely to provide a complete solution to users' information filtering needs. Annotations require effort and have value, so the cost of obtaining those annotations will eventually come into balance with their value. Since content-based filtering offers a competitive approach to document detection, the effectiveness and efficiency of content-based selection techniques have the potential to significantly influence the price of the annotations on which social filtering is based.

Because humans and machines base their evaluations on different features, systems which incorporate both social and content-based filtering have the potential to achieve greater effectiveness than those which use either technique in isolation. Content-based and social filtering will almost certainly prove to be complementary in other 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, on the other hand, should 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 filtering systems can, in turn, help to reduce the volume of information to manageable levels. Thus, both content-based and social filtering can contribute to the other's effectiveness, potentially allowing an integrated system to achieve both reliability and serendipity. Until content-based

approaches are integrated with collaborative approaches, some information filtering applications may fail to achieve their full potential. In this light, we are encouraged by the work of Schütze, *et al.* which suggests that machine learning techniques that effectively exploit multiple sources of evidence can be found (Schütze et al., 1995).

One reason that large-scale systems which integrate content-based and social filtering techniques have not yet emerged is that social filtering itself has yet to realize its potential. Construction of insightful social filtering experiments is challenging both because it is difficult to assemble sufficiently large user populations and because suitable measures of effectiveness are not widely yet agreed upon. Recall, precision and fallout, which are of some use in evaluating content-based filtering systems, rely on normative judgments of topical relevance that suppress exactly the kind of individual variations that social filtering techniques seek to exploit. The concept of "utility" captures this dependence on individual user needs, requiring that in addition to topical relevance that the user not already have obtained the information from another source, be able to understand the information contained in the document, and be able to apply the information to the problem at hand (Soergel, 1994). But because utility is a relation between a document and a user rather than between a document and a topic, *a priori* evaluations of utility are not possible. Furthermore, the dependence on prior information can cause the utility of a specific document to depend upon the order in which documents are presented to the user.

The explicit feedback on which present social filtering experiment designs depend represents an attempt to capture some measure of utility. Implicit feedback is an alternative source for such information that has two possible advantages. By reducing the cognitive load on the user, lesser inducements may suffice to assemble a sufficient number of participants. And by using more than one type of implicit feedback measure, it may be possible to construct a richer representation of document utility than with the single numeric values that are presently solicited by systems that depend on explicit feedback. Some potentially useful sources of implicit feedback are already known. Hill, *et al.* 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" (Hill et al., 1992). 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 measures of utility (Morita and Shinoda, 1994; Stevens, 1992b). Given the potential value of implicit feedback for both experimental and operational social filtering systems, further work on this topic should be accorded a high priority.

Another approach to evaluation of social filtering techniques is to exploit simulated users in a manner similar to that used by Sheth. Just as recall, precision and fallout characterize important aspects of content-based filtering performance, learning rate and variability in learning behavior across large heterogeneous populations

can be used to characterize important aspects of social filtering performance. Large collections of simulated users are relatively easy to construct, but the interaction between measures of effectiveness and simulated user design must be explored once suitable sources of evidence about document utility are identified.

When appropriate evaluation techniques for social filtering are available it should become practical to design experiments which investigate the interaction between social and content-based filtering. This will raise important user modeling issues (how best to combine content and annotations when building the user model), new interface design issues (to what extent can users benefit from explicit control over the way content and annotations are combined and how best to provide that control) and still more challenging evaluation issues (can useful measures of effectiveness that provide insight into the interaction be designed?). It is not too early to begin collecting anecdotal evidence of the value of combining content-based and social filtering, but experience with content-based ranked retrieval techniques suggests that widespread deployment of information management systems that incorporate significant new paradigms depends at least in part on the development of widely understood and accepted techniques for performance evaluation.

Another issue that is becoming increasingly important is the ability to filter documents in more than one language. It is now quite practical to obtain access to multilingual document streams in many applications, and the availability of such sources will likely continue to increase. While it would be possible to construct separate text filtering systems for each language, users will likely prefer systems that can exploit the same profile to detect documents in any language. The Paracel Fast Data Finder uses such an approach with an explicit user model (Mettler, 1993). Systems which incorporate an implicit user model can particularly benefit from multilingual text filtering because such systems could then learn from the user's reaction to documents in one language to select documents in any language, reducing the time and complexity of the training process. We have recently obtained promising results with three techniques in an experimental evaluation of multilingual text filtering using implicit user models (Oard, 1997). Future work to integrate these techniques with social filtering would be particularly interesting because the annotations on which social filtering is based do not depend on the languages in which the documents are written.

6. Conclusions

Early information filtering systems (then known as SDI) were developed to help manage the process of disseminating scientific information. When the printed page was the dominant 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 be both a consumer and a producer of information. This drastic reduction in publishing costs has greatly increased the importance of filtering the resulting flood of information, but the resulting variability in 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.

The design of text filtering systems benefits from research in information retrieval, recommender systems, machine learning 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 paper has reviewed progress in the field with particular emphasis on construction of the user model. Other useful perspectives are offered by Jiang (1993), Mock (1996), Stevens (1992b), and Wyle (1995).

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 detection. Text retrieval research has produced a number of content-based representations that use the frequency with which terms appear in documents, and the evolving field of recommender systems is producing a complementary set of features based on shared annotations from other users. We have argued that a synergistic combination of the two approaches offers the potential for better performance than either approach can produce in isolation. When combined with implicit or explicit feedback from the user about the documents they have examined, text representations provide a basis for construction of profiles which represent user interests. Existing work on implicit feedback shows promise, and by capitalizing on that promise it may be possible to significantly improve the performance of the collaborative component of an integrated text filtering system. With such a rich basis for further development, we are confident that future text filtering systems will explore interesting issues and find application in areas that are critical to the effective use of the emerging global information infrastructure.

ACKNOWLEDGMENTS

The author would like to express his appreciation to Nicholas DeClaris, Bonnie Dorr, Virgil Gligor, Gary Marchionini, Robert Newcomb, John Riedl, Charles Silio, Stuart Stubblebine, and the anonymous reviewers for their useful comments. This work was supported in part by NSF grant IRI-9357731, DOD grant MDA9043C7217, the Medical Informatics Network project of the Pathology Department, and the Logos Corporation.

References

Allan, J.: 1996, 'Incremental Relevance Feedback for Information Filtering'. In: H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson (eds.): *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. http://ciir.cs.umass.edu/info/psfiles/irpubs/james-sigir96.ps.gz.

Avery, C. and R. Zeckhauser: 1997, 'Recommender Systems for Evaluating Computer Messages'. *Communications of the ACM* **40**(3), 88–89.

Baclace, P. E.: 1992, 'Competitive Agents for Information Filtering'. *Communications of the ACM* **35**(12), 50.

Balabanović, M. and Y. Shoham: 1997, 'Content-Based, Collaborative Recommendation'. *Communications of the ACM* **40**(3), 66–72. http://robotics.stanford.edu/people/marko//papers/cacm.ps.

Belkin, N. J. and W. B. Croft: 1992, 'Information Filtering and Information Retrieval: Two Sides of the Same Coin?'. *Communications of the ACM* **35**(12), 29–38.

Bielefield, A. and L. Cheeseman: 1994, *Maintaining the Privacy of Library Records*. New York: Neal-Schuman.

Blair, D. C.: 1990, *Language and Representation in Information Retrieval*. Amsterdam: Elsevier.

Bowen, T. F., G. Gopal, G. Herman, T.Hickey, K. Lee, W. H. Mansfield, J. Raitz, and A. Weiribnrib: 1992, 'The Datacycle Architecture'. *Communications of the ACM* **35**(12), 71–80.

Brewer, R. S. and P. M. Johnson: 1994, 'Toward Collaborative Knowledge Management within Large, Dynamically Structured Information Systems'. Technical Report ICS-TR-92-22, University of Hawaii, Department of Information and Computer Sciences, Honolulu. ftp://ftp.ics.hawaii.edu/pub/tr/ics-tr-94-02.ps.Z.

Chaum, D. L.: 1981, 'Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms'. *Communications of the ACM* **24**(2), 84–88.

Cooper, D. A. and K. P. Birman: 1995, 'Preserving Privacy in a Network of Mobile Computers'. In: *Proceedings of the 1995 IEEE Symposium on Security and Privacy*. pp. 26–38. http://cs-tr.cs.cornell.edu.

Denning, P. J.: 1982, 'Electronic Junk'. *Communications of the ACM* **25**(3), 163–165.

Denton, B.: 1995, 'Ten Ways to Control DIALOG Alert Costs'. *Online* **19**(2), 47–48.

Foltz, P. W.: 1990, 'Using Latent Semantic Indexing for Information Filtering'. In: F. H. Lochovsky and R. B. Allen (eds.): *Conference on Office Information Systems*. pp. 40–47. http://www-psych.nmsu.edu/∼pfoltz/cois/filtering-cois.html.

Foltz, P. W. and S. T. Dumais: 1992, 'Personalized Information Delivery: An Analysis of Information Filtering Methods'. *Communications of the ACM* **35**(12), 51–60. http://www-psych.nmsu.edu/∼pfoltz/cacm/cacm.html.

Frakes, W. B. and R. Baeza-Yates (eds.): 1992, *Information Retrieval: Data Structures and Algorithms*. Englewood Cliffs, NJ: Prentice Hall.

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

Harman, D.: 1992, 'The DARPA TIPSTER Project'. *ACM SIGIR Forum* **26**(2), 26–28.

Harman, D.: 1993, 'Overview of the First TREC Conference'. In: R. Korfhage, E. Rasmussen, and P. Willett (eds.): *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 36–47.

Harman, D. K. (ed.): 1997, 'The Fifth Text REtrieval Conference (TREC-5)'. Gaithersburg, MD: National Institutes of Standards and Technology, Department of Commerce. To appear. http://www-nlpir.nist.gov/TREC.

Hearst, M. A.: 1994, 'Content and Structure in Automated Full-Text Information Access'. Ph.D. thesis, University of California, Berkeley. http://www.parc.xerox.com/istl/members/hearst/publications.shtml.

Hill, W., M. Rosenstein, and L. Stead: 1994, 'Community and History-of-Use Navigation'. In: *Electronic Proceedings of the Second World Wide Web Conference '94*. Not available in print. http://community.bellcore.com/navigation/home-page.html.

Hill, W. C., J. D. Hollan, D. Wroblewski, and T. McCandless: 1992, 'Read Wear and Edit Wear'. In: *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI '92*. pp. 3–9.

Hirschman, L.: 1991, 'Comparing MUCK-II and MUC-3: Assessing the Difficulty of Different Tasks'. In: *Proceedings, Third Message Understanding Conference (MUC-3)*. pp. 25–30.

Housman, E. M.: 1969, '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.

Jacobs, P. S. and L. F. Rau: 1990, 'SCISOR: Extracting Information from On-line News'. *Communications of the ACM* **33**(11), 88–97.

Jennings, A. and H. Higuchi: 1993, 'A User Model Neural Network for a Personal News Service'. *User Modeling and User-Adapted Interaction* **3**(1), 1–25.

Jiang, Z.: 1993, 'Understanding Information Filtering and Providing and Information Filtering System Model'. Master's thesis, University of Missouri, Kansas City.

Karlgren, J., K. Hook, A. Lantz, J. Palme, and D. Pargman: 1994, 'The Glass Box User Model for Filtering'. Technical Report T94:09, Swedish Institute of Computer Science. http://www.dsv.su.se/~fk/if_Doc/JPfilter-filer/Glassbox1.1.ps.Z.

Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl: 1997, 'GroupLens: Applying Collaborative Filtering to Usenet News'. *Communications of the ACM* **40**(3), 77–87.

Langley, P.: 1996, *Elements of Machine Learning*. San Francisco: Morgan Kaufmann.

Lehnert, W. and B. Sundheim: 1991, 'A Performance Evaluation of Text Analysis Technologies'. *AI Magazine* **12**(3), 81–94.

Loeb, S.: 1992, 'Architecting Personalized Delivery of Multimedia Information'. *Communications of the ACM* **35**(12), 39–48.

Luhn, H. P.: 1958, 'A Business Intelligence System'. *IBM Journal of Research and Development* **2**(4), 314–319.

Malone, T. W., K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen: 1987, 'Intelligent Information Sharing Systems'. *Communications of the ACM* **30**(5), 390–402.

Marchionini, G.: 1995, *Information Seeking in Electronic Environments*. Cambridge: Cambridge University Press.

Marchionini, G.: 1996, 'Browsing: Not Lazy Searching'. In: S. Hardin (ed.): *Proceedings of the 59th Annual Meeting of the American Society for Information Science*. p. 267.

Mettler, M.: 1993, 'TRW Japanese Fast Data Finder'. In: *TIPSTER Text Program Phase I: Proceedings of a Workshop held at Fredricksburg, Virginia*. pp. 113–116.

Mock, K. J.: 1996, 'Intelligent Information Filtering via Hybrid Techniques: Hill Climbing, Case-Based Reasoning, Index Patterns, and Genetic Algorithms'. Ph.D. thesis, University of California Davis. http://phobos.cs.ucdavis.edu:8001/~mock/infos/infos.html.

Morita, M. and Y. Shinoda: 1994, 'Information Filtering Based on User Behavior Analysis and Best Match Text Retrieval'. In: W. B. Croft and C. van Rijsbergen (eds.): *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*. pp. 272–281. http://shinoda-www.jaist.ac.jp:8000/papers/1994/sigir-94.ps.

Oard, D. W.: 1997, 'Adaptive Filtering of Multilingual Document Streams'. In: *Fifth RIAO Conference on Computer Assisted Information Searching on the Internet*. To appear. http://www.clis.umd.edu/dlrg/filter/papers/riao.ps.

Olsen, K. A., R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams: 1993, 'Visualization of a Document Collection: The VIBE System'. *Information Processing and Management* **29**(1), 69–81.

Pizzani, M., J. Muramatsu, and D. Billsus: 1996, 'Syskill and Webert: Identifying Interesting Web Sites'. In: M. Hearst and H. Hirsh (eds.): *AAAI Spring Symposium on Machine Learning in Information Access*. http://www.parc.xerox.com/istl/projects/mlia/papers/pazzani.ps.

Pollock, S.: 1988, 'A Rule-Based Message Filtering System'. *ACM Transactions on Office Information Systems* **6**(3), 232–254.

Ram, A.: 1992, 'Natural Language Understanding for Information Filtering Systems'. *Communications of the ACM* **35**(12), 80–81. ftp://ftp.cc.gatech.edu/ai/ram/er-92-08.ps.Z.

Resnick, P. and J. Miller: 1996, 'PICS: Internet Access Controls Without Censorship'. *Communications of the ACM* **39**(10), 87–93. http://www.w3.org/pub/WWW/PICS/.

Resnick, P. and H. R. Varian: 1997, 'Recommender Systems'. *Communications of the ACM* **40**(3), 56–58.
Rich, E. A.: 1979, 'User Modeling via Stereotypes'. *Cognitive Science* **3**, 329–354.
Salton, G. and M. J. McGill: 1983, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill.
Schütze, H., D. A. Hull, and J. O. Pedersen: 1995, 'A Comparison of Classifiers and Document Representations for the Routing Problem'. In: E. A. Fox, P. Ingwersen, and R. Fidel (eds.): *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 229–237.
Sheth, B.: 1994, 'A Learning Approach to Personalized Information Filtering'. Master's thesis, MIT, Media Lab. http://agents.www.media.mit.edu/groups/agents/papers/newt-thesis/main.html.
Singhal, A., C. Buckley, and M. Mitra: 1996, 'Pivoted Document Length Normalization'. In: H.-P. Frei, D. Harman, P. Schaüble, and R. Wilkinson (eds.): *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. pp. 21–29. http://cs-tr.cs.cornell.edu/.
Soergel, D.: 1994, 'Indexing and Retrieval Performance: The Logical Evidence'. *Journal of the American Society for Information Science* **45**(8), 589–599.
Stadnyk, I. and R. Kass: 1992, 'Modeling Users' Interests in Information Filters'. *Communications of the ACM* **35**(12), 49–50.
Stevens, C.: 1992a, 'Automating the Creation of Information Filters'. *Communications of the ACM* **35**(12), 48. http://www.holodeck.com/curt/mypapers.html.
Stevens, C.: 1992b, 'Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces'. Ph.D. thesis, University of Colorado, Department of Computer Science, Boulder. http://www.holodeck.com/curt/mypapers.html.
Taylor, R. S.: 1962, 'The Process of Asking Questions'. *American Documentation* **13**(4), 391–396.
Terry, D. B.: 1993, 'A Tour Through Tapestry'. In: *Proceedings of the ACM Conference on Organizational Computing Systems (COOCS)*. pp. 21–30.
Turtle, H. and W. B. Croft: 1990, 'Inference Networks for Document Retrieval'. In: J.-L. Vidick (ed.): *Proceedings of the 13th International Conference on Research and Development in Information Retrieval*. pp. 1–24.
Turtle, H. R. and W. B. Croft: 1992, 'A Comparison of Text Retrieval Models'. *The Computer Journal* **35**(3), 279–290.
Winiwarter, W., M. Höfferer, and B. Knaus: 1997, 'CIFS — A Cognitive Information Filtering System with Evolutionary Adaptation'. *User Modeling and User-Adapted Interaction*. This issue.
Wresch, W.: 1996, *Disconnected: Haves and Have-nots in the Information Age*. New Brunswick, NJ: Rutgers University Press.
Wyle, M. and H. Frei: 1989, 'Retrieving Highly Dynamic, Widely Distributed Information'. In: N. J. Belkin and C. van Rijsbergen (eds.): *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*. pp. 108–115.
Wyle, M. F.: 1995, 'Effective Dissemination of WAN Information'. Ph.D. thesis, LaSalle University, Mandeville, LA. http://vhdl.org/∼wyle/diss/diss.html.
Yan, T. W. and H. Garcia-Molina: 1994, 'Distributed Selective Dissemination of Information'. In: *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*. pp. 89–98. ftp://db.stanford.edu/pub/yan/1994/dsdi.ps.
Yan, T. W. and H. Garcia-Molina: 1995, 'SIFT — A Tool for Wide-Area Information Dissemination'. In: *Proceedings of the 1995 USENIX Technical Conference*. pp. 177–186. 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 these and other information filtering resources can be found on the World

Wide Web at http://www.clis.umd.edu/dlrg/filter/. The author would appreciate being notified of additional online resources or changed URL's by electronic mail.

Contents

List of Figures

List of Tables