



MessageWorld: A New Approach to Facilitating Asynchronous Group Communication

Daniel E. Rose

Jeremy J. Bornstein

Kevin Tiene

Advanced Technology Group
Apple Computer, Inc.
One Infinite Loop MS 301-4A
Cupertino, CA 95014 USA
{rose,jeremy,tiene}@apple.com

Abstract

We describe a prototype system called MessageWorld designed to offer a new approach to online communication. MessageWorld provides an alternative to traditional group information sharing tools such as e-mail, online bulletin boards, and searchable storage archives. MessageWorld relieves message authors of the burden of addressing, while prospective readers are relieved of the burden of searching; the system provides a rendezvous mechanism for uniting readers with messages of interest to them. We rely on a variety of techniques including traditional content analysis and the correlation of user preferences. Adaptive algorithms serve to incorporate some of the knowledge of the user community in the system's assessment of each user's interests.

1 Introduction

It is becoming a truism that many people will have access to far more information¹ than they can reasonably hope to process. This "information overload" assumption has inspired an increasing body of tools (both research and commercial products) devoted to helping us manage various information sources. These range from "kill files" in Usenet news readers to filters in e-mail clients², and are often recent additions to preexisting systems. In particular, information filtering has become an active research area of its own, relying on many (but not all) of the lessons learned from information retrieval [2]. (Some of this filtering research will be discussed in Section 5.) Regardless of the techniques used, the service that these tools provide to the user, broadly characterized, is to determine which items are irrelevant (and should therefore be discarded) and which are relevant (and should therefore be kept). In other words, they attempt to separate the wheat from the chaff.

¹We use the term "information" to refer to documents (primarily text) manually created by human authors and intended for use by other human users

²The term "client" refers to the client software in a client-server communication protocol, and/or the user's application software in a multiuser system.

Permission to make digital/hard copies of all or part of this material for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.

CIKM '95, Baltimore MD USA

© 1995 ACM 0-89791-812-6/95/11..\$3.50

We propose a more radical extension of the overload assumption which leads us to believe that these existing tools will be insufficient: *Many people will have access to far more RELEVANT information than they can reasonably hope to process.* If this hypothesis proves to be true, then it is not enough to separate the relevant items from the nonrelevant ones; users are drowning in wheat. Instead, our task is to figure out which items to present to each user at a particular time, given his or her specific resource and task constraints.

To address this, we have designed a system called MessageWorld, an example of a new class of rendezvous tools that attempt to facilitate meaningful group communication by providing a mechanism to unite users with the information items of most value to them. We believe that a system like MessageWorld may offer a new paradigm for online communication, one that is related to (but distinct from) current e-mail, bulletin board, and online search systems.

2 Existing Communication Tools

There are three kinds of tools commonly used for facilitating asynchronous information flow within a group of users: Electronic mail (e-mail), online bulletin board systems (BBSs), and relatively static searchable storage archives. These are shown conceptually in Figure 1. As the figure suggests, the capabilities of each system partially overlap, as do the types of communication supported. Each type has its own strengths and weaknesses.

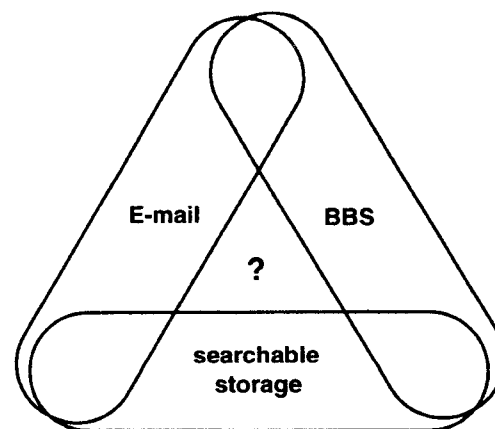


Figure 1: A conceptual space of existing tools.

E-mail. In an e-mail environment, users typically receive documents in their personal electronic mailboxes without taking any explicit action. Accessing items that have arrived then requires little effort — as little as typing the word “mail” or clicking on a mailbox icon with a mouse. E-mail systems may be based on a local-area network server (e.g. QuickMail) or on a distributed wide-area network message protocol (SMTP on the Internet). In either case there is almost always a server on which the user mailboxes reside, at least until the mail is delivered. Once users receive messages they typically can maintain them indefinitely in the mail system (space permitting), or save them to files in their personal directories.

Among the benefits of e-mail are its personal (as opposed to public), point-to-point nature, and its immediacy. E-mail is well-suited for many types of communication between pairs of individuals. One reason for this is that the messages have the presumption of privacy, and additional recipients must be added explicitly. Another is that the sender can reasonably assume that the message has been delivered. E-mail programs traditionally place little burden on the user in attending to the e-mail task; many systems allow users to set up mail receiving once and require no further action (besides opening mail when it arrives). Some environments eliminate even this installation step by providing at least a default e-mail tool with the operating system.

However, these attributes also lead to some of the limitations of e-mail. Part of being personal means that each recipient (in the form of his or her mailbox on the server) gets an entire copy of the data being sent. This is extremely inefficient when the list of recipients is very large. The point-to-point addressing scheme means that the burden is on the sender to not only be aware of anyone who should receive the message, but also to know those users’ e-mail addresses. Various mechanisms have been devised to simplify this problem by the creation of group addresses (mail aliases, *LIST-SERV* lists, etc.) However, these solutions introduce problems of their own (not least of which is people sending the message “please add/remove me to/from the list” to everyone on it). Finally, the immediacy and lack of required intervention becomes a problem when users are constantly interrupted.

As an example of these problems, consider a situation in which a group of a few dozen people work in the same organization and try to use e-mail for group communication. Group addresses are set up so that one can easily get a message to the whole group. This results in the creation of an entire class of messages that might or might not be of interest to others. For example, announcements about the company softball games may be useful to some, but viewed as “junk mail” by many other users. Meanwhile, other mass mailings are sent to lists of users who the sender suspects may be interested (“So and so is giving a talk next week...”). Unfortunately, this often results in critical people missing the event because the sender didn’t know of their interest.

To address some of these issues, many users are turning to mail filtering tools. These tools may support operations as simple as “delete messages from this user” and as complex as “put messages whose content matches this topic into that mailbox.” Techniques ranging from simple pattern matching to full-text indexing may be used. However, filtering cannot overcome the fundamental characteristic of e-mail that users only receive messages which explicitly list them (or groups they belong to) as addressees.

BBSs. In a BBS environment, messages are “posted” by

other users to directories corresponding to specific topics, and users must actively select and open these directories (often called “groups”) to view the messages in them. Usenet news, eWorld discussion groups, and WELL conferences are all BBSs by this definition. Articles posted to BBSs typically expire after some time period, at which time they are deleted. In many BBS clients, read messages are by default no longer visible to users.

In contrast to e-mail systems, BBSs are better suited toward efficiently communicating public information with a large group of users. The sender’s burden is greatly reduced, since nearly all possible “addressees” (discussion groups) are known at send time, and there are far fewer of them than there are users. Furthermore, users benefit from seeing the exchange of information between other users.

However, there are two major limitations. First, the organization of groups is typically performed by someone other than the sender or receiver (at best, they may have a vote in whether new groups are created); the burden is still on the sender to decide what audience is appropriate for the message. A common result of this problem is cross-posting, whereby the sender sends the message to every group that might possibly apply. Second, and more importantly, the receiver must seek the information in the same place (discussion group) that the sender chose to post it. If we consider the choice of discussion group to be analogous to choosing keywords from a previously defined set, the vocabulary problem [5] makes it unsurprising that potential receivers of information often fail to look in the “right” (that is, the sender’s) place.

A further problem arises from two common properties of BBS clients: the expiration of messages, and the lack of notification of new messages. Thus recipients must not only find the same place that senders chose, they must be at that place at the right time, and remember to keep checking if the message isn’t there yet. (Fortunately, newer BBS clients have begun to address the notification problem.)

Searchable storage. The advent of inexpensive tools for information sharing and search has introduced an additional method of group communication. A distributed database product such as Lotus Notes allows multiple users to share and access a collective repository of documents; file sharing combined with server-based search tools such as AppleSearch provide similar functionality; WAIS extends this type of interaction to the Internet. In each of these environments, a group of users may put documents of common interest on a file server, each using information retrieval tools to locate files matching a specific topical query. Users may similarly search their own personal databases or (if file sharing is available) those of other users. Gopher and Worldwide Web servers on the Internet are similar, but we do not include them here because most users do not yet have the ability to publish their own personal information, nor is full-text indexing an integral part of these tools.

Searchable storage systems shift the burden almost entirely onto the consumer of the information. Producers need only remember to put their information on the server or make it group-readable; they don’t need to make any decisions about where it should be posted or to whom it should be sent. Consumers, in contrast, need to know that what they are interested in exists at all, and find it if it does. Unlike a BBS (where users observe the questions of others and the discussions of the whole community), a file server or other storage system doesn’t make clear to users whether their questions and opinions are common or rare, insightful or dense. And unlike e-mail, searchable storage doesn’t tell

you when you should read it.³

There is an additional limitation that is common to all three of the systems listed above (e-mail, news, and searchable storage): Despite the fact that each system is designed to facilitate group communication, none of them takes advantage of the knowledge of its community of users. It may be that a posting to a mailing list has been read by hundreds of people, each of whom concludes that it is useless. Yet the next reader is not spared receipt of the message, nor even warned that it may not be worth reading. Conversely, a posting in a BBS may be found critically relevant to its community of readers. Yet the system has no mechanism to bring the posting to the attention of other readers, nor even to keep it from expiring before they've read it.

3 An Alternative Approach

Each of the tools described in the previous section is very good for some tasks, but not for others. In particular, the problems described in the previous section provide evidence that certain group communication tasks — especially the flow of announcements, ideas, and opinions not intended for any particular user — are not well-supported by existing tools. These tasks fall “between the cracks” of existing tools. What is needed is a tool designed specifically for this particular type of interaction.

3.1 Goals

What might an ideal system that fills this niche be like? We believe it would need to satisfy three primary goals. First, it would show all the messages of interest, and only those. Second, it would eliminate the burden of addressing and searching, so that relevant messages somehow found their way to interested users. Third, it would take advantage of the knowledge of the user community. We now consider how these goals might be achieved.

Getting all and only desired messages To avoid missing items of interest, every message must potentially be available to every user. Since there will be more relevant items than the user can handle at a given time (given our version of the “information overload” assumption), and that what a user deems relevant changes over time, the system would, in theory, keep all items in its database indefinitely, rather than expiring.

For this same reason — that the user will never have time to read all the relevant information — the system would present the users with only a small ranked list of the most important items. If these did not satisfy a user's current need for information, the user could simply ask for the next set of items (those ranked just below the ones already seen).

Clearly, the ranking mechanism is crucial to satisfying the “only” part of the goal. What factors might determine which items were of interest to a user? Some items might be highly weighted because their topical content matched the interests of the user. Items might score well because they were recommended by other users. Some might have date-dependent relevance; for example, a call for papers would increase in relevance as the deadline approached, then drop to near zero. The relevance might depend on the user's location or task context. One can imagine many factors that an ideal system could use to help predict a user's interest.

³Unless it offers a persistent query mechanism such as AppleSearch reporters. In this system, searches on users' queries are repeated at regularly scheduled intervals, and users are given an update indicating what new information has arrived.

Removing the “addressing” burden from senders and receivers. Instead of requiring creators of messages to figure out who would be interested in reading them (as e-mail requires), our system would allow authors to add their contributions to a global “soup” of items; instead of requiring prospective readers to look in particular places for items of interest (as BBSs and storage systems require), the system would deliver items of interest directly to those users. In short, the system would provide a rendezvous mechanism, uniting readers with items of individual interest.

Taking advantage of the intelligence of the user and the community. There are several ways in which the ideal system could incorporate communal knowledge. It could observe the behavior of individual users, extracting information about which type of messages they found useful. It could record usage patterns of the entire community, and incorporate such measures as the popularity of an item in its predictions. It could attempt to correlate users' tastes, so that one could benefit from the past experience of another. Ultimately, it could combine these and other methods.

4 MessageWorld

We designed MessageWorld as a first attempt to satisfy most of the goals of the ideal group communication tool described above. It is a tool related to e-mail and BBSs, but one whose primary intended use might be described as “finding out what's going on.” Much as readers of a daily newspaper may not have a specific information need in mind, MessageWorld users are expected to be most interested in keeping abreast of general issues and events than might concern them. In order to learn users' interests automatically, the system incorporates a voting mechanism whereby each user expresses an opinion on each item read.

4.1 Implementation

The prototype MessageWorld system was implemented for use on Macintosh computers over an AppleTalk network. The system uses a client-server architecture, with the client and server software communicating via the Apple Event interapplication communication protocol. The prototype server runs on a Quadra 900 with 24MB of RAM; clients have been tested on a variety of desktop and portable Macintoshes. Both main software components were written in Smalltalk, while a separate program, written in C++, is used to index the text documents.

The prototype system was designed to support a test with a few dozen users and a few thousand messages. However, design problems (described below) limited the utility of the prototype, and it was never tested with more than 20 users and 300 messages.

Two message-ranking algorithms, described below, were implemented in the initial prototype. One, based on information retrieval indexing, relied on the content of past messages preferred by a given user to predict that user's future interests; the other found correlations between users' voting patterns. In order to explore the capabilities of each algorithm, we set up two distinct databases that were well-suited to the two different ranking schemes. One contained text messages created by users, as well as news-like items (such as press releases) imported into the system. The other contained descriptions of movies.

Your Messages from "Standard"	
Score	Title
■■■■■	50,000 Newton Messagepads Sold In Under 2 Months
■■■■■	Apple Talks About Newton Messagepad Launch
■■■■■	Improved IgNobels
■■■■■	Newton Developer Conference
■■■■■	Microsoft Slashes Price Of Multimedia Encyclopedia
■■■■■	Up Close And Personal -- A Sneak Peek At Newton
■■■■■	Cheat At Computer Games With New Utility For Mac, PCs
■■■■■	Time To Rethink The Meaning Of 'Safety'
■■■■■	Newton Demo'd, Sells Out At SIGGRAPH
■■■■■	Bob Puette on regional pay scale
■■■■■	BILL WE HARDLY KNOW YE
■■■■■	Department Of Justice Takes Over Microsoft Probe
■■■■■	Seybold - Sculley's Live Demo Of Newton
■■■■■	Paging Networks Become Messaging Networks
■■■■■	Microsoft Acquires Part Of Gates' Own Company
■■■■■	Motorola To Enter PDA Race
■■■■■	Sharp Express Pad, Differences From Newton Explained
■■■■■	Apple rumors from NewsBytes -- AT&T Merger?
■■■■■	NAFTA clears critical hurdle
■■■■■	U.S. TEACHER PAY DOESN'T MEASURE UP
■■■■■	Trade deficit widens to more than \$12 billion on record impo
■■■■■	Microsoft "Food for Windows" parody
■■■■■	Apple Seeds AOCE Software
■■■■■	Dow Jones Debuts Personal Journal On-Line Service
■■■■■	IBM Ready To Launch OS/2 Without Windows
■■■■■	Windows Positively Affects Software Industry, Says Survey
■■■■■	A Powerbook For Pope John Paul II
■■■■■	Gates and text retrieval, continued
■■■■■	Sony MiniDisks for Data
■■■■■	Apple Tries To Up Sales W/ New Products, Price Cuts
■■■■■	Novel Released As "ShareWare" On The Internet
■■■■■	E-mail to President and Vice-President
■■■■■	Online Services Update
■■■■■	Arizona Firm Intros Low-Cost PDA
■■■■■	Sociology Of Cyberspace Course Deemed A Success
■■■■■	AT&T & Sierra Online Offer Virtual Communities
■■■■■	New Apple X.400 E-Mail Server
■■■■■	Virus Warning (for HyperCard-to-Newton stack)
■■■■■	Are You Still Picking Individual Stocks?
■■■■■	Apple's Future
■■■■■	SkyTel Launches Quote Service

Figure 2: The original text message list.

4.2 Example of Use

We now illustrate how a typical user might interact with the MessageWorld system. Although this example was artificially created (that is, it is a demonstration scenario and not taken from a naturalistic use of the system), it shows actual output from the working prototype.

Using the Text Database. The first database consisted of text messages. A hypothetical user of the system might arrive at the office in the morning and check her messages. She would then be shown a list of items, ranked by the system's prediction of her interest. This is shown in Figure 2; the system's predictions are depicted by the thermometer-like indicators in the "Score" column. It is clear from the list that the user's interests, based on her past interaction with the system, include information about the Newton personal digital assistant, and news about Microsoft Corporation.

Scrolling down the list of items, the user notices a subject line that catches her interest, and opens the item by double-clicking it with the mouse. The result is shown in Figure 3. She now reads the item — an announcement about the U.S.

Your Messages from "Standard"	
Score	Title
■■■■■	50,000 Newton Messagepads Sold In Under 2 Months
■■■■■	Apple Talks About Newton Messagepad Launch
■■■■■	Improved IgNobels
■■■■■	Newton Developer Conference
■■■■■	Microsoft Slashes Price Of Multimedia Encyclopedia
■■■■■	Up Close And Personal -- A Sneak Peek At Newton
■■■■■	Cheat At Computer Games With New Utility For Mac, PCs
■■■■■	Time To Rethink The Meaning Of 'Safety'
■■■■■	Newton Demo'd, Sells Out At SIGGRAPH
■■■■■	Bob Puette on regional pay scale
■■■■■	BILL WE HARDLY KNOW YE
■■■■■	Department Of Justice Takes Over Microsoft Probe
■■■■■	Seybold - Sculley's Live Demo Of Newton
■■■■■	Paging Networks Become Messaging Networks
■■■■■	Microsoft Acquires Part Of Gates' Own Company
■■■■■	Motorola To Enter PDA Race
■■■■■	Pilot e-mail program for U.S. House of Representatives
■■■■■	Sharp Express Pad, Differences From Newton Explained
■■■■■	Apple rumors from NewsBytes -- AT&T Merger?
■■■■■	NAFTA clears critical hurdle
■■■■■	U.S. TEACHER PAY DOESN'T MEASURE UP
■■■■■	Lotus Strategy Includes Skytel Paging, Client Upgrades
■■■■■	Microsoft "Food for Windows" parody
■■■■■	Trade deficit widens to more than \$12 billion on record impo
■■■■■	Apple Seeds AOCE Software
■■■■■	Dow Jones Debuts Personal Journal On-Line Service
■■■■■	A Powerbook For Pope John Paul II
■■■■■	IBM Ready To Launch OS/2 Without Windows
■■■■■	Gates and text retrieval, continued
■■■■■	Invitation to HANDSHAKE interactive telecommunication p
■■■■■	Windows Positively Affects Software Industry, Says Survey
■■■■■	How To Buy Load Funds No-Load
■■■■■	Online Congressional Hearing 7/26
■■■■■	Sony MiniDisks for Data
■■■■■	Online Services Update
■■■■■	One-Stop Mutual Fund Buying Services
■■■■■	New Apple X.400 E-Mail Server
■■■■■	Court rules electronic records must be saved
■■■■■	Apple Tries To Up Sales W/ New Products, Price Cuts
■■■■■	Novel Released As "ShareWare" On The Internet
■■■■■	AT&T & Sierra Online Offer Virtual Communities

Figure 4: The next list displayed after voting.

President getting an e-mail address — which she finds interesting. Instead of a conventional "close box" for closing the window, MessageWorld uses a pair of "vote buttons" (labeled with "thumbs up" and "thumbs down" pictures, for positive and negative opinions). These serve the dual purpose of closing the window while obtaining the user's opinion about the item; this prevents voting from becoming an additional burden unrelated to the act of reading a message. In this case, the user presses the thumbs-up button, indicating that she found the message useful and would like to receive similar items in the future.

Because the user has access to a potentially unlimited number of messages, she makes no attempt to read all of them. Instead, in a given session using the system, she will typically scan the list of items with highest ranking, and sample these as time and interest permit.

The next time this user begins a session with the system, which might typically be after lunch or the next morning, she once again asks for the list of suggested messages. As always, the rankings have been updated to take into account not only new messages but also the information embodied

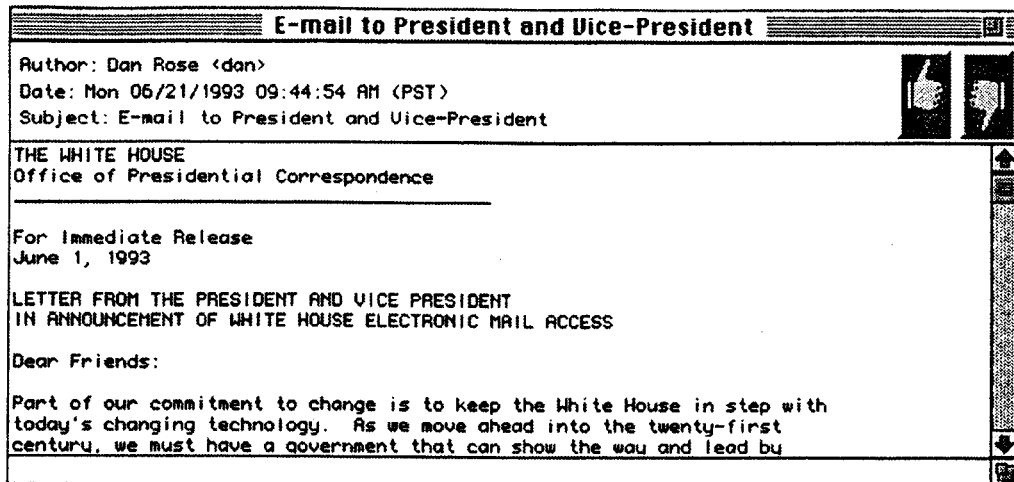


Figure 3: The message window displayed after user opens selection shown in Figure 2.

by her recent votes. This time, her list might look like the one shown in Figure 4. An item about sending e-mail to the U.S. Congress, which previously had a much lower rank (too low to be visible in the initial list) has now moved into the top 20 items. The system has incorporated the user's new interest into its model, without losing her other preferences.

In fact, the system is designed to gradually track users' changing interests. The algorithm has an "inertia" factor that determines how stable the profiles should be in light of new information. We expect this term to be tuned during user testing.

Movie Recommendations. In a second experimental database, we used a vote correlation algorithm, described below, to predict users' tastes in movies. The general usage scenario is similar to the previous one. However, we will illustrate use of the database by examining how a first-time user might encounter the system.

Our second hypothetical user selects the movie database and asks the system for its list of current recommendations. Because the user has no previous votes recorded, the system does not try to make predictions. Instead, it displays a short list of movies which have proven to be good discriminators. The user can vote on any of the items, opening them to reveal descriptions if desired, or simply pressing up and down arrow keys to vote in the list view. The resulting list is shown in Figure 5. The system can now make predictions — based on the correlation of the user's tastes with others in the community — which are shown in Figure 6.

It is important to note that the predictions are not popularity ratings of the movies. Nor do they represent any attempt to analyze attributes of the movie such as its director, stars, or genre. Instead, the predictions are based solely on the votes of others who have seen the movie and how well they correlate with the given user. Because correlations may actually be negative (if you like it, I'll dislike it), we use a two-way score indicator.

4.3 Prediction Algorithms

In order to address the goal of showing all and only the messages of interest to each user, we chose to focus on the first two criteria for prioritizing messages: content (how similar a message is to those the user has liked in the past) and correlation (whether the message was liked by people with taste similar to the given user's). We refer to these algorithms as

Your Messages from "Movie Recommendation"	
Score	Title
████████	The River
████████	Husbands and Wives
████████	Mishima
████████	Black Robe
████████	Romeo and Juliet
████████	Killer Klowns from Outer Space
████████	A Few Good Men
████████	The Right Stuff
████████	Ordeal by Innocence
████████	Rear Window
████████	Black Rain
████████	The Silence of the Lambs
████████	Melvin and Howard
████████	D.O.A.
████████	Jurassic Park
████████	What about Bob?
████████	Quick Change
████████	Duel
████████	Hannah and Her Sisters
████████	The Year of Living Dangerously

Figure 5: The votes of a new user on the initial movie list.

predictors, since they attempt to predict how much a user will like a given item.

Content Predictor. The content predictor is based on a statistical analysis of the text content of each message. We use the vector model of indexing [14] to represent each item; the document is represented by a set of weighted components, each corresponding to the statistical importance of a unique word in the document as compared to others in the database. Each user's interest is represented by a similar profile vector. The prediction score of each new message is the similarity (formally, the cosine) between the message vector and the user's profile vector.

Each time a user votes on an item, a fraction of the item's vector is added to or subtracted from the user's profile vector (depending on the vote).⁴ The fraction depends on an inertia term that determines how stable the profile should

⁴One complication occurs with negative votes. Since our similarity

Your Messages from "Movie Recommendation"	
Score	Title
■	The Wizard of Oz
■	Tootsie
■	Raiders of the Lost Ark
■	North by Northwest
■	One Flew Over the Cuckoo's Nest
■	Groundhog Day
■	Big
■	Repo Man
■	Sleepless in Seattle
■	The Terminator
■	The Birds
■	Do the Right Thing
■	sex, lies, and videotape
■	The Crying Game
■	A Clockwork Orange
■	2001
■	Apocalypse Now
■	The Player
■	My Cousin Vinny
■	Diva
■	Little Man Tate
■	Cinema Paradiso

Figure 6: Predictions based on the votes in Figure 5.

be in light of new votes. To prevent the computations from becoming too expensive, we prune all but the strongest N components of the profile and renormalize the vector after each vote.

Over time, the system develops a profile of what content words are commonly found in the messages preferred by each user (and how important they are). Thus this content algorithm is an attempt to learn about individual users in isolation.

Correlation predictor. The second algorithm is based on a pseudo-correlation of users' voting patterns. Conceptually, the algorithm formalizes the notion that "if person X liked this item, past experience suggests that person Y will also like it." Thus it ignores content and makes predictions entirely on the basis of the community.

To compute the pseudo-correlation, we take the inner product of each user's vote history with each other user's, placing the result in a matrix. We then normalize the matrix so that these raw values are replaced by the fraction of results predicted by comparison with that user. As a result, we can express the prediction for user U 's vote on item n as a weighted sum of the votes of other users. For example, we might have

$$U_n = 0.25W_n + 0.15X_n - 0.2Y_n + 0.4Z_n$$

where W_n , X_n , Y_n , and Z_n are the actual votes of other users on item n , and the coefficients are the correlation factors learned by the algorithm.

We chose this algorithm for its computational simplicity and ease of implementation. However, it has several shortcomings (such as its mishandling of missing data and its fairly arbitrary normalization scheme). While it served its purpose well enough to allow us to experiment with the general concept of prediction based on similarity of "taste," we intend to replace it in future versions of the system.

metric only makes sense for positive values, we truncate the result of subtracted components at zero.

4.4 Assessment

The prototype MessageWorld system was designed as an initial investigation into building rendezvous tools. There are two aspects of the system which require evaluation: the utility of a rendezvous tool in general, and the performance of the specific prediction algorithms.

In building the system and testing it informally, we were able to explore the capabilities of the architecture, and expose the strengths and weaknesses of the interface. In particular, we were able to isolate specific design choices that prevented the system from being fully effective.

The first of these design choices concerned how to encourage users to vote, and how to represent the voting process. Our primary concern was that voting be seamlessly integrated into the user's interaction with the system, and that it not require any additional effort. At the same time, we wanted to encourage the act of voting as much as possible, since it caused the system's predictions to improve. Our solution was to replace the close box of the window used to read a message by the two voting buttons, as was illustrated previously in Figure 3. However, this produced a problem we had not anticipated: Some users knew how they wanted to vote (on the basis of the subject line) before they opened the message window. Since the vote buttons were only visible once the messages were opened, and since opening a message incurred a delay while the text was downloaded from the server, browsing the system became a tedious process. To remedy this, our next version will cache the highest-ranked messages on the client (eliminating the download delay during browsing), and will allow voting directly from the message list.

A second problem occurred in dealing with previously read messages. By default, the list of items shown when a user launched the system was the list of unread messages. However, this had somewhat confusing consequences. Once the user voted on a message, the only way to examine it again was to open a separate (and nonobvious) window of read items. Since we expected the number of read items to increase indefinitely, we provided a separate query interface that let the user search for items on a given topic. However, the system then degenerates to a searchable storage archive, whose problems we described earlier.

Users ended up submitting few votes and fewer messages, in part because of difficulties like these that made the system less enjoyable to use. This lack of content further reduced the utility of the system. In fact, the message database never became large enough to support the intended usage model. Users often found that, instead of being shown the top 25 unread items of interest, there were no new items at all. It was clear from this experience that a critical mass of messages (and users) is a necessary condition for the usability of the system. Accordingly, we have since developed mechanisms for importing messages from preexisting message collections. These will be used to bootstrap the system, providing enough initial content to make users' participation worthwhile.

Our assessment of the prediction algorithms was based on well-known performance properties of relevance feedback in the vector model [14], on informal feedback from our users, and on analysis of user voting data. Both predictors were acceptable to users, but there were insufficient data for formal measurement. Use of the content predictor produced an important question for further study: how many areas of interest can be represented in a user profile before it devolves into an "average document" matcher? And while the correlation predictor often satisfied users, it did so by

predicting popular items rather than by fully exploiting the user's voting patterns.

We believe the prototype went a long way toward meeting the goals of our ideal system, despite the problems we have described. MessageWorld gave users a new interaction paradigm in which they were shown items deemed most interesting but allowed to look at all the items. It removed the necessity for addressing and (at least with respect to new items) for searching. Finally, it incorporated two methods for learning from the community.

4.5 Next Steps

We are currently conducting a user study to examine how to improve specific aspects of users' interaction with the system. These aspects include voting, portrayal of a message's rank, and presentation of read items. The results of this study will be used to redesign and reimplement the client software. Simultaneously, we are exploring more robust prediction algorithms. We will incorporate better measurement techniques which will give us a quantitative method for evaluating prediction performance in actual use.

5 Related Work

Ideas about MessageWorld have evolved from a large body of related work on information handling. Filtering of e-mail and/or news, in which users explicitly form lists of rules which specify how messages should be processed, have been around for many years and are becoming increasingly common in commercial e-mail systems or adjunct products. Among the best-known research systems of this type was Malone's Information Lens [11]. A similar concept is found in Selective Dissemination of Information services, where items in a news wire or other broadcast service are routed to different users depending on their profiles (typically a series of keywords chosen from a predetermined list of topics).

An extension of this work is found in systems that assist in the filter-creation process by learning from the user's behavior. This learning may take many forms. In the Maxims system [10], a Lisp program dynamically observed situation-action pairs in users' mail handling and began to predict (and optionally automate) future actions. The situations were based on structured attributes of the data and some hand-coded knowledge; a typical pair might be "when the sender is in our workgroup, put the message in mailbox X." One of the more innovative aspects of the system is that when one user's system fails to find a match for a given situation, it may ask another user's system for possible actions.

Another example of filter-creation can be found in the "Virtual Newsgroups" of Stevens' Infoscope system [4]. Infoscope is a Usenet news client that observes user behavior and, using a set of rules, suggests the creation of new filters. The filters are represented as virtual newsgroups. Similarly, Sheth and Maes' NewT system [16] filters Usenet news into previously nonexistent categories. But while Infoscope suggests the categories, NewT requires that users set up categories a priori, training the system by giving examples of each. On the other hand, NewT's filters incorporate ranking based on vector-model text indexing, while Infoscope's are limited to structured attributes.

For a long time there were relatively few systems in which the actions of other users affected one's interaction with the system. In the information retrieval domain, Brauen [3] pioneered the notion of users' feedback causing permanent changes to the system's representation. Belew's AIR

system [1] used a similar concept in an attempt to incorporate a "consensual" notion of relevance; SCALIR [13] attempted to extend this paradigm to a "communal repository of knowledge." Both systems induced a single representation of terms and documents from the collective behavior of all users. Hill [7] also explored how representations — in this case, how documents appear in the user interface — can emerge from communal usage patterns.

More recently, the notion of using the ratings by individuals to predict what others in the community will like has gained popularity. Karlgren [9] has attempted to predict the similarity of documents based on user ratings. Resnick's GroupLens system [12] ranks Usenet news messages based on correlations of user ratings. A system at Bellcore [8] performs a similar task for predicting what movies people would like. Shardanand and Maes' Ringo⁵ [15] system recommends music based on finding agreement in users' tastes. These last three use quite similar techniques to the correlation predictor used on MessageWorld's movie database. However, they are motivated by goals — improving existing tools (in the case of GroupLens), forming virtual communities (Bellcore), and filtering (Ringo) — that differ somewhat from MessageWorld's rendezvous task.

Another related system is Tapestry, a collaborative filtering tool [6]. Like MessageWorld, Tapestry is motivated by the problems with current mail and bulletin board tools, and by a belief in the value of the community in helping handle the information glut. Unlike MessageWorld, however, Tapestry users create specific filters from templates or a formal query language, explicitly relying on the annotations of other users to assist them.

6 Conclusion

Much research has focused on how to improve existing online communication tools in order to manage the flood of information faced by many users today. We have chosen instead to consider the situations for which existing tools seem ill-suited (namely, keeping up to date with new developments and opinions in a group), and have designed a new class of tool for these situations, using the technology of information retrieval. The MessageWorld prototype represents our first attempt to implement such a tool. The prototype uses two prediction techniques — content and correlation — to match users with messages of interest. An assessment of the general utility of the system will require a longitudinal study of a more robust version. Nevertheless, our experiences with MessageWorld suggest that the framework offers a rich avenue of exploration for improving group communications.

Acknowledgements

We would like to thank Doug Cutting, Andrea Gallagher, Sally Grisedale, John Hatton, Bob Hon, Dulce Ponceleón, and Dan Walkowski for their participation in, support for, and constructive criticism of the MessageWorld project.

References

- [1] BELEW, R. K. Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. In *Proceedings of the 12th International Conference on Research and Development in In-*

⁵Later renamed HOMR.

- formation Retrieval* (Cambridge, MA, June 1989), N. J. Belkin and C. J. van Rijsbergen, Eds., pp. 11–20.
- [2] BELKIN, N. J., AND CROFT, W. B. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM* 35, 12 (December 1992), 29–38.
- [3] BRAUEN, T. L. Document vector modification. In *The SMART Retrieval System — Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1971, ch. 24.
- [4] FISCHER, G., AND STEVENS, C. Information access in complex, poorly structured information spaces. In *Human Factors in Computing Systems (CHI-91 Conference Proceedings)* (1991), pp. 63–73.
- [5] FURNAS, G. W., LANDAUER, T. K., GOMEZ, L. M., AND DUMAIS, S. T. The vocabulary problem in human-system communication. *Communications of the ACM* 30, 11 (November 1987), 964–971.
- [6] GOLDBERG, D., NICHOLS, D., OKI, B., AND TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35, 12 (December 1992), 61–70.
- [7] HILL, W., HOLLAN, J., WROBLEWSKI, D., AND McCANDLESS, T. Edit wear and read wear. In *Human Factors in Computing Systems (CHI-93 Conference Proceedings)* (1993), pp. 3–9.
- [8] HILL, W., STEAD, L., ROSENSTEIN, M., AND FURNAS, G. Recommending and evaluating choices in a virtual community of use. In *Human Factors in Computing Systems (CHI-95 Conference Proceedings)* (1995), pp. 210–217.
- [9] KARLGREN, J. An algebra for recommendations. Tech. Rep. Working Paper #179, Systems Development and Artificial Intelligence Lab, Royal Institute of Technology (Sweden), 1994.
- [10] LASHKARI, Y., METRAL, M., AND MAES, P. Collaborative interface agents. In *Proceedings of the National Conference on Artificial Intelligence* (1994).
- [11] MALONE, T. W., GRANT, K. R., AND TURBAK, F. A. The information lens: An intelligent system for information sharing in organizations. In *Human Factors In Computing Systems: CHI '86 Conference Proceedings* (1986), pp. 1–8.
- [12] RESKNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P., AND RIEDL, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the Conference on Computer Supported Cooperative Work* (1994), pp. 175–186.
- [13] ROSE, D. E. *A Symbolic and Connectionist Approach to Legal Information Retrieval*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1994.
- [14] SALTON, G., AND MCGILL, M. J. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, 1983.
- [15] SHARDANAND, U., AND MAES, P. Social information filtering: Algorithms for automating “word of mouth”. In *Human Factors in Computing Systems (CHI-95 Conference Proceedings)* (1995), pp. 210–217.
- [16] SHETH, B., AND MAES, P. Evolving agents for personalized information filtering. In *Proceedings of CAIA-93* (1993).