# A Brief Comparison of News Filtering Software

Fredrik Kilander
fk@dsv.su.se
http://www.dsv.su.se/~fk

June 19, 1996

IntFilter
K2LAB — Laboratory for Knowledge Engineering and Communication
Department of Computer and Systems Sciences
Stockholm University and the Royal Institute of technology
Electrum 230, S-164 40 KISTA
Sweden

## 1 Introduction

A number of workers are currently providing or researching mechanisms to filter Internet messages. The most common target of these efforts are directed at Usenet News, a system for computer-mediated communication with world-wide distribution. Inspired by a compilation of references to message filtering systems provided by Doug Oard[1] *(oard@glue.umd.edu)* this paper compares a few systems using a number of informal, but no less informative, measures.

## 2 Comparative Measures

In this paper the filtering systems are discussed using the following terms:

- *Model*—The general filtering model: does it retrieve or rate.
- *Adaptation*—How the filter is adapted to the individual needs of the user.
- *Classification*—How the filter performs its classification.
- *Motivation*—How the filter shows or defends its classification of a particular article.
- *Operation*—How and when the filter operates.
- *Implementation*—How the filter is implemented.
- *Results*—What are the results (if any).

[1] http://www.glue.umd.edu/enee/medlab/filter/filter.html

# 3 SIFT

The Stanford Information Filtering Tool (SIFT) [YGM95] is a service provided over the World Wide Web (WWW). Users access the filter by submitting one or more *subscription profiles* to the service using a WWW-browser with forms cabability. Each profile specifies a number of keywords using two alternative matching strategies; a boolean conjunction of (optionally negated) terms, or, a set of keywords and a weight threshold. The profiles are matched against the entire news-feed received at Stanford. Matching articles are sent to the user by email.

## 3.1 Model

SIFT is a retrieval system. It examines articles across all newsgroups. Neither newsgroup or thread structure is preserved by this, although it appears possible to narrow a profile to a group by using strict boolean matching and the name of the group as a keyword.

## 3.2 Adaptation and Classification

The user supplies a uniquely identified profile using a WWW-browser with forms cabability. Basically the profile is a conjunction of keywords. The user can choose between two alternative matching algorithms.

### 3.2.1 Boolean matching.

The first algorithm treats a list of keywords as a boolean conjunction of keywords. All keywords must appear in the article in order to retrieve it. Keywords may be negated, in which case the presence of a negated keyword prevents the article from being retrieved.

The only way to provide a disjunction of terms is to submit multiple profiles. Each profile will generate its own report.

### 3.2.2 Weighted matching.

The second algorithm is the vector space model (Cosine measure) [SM83]. In SIFT it uses a list of keywords in combination with a threshold set by the user. Each article is given a similarity score (the article's distance from the query) in the range 0–100. Articles that score above the threshold are retrieved.

### 3.2.3 Abstract length and report interval.

Two further parameters are available to the user; the interval in days between reports and the number of lines to be abstracted from each article.

### 3.3 Operation and Implementation

SIFT operates at a remote site, out of the user's control and sends abstracts of the retrieved articles to they user by email. It is implemented in C and has been compiled on several Unix platforms (DEC Ultrix 4.2, HPUX 8.07 and SunOS 4.1).

### 3.4 Results

Results available so far appears to deal primarily with the economy and efficiency of the filtering engine itself.

## 4 NewsWeeder

NewsWeeder [Lan94, Lan95b] is a system in experimental use at Carnegie Mellon University. A special WWW-browser serves as user interface. The user visits each newsgroup of interest in turn. For each group, the user rates interesting articles using a number in the range 1–5.

The system collects the rating information and presents the overall rating averaged across all other NewsWeeder users. As such the systems provides collaborative filtering. NewsWeeder can also provide a *prediction* of the score, computed with machine learning techniques from the user's profile of past ratings.

### 4.1 Model

NewsWeeder appears to be both a conventional newsreader with the capability to rate articles, in conjunction with a retrieval system. The user may either read news as usual, or go to a *virtual* newsgroup, containing the 50 top articles selected for that particular user.

### 4.2 Adaptation

While reading, the user ranks each article as belonging to one of five categories: *essential, interesting, borderline, boring* or *gong*. The so labelled articles provide training examples for a machine learning routine which is invoked nightly to create a user profile for the next day.

### 4.3 Classification and Motivation

NewsWeeder has experimented with a number similarity measures. However, two of them, the Cosine measure and Minimum Description Length (MDL) appeared to outperform the others. Of the two, MDL gave better precision. New articles are measured for similarity with the ranking categories in the user profile. The articles are then given a predicted rating, i.e. the machine learning component attempts to predict the rating the user will assign to the article. The rating is continuous on the scale from *essential* (1.0) to *gong* (5.0), while the user is limited to discrete ratings.

NewsWeeder also has a capacity for collaborative filtering by combining article scores from several users. In this mode, each article is displayed with a rating averaged from all users of the system. There appears to be no interconnection between the rating predicted for each user and the collaborative estimate.

### 4.4 Operation and Implementation

Retraining of the user model is done each night. It is not currently clear when the actual prediction of unread articles takes place. There is no information on implementation, except that the news-reading client is (a possibly hacked) WWW-browser.

### 4.5 Results

Results are promising, but practical user studies are so far very limited. Two users (out of 40 registered) stuck with the system for a year and reached a precision of 44 % and 59 % respectively. The results appear to be conditioned on the stability of the user's interests. Work is under way to address the issue of modelling the user-interest life cycle and to allow explicit user feedback on interest changes [Lan95a].

## 5 Browse

Browse [JH92] is a reader for Usenet News with a neural network model of the user's interests. Each message presented to the user is classified as either being accepted (read) or rejected (not read). The first 300 words of the message (filtered by stop list of common words) are used to represent the features of the message.

### 5.1 Model

Browser rates articles and can be used in one of three modes: *all articles, selected ranked articles* and *augmented keyword search.* The *all* mode simply shows all articles in chronological order. The *search* mode employs the neural network model to rank each article and filter them against a threshold value (settable on the GUI). In the augmented keyword search mode the user can type in keywords that increase the acceptance or rejection of an article. In this mode the supplied keywords are used to temporarily modify the neural network model by forcing high or low the activation levels of the corresponding keywords (nodes) in the network.

By varying the threshold, the user has coarse control over precision and recall. However, the system is basically designed to emphasize the difference between accepted and rejected articles, forcing the user to classify each article as belonging to either class.

## 5.2 Adaptation

Articles shown to the user are assumed to be read and therefore also accepted, unless the user explicitly flags it as rejected. Unread articles are implicitly marked as uninteresting. There appears to be no special difference in feature extraction between read and unread articles. Such behaviour would be expected on the grounds that articles never shown to the user cannot have been rejected on their contents. The subject line or other header fields are the only possible cues in this case, disregarding the infinity of other reasons which may have prevented the user from requesting the text of the article.

## 5.3 Classification

The neural network model forms associations between pairs of words (not necessarily adjacent in the text) in such a way that it becomes sensitive to co-occurrence probabilities. It is a more powerful approach than simply comparing similarity in terms of common terms between documents since the network has the ability to reach associations that stretch over words not actually present in a particular document but still pertinent to the general concept.

## 5.4 Operation

The Browse system recognizes different newsgroups but apparently no threading. Weight is placed on individual articles and all articles in the group are filtered together. Each newsgroup has a separate article history, neural network model and stop list of common words.

The system actually consists of two subprograms, the browser proper and *snooze*, a program which updates the neural network model. It must be run manually after each reading session.

The ranking of articles takes places when the user selects *selected articles* or *augmented keyword search*. The scoring process takes several seconds for each article, forcing the user to wait for the process to complete.

## 5.5 Implementation and Results

The browser is implemented in C using the X Window System as the GUI. The authors suggest a distributed system where a compute-bound server runs both the neural network model processing for all users in addition to the regular tasks of a Usenet News (NNTPD[2]) server.

## 5.6 Results

Available results deals mostly with precision and recall of the neural network model.

[2]Network News Transfer Protocol Daemon, RFC 977.

# 6 NewsClip

NewsClip[3] is a programming language focussed on news filtering. The compiled filter operates offline by comparing the user's *.newsrc* file with the contents of the news server. It applies its rules to all unread messages and rewrites the *.newsrc* file in such a way that messages that are filtered out are marked as already read. As a result, the user is free to employ any conventional newsreader.

## 6.1 Model

NewsClip works by binary rating. Articles rejected by the filter never reach the user since they have been marked as already read by NewsClip.

## 6.2 Adaptation and Classification

The user must manually write the filtering program using the specialized NewsClip language. The complexity of the language appears to be on the level of Pascal or Perl.

The compiled filter mechanically applies the rules to the set of unseen messages.

## 6.3 Motivation

Apparently none.

## 6.4 Operation

NewsClip runs as a batch program, manually or automatically. Since it modifies the *.newsrc* file it is probably a bad idea to read news while the filter is running.

## 6.5 Implementation

The NewsClip compiler translates source code into C, which is then compiled with a regular C compiler and linked with a special subroutine library.

## 6.6 Results

None known at this time.

# 7 Lurker

Lurker is a personal retrieval program written in Perl by Vince Taluskie (*taluskie@utpapa.ph.utexas.edu*). In short, it can be seen as a way to periodically perform *grep* on the Usenet News server and have the selected articles stored on disk or sent via email. In spirit it is similar to SIFT.

[3] http://www.clarinet.com/newsclip.html

### 7.1 Model

Retrieve matching articles.

### 7.2 Adaptation and Classification

The user writes a set of rules for each newsgroup. The rules are boolean expressions consisting of keywords and (Perl) operators for conjunction, disjunction and negation.

When run, Lurker applies the rules mechanically and retrieves all selected articles.

### 7.3 Motivation

Apparently none, with the possible exception of debugging output.

### 7.4 Operation

Lurker runs as a batch program, manually or automatically.

### 7.5 Implementation

Lurker is written in Perl and requires a Perl interpreter.

### 7.6 Results

None known at this time.

## 8 SMART

SMART [Buc85] is an old and ambitious software package. It is basically an information retrieval system, with its functions being open to specialized applications. The general system can be used with the newsfeed as the message store, or its routines can be built into a special-purpose browser.

### 8.1 Model

SMART is basically a retrieval system.

### 8.2 Adaptation and Classification

The user has a specific information need and uses iterative query refinement to focus precision and recall. SMART uses the vector space model (Cosine measure [SM83]) in combination with database management routines.

## 8.3 Motivation and Operation

No special motivation is provided when the user works with iterative feedback queries. Other applications built upon the search engine may operate differently.

## 8.4 Implementation

SMART has been re-implemented under Unix, which makes it likely that C or $C^{++}$ is the language of choice. The basic task of the system is experimental although later versions are targeted at naive users.

## 8.5 Results

Results undoubtedly exists, but are most likely concerned with precision, recall and the efficiency of the retrieval system.

# 9 GroupLens

GroupLens by Resnick et al [RIS$^{+}$94] is a system for distributed, collaborative filtering of Usenet News. Unlike most other systems who regards the News systems as an opaque message store, GroupLens ties deeply into the infrastructure of News and exploits its mechanisms for its own purpose. Briefly, GroupLens allows users to rate articles as they are read. The ratings from each user is then stored in a rating server, a *Better Bit Bureau*. Just as in the News system, there can be many servers across the Internet and users subscribe to the server nearest to them. The rating servers use a distinguished newsgroup in the ordinary news system to exchange ratings.

The synergy achieved by having articles rated by many readers is that the rating servers can correlate users with similar preferences. This allows the rating server to hypothesize the rating a user will give to an unread article.

## 9.1 Model

GroupLens is a rating system with the ratings supplied by the users. Each rating is stored in a rating server as a real number in the range 1.0 – 5.0. The time the user spent reading the article is also stored but the information is not (yet) used. The ratings are shared between the servers.

## 9.2 Adaptation and Classification

As the user continues to submit rated articles, the rating server is able to correlate the user's preferences against those of other users. The basic heuristic used is that people who have agreed in the past will agree in the future. The user-profile for a particular user is simply a vector of weights, indexed by the *other* users. The vector is constructed by correlating past ratings with other users. A prediction for a particle article and user is then possible by combining the existing ratings with the weights in the user profile.

## 9.3 Motivation

The information from the rating server is the predicted rating of the article. It may be possible to query the server for the user profile (the vector of weights) since care has been taken to preserve the possibility for rating users to hide their true identity, but this information is likely of limited use.

The display of the predicted rating is at the option of the News client. A floating point number, rounding to one of the letters 'A' – 'E' and a bar graph has been tried.

## 9.4 Operation and Implementation

The GroupLens development team has modified and tried at least three different newsreaders, Gnus (emacs, unix), NN (unix) and NewsWatcher (Macintosh). The modifications included communication with the rating server, the ability to submit ratings and a presentation mechanism of the predicted ratings.

## 9.5 Results

Two studies involving seven and four subjects were performed. The input from the subjects served chiefly to evolve the user interfaces.

# 10 Borges

*[For lack of time, this and the following systems must be described very briefly.]*

The BORGES project[4] is a European effort to build a News and WWW filtering system using a WWW-browser as the client and the SMART system as the retrieval engine.

BORGES consists of a WWW-server interfacing a SMART database. The database stores (or can access) user profiles and News articles. The user submits standing queries (as with SIFT) and then retrieves the result in the form a WWW-page updated each day.

The interface for query submission allows the interactive refinement of keywords and phrases based on polysemous words. That simply means that the interface highlights words and phrases that may have multiple meanings, like the word *"bank"* which may refer to a financial institution, a subset of a computer's memory or the sloping sides of a riverbed.

## 10.1 Highlights of Borges

- BORGES is a retrieval system.

[4] http://borges.compapp.dcu.ie

- Any standard WWW-browser can serve as the client.
- Queries are submitted using keywords and phrases.
- Digests of News articles and WWW-pages are updated daily.

# 11 InfoScan

InfoScan[5] is a commercial system which in essence presents a two-dimensional view of how a number of locally stored texts relate to a particular query.

The texts (which must be files on the local file system) are arranged into what InfoScan calls *sources*. These sources are then further aggregated into *collections*. The user maintains a set of *filters* where each filter is a set of keywords. When a filter is combined with a collection the result is a query, and infoscan visualizes the relation between the filter and the texts found in the collection.

The system appears to be developed for an environment where a News server is present in the local file system.

## 11.1 Highlights of InfoScan

- InfoScan is a retrieval system.
- The GUI is graphically well developed.
- Interactive query refinement is possible.
- Keywords can be weighted and have scope (header, field, body).
- Keywords can be loosely matched, allowing for misspellings.

# 12 RAMA

RAMA [BY96] is a retrieval system where a query server accepts queries via email. It then searches through a collection of News, FTP and WWW sources it knows about and returns the result via email. The user may have standing queries. The server keeps track of users, queries and the time answers were reported to avoid serving the same answer twice. The matching algorithm uses the Cosine measure and simple pattern matching (agrep).

## 12.1 Highlights of RAMA

- RAMA is a retrieval system.
- Standing queries offers incrementally updated answers.
- Communication with RAMA is via email.

[5] http://www.machinasapiens.qc.ca/english/infoscanang.html

# 13 Pefna

PEFNA is a rating system, developed by Kilander et al in the IntFilter project[6] at the department of computer and systems sciences (DSV) at Stockholm University.

PEFNA consists of two pieces of software, a Usenet News client and a classifier agent. The user runs the client to read news from a standard news server. The client allows the user to create and maintain an ordered list of freely named categories. An article which is judged by the user to be a good example of a particular category is saved under the name of the category. Over time, the user accumulates a number of categories and one or more example articles in each.

The agent runs as a batch program, preferrably in the morning hours. It compares the user's *.newsrc* file with the contents of the news server and retrieves any new articles. Each article is then compared against the examples in each category and a list of distances is built using the Cosine measure. The list is stored in a database in the user's home directory.

When the user launches the client the next time and visits a newsgroup, the client builds a list of threads for unread articles in the group. It also examines the user's database for the ratings provided by the agent. In each thread, a category is selected as representative and used to label the thread. The list of threads presented to the user is then sorted to correspond to the sequence in which the user has arranged the list of categories.

## 13.1 Highlights of Pefna

- PEFNA is a rating system.
- Categories and examples are supplied by the user.
- Categories can be global or per newsgroup.
- The PEFNA agent categorizes articles.
- The News client categorizes threads.

# 14 Comparisons

## 14.1 Model

The *model* dimension divides the systems in two groups:

- Retrieval systems.
- Rating systems.

The distinction may seem artificial but there is a difference in view which affects system design. SIFT and SMART, two of the most ambitious systems in the survey, are retrieval systems. They regard the flow of news as a

[6] http://www.dsv.su.se/~fk/if_Doc/IntFilter.html

| Retrieval | Rating |
|---|---|
| SIFT | NewsWeeder |
| Browse | Browse |
| SMART | InfoScan |
| Lurker | GroupLens |
| NewsClip | PEFNA |
| RAMA | |
| BORGES | |

Table 1: Filter systems categorized by model.

| Programming | Query | Training | Collaborative |
|---|---|---|---|
| NewsClip | SIFT | NewsWeeder | NewsWeeder |
| Lurker | Browse | Browse | GroupLens |
| | SMART | PEFNA | |
| | RAMA | | |
| | BORGES | | |
| | InfoScan | | |

Table 2: Filter systems categorized by adaptation mode.

message store in which tidbits of interesting material can be found. The retrieval model is not too occupied with such concepts as multiple message streams and threads which reflect the dynamics of computer-mediated communication.

Rating systems are perhaps the kind of systems most closely associated with the term filtering. They tend to identify the messages that the user is likely to find uninteresting and arrange so that they do not reach the eyes of the user. They do this either by identifying interesting messages and cancelling the complement set or by ordering each message on a scale and possibly cancelling or hiding the messages that score beneath a threshold.

Table 1 shows the 11 systems categorized according to their working model. Note that Browse is present in both columns.

## 14.2 Adaptation

The filters in this survey are instructed in three ways:

- The filter is *programmed* by the user.
- The user creates a *query* which the filter attempts to answer.
- The filter is *trained* from examples given by the user.

Table 2 shows the 11 systems. NewsWeeder and Browse are basically trained systems but NewsWeeder is open for collaborative filtering and Browse allows the user to place queries.

| Boolean expression | Cosine measure | MDL | Neural network | User correlation |
|---|---|---|---|---|
| NewsClip | NewsWeeder | NewsWeeder | Browse | GroupLens |
| SIFT | SIFT | | | |
| Lurker | SMART | | | |
| | RAMA | | | |
| | InfoScan | | | |
| | BORGES | | | |
| | PEFNA | | | |

Table 3: Filter systems categorized by classification method.

## 14.3 Classification

The purpose of the classification mechanism is to assign each article to a class, or more generally to a point on a scale of attractiveness. Methods that use boolean expressions commonly reach a point where a binary decision is made; the article is either interesting enough to be recommended or it is not. Other classification schemes allow a softer gradation, allowing the user to modify the cutoff threshold at will.

Five different types of classification schemes are represented among the examined filters:

- Boolean expressions.
- The Cosine measure.
- Minimum description length (MDL).
- Neural network modelling.
- Correlation of user profiles.

Table 3 shows the 11 systems and their method of classification.

## 14.4 Motivation

Motivation appears to be very limited among the examined systems. The user must be content with the selection, and when provided, the filter's estimated rating. As a result it appears very difficult for the average user to gain access to the exact components of the filter that participated in the classification of a particular article.

## 14.5 Operation

The filters operate in three modes, interactive with the user, as a two-component system with an interactive reader and a batch training component or as an independent batch program:

- Interactive.

| Interactive | Interactive and batch training | Batch |
|---|---|---|
| SMART | NewsWeeder | Lurker |
| InfoScan | Browse | SIFT |
| | BORGES | NewsClip |
| | GroupLens | RAMA |
| | PEFNA | |

Table 4: Filter systems categorized by operation method.

- Interactive and batch training.
- Batch.

Table 4 shows the 11 systems and their method of operation.

## 15 Conclusions

The 11 systems examined in this paper presents a wide variety of approaches to the filtering task. The major differences are— in the author's opinion— three:

- The filter either views Usenet News as a message store from which interesting articles can be retrieved, or as a stream of new messages, some of which are more interesting than others.
- The filter is a complex software product, built upon established practice in information retrieval or experimental thrust in machine learning, or it is a relatively simple pattern matching engine using boolean expressions.
- The filter uses a specially adapted user interface, equipped with the proper controls, or it works more or less silently in the background. Combinations of these two techniques are also present.

Among the examined filters, few authors appear to explicitly recognize the effect of the filter on the conventional reading situation. The effect on thread structure, newsgroup structure and temporal ordering poorly documented. Instead, the consequences must be inferred from the way each filter operates.

Results concerning the end users are rarely available, most workers appear to be content to study the efficiency of their system or leave any evaluation to the end user. When results are presented they mostly deal with quantities and qualities from information retrieval; precision, recall and efficiency.

# 16 Comments

## 16.1 Collaborative Filtering

For the cynic, efficient collaborative filtering resides upon two assumptions; that others have rated the message before me and that it works better the more you conform to them.

Obviously, someone must be the first person to rate a message. Subsequent readers may add ratings of their own, but as the message become of age the incentive to rate it decreases. Rating the message for the benefit of others is meaningless for the last reader. In general, fast and early readers will have to do a lot of rating while receiving little or no support themselves. It follows that these persons either will refrain from submitting ratings or that their input will guide the selection of the majority of the subsequent readers.

The reason why collaborative filtering is feasible in spite of the cynic's view is that with a *sufficiently large group* there will be enough variation in who reads a particular message first to give everyone support over time. There will also be enough participants to enable almost everyone to recognize a subgroup matching their own taste.

A reasonable compromise between pure collaborative filtering and personal filter systems is to allow for the exchange of user profiles. That way the individual still receives filtering support from a personalized service, but is given the opportunity to import queries or profiles from other users. This removes the frailty of the first assumption since the first reader will always be the computer. It also moves the altruistic nature implicit in collaborative filtering from articles to generalizations of articles. Conformity is also less of an issue, since each profile is shaped after its user. Software can perform the matching and brokering of profiles without having the user feel guilty for not providing enough feedback to the community.

## 16.2 The Learning Filter Contradiction

There is an inherent problem present in the contradiction between accurate filtering and the usage of machine learning techniques. Machine learning needs training examples to learn from, and these examples should be both proper examples and counter-examples.

Given the vast space of possible user interests, common ML techniques requires rather large amounts of training data before they can settle on an accurate user profile. This need was noted in the work on NewsWeeder [Lan95b]. Since people's interests evolve over time (both gradually and rapidly) it can be expected that any automatically maintained user model will lag behind the actual interests of the user. The filter will still need examples of good and bad in order to keep track of the user. But, the purpose of the filter is to reduce the amount of text that reaches the user and this works against the production of good training examples.

For a user with a stable interest it can be expected that the user profile gets successively honed into something which is close to perfection. This would reduce the number of counterexamples, as only positive articles are seen by the user. But, when the interest of the user is changing, articles that previously were candidates for positive training examples will be regarded as negative. In the extreme, there will only be negative examples from the user to the filter, since the filter is already removing the set of articles in which the potential new positives are members.

To avoid ending up in a dead-end and a gravely dissatisfied user, the filter should be prepared to "open up" its filter when the amount of positive feedback is dwindling. At this point, any indication from the user regarding new positive directions should be treated with great emphasis. It may also be convenient to store old user profiles so that old interests can be recognized or quickly reestablished.

# References


[Buc85] Chris Buckley. Implementation of the smart information retrieval system. Technical report, Dpt of Computer Science, Cornell University, Ithaca, New York, May 1985. http://cs-tr.cs.cornell.edu:80/TR/CORNELLCS:TR85-686?abstract=.

[BY96] Jim Binkley and Leslie Young. Rama: An architecture for internet information filtering. ftp: // ftp.cs.pdx.edu / pub / faculty / jrb / rama / rama.k1.ps, 1996. Computer Science Department, Portland State University, Portland, OR 97207.

[JH92] Andrew Jennings and Hideyuki Higuchi. A personal news service based on a user model neural network. In *IEICE Transactions on Information and Systems*, March 1992. http://www.cse.rmit.edu.au/~rdsajj/brow.ps.

[Lan94] Ken Lang. Newsweeder: An adaptive multi-user text filter. Technical report, School of Computer Science, Carnegie Mellon University, 1994. http://anther.learning.cs.cmu.edu/res_sum.ps.

[Lan95a] Ken Lang. Personal communication, May 1995.

[Lan95b] Ken Lang. Newsweeder: Learning to filter netnews. Technical report, School of Computer Science, Carnegie Mellon University, 1995. http://anther.learning.cs.cmu.edu/ml95.ps.

[RIS+94] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186. Chapel Hill, NC, 1994.

[SM83] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[YGM95] Tak Y. Yan and Hector Garcia-Molina. Sift — a tool for wide-area information dissemination. In *Proceedings of the 1995 USENIX Technical Conference*, pages 177–186, 1995. ftp://db.stanford.edu/pub/sift/sift.ps.