A Brief Comparison of News Filtering Software

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

June 9, 1995

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 (*http://www.clarinet.com/newsclip.html*) 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.

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.

Retrieval	Rating
SIFT	NewsWeeder
SMART	
Browse	Browse
Lurker	
NewsClip	

Table 1: Filter systems categorized by model.

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 Comparisons

9.1 Model

The *model* dimension divides the systems in 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 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. Lurker and NewsClip are also in this category.

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

Programming	Query	Training	Collaborative
NewsClip Lurker	SIFT Browse SMART	NewsWeeder Browse	NewsWeeder

Table 2: Filter systems categorized by adaptation mode.

Boolean expression	Cosine measure	MDL	Neural network model
NewsClip SIFT Lurker	NewsWeeder SIFT SMART	NewsWeeder	Browse

Table 3: Filter systems categorized by classification method.

cancelling the messages that score beneath a threshold. NewsWeeder and Browse belong to this group.

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

9.2 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 six systems, with an extra column for NewsWeeder which has the extra capacity for collaborative filtering. Also note that Browser is basically a trained system, but still allows the user to place queries.

9.3 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.

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

- Boolean expressions
- The Cosine measure
- Minimum description length (MDL)

Interactive	Interactive and batch training	Batch
SMART	NewsWeeder Browse	Lurker SIFT NewsClip

Table 4: Filter systems categorized by operation method.

- Neural network modelling

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

9.4 Motivation

Motivation appears to be practically non-existent among the six filters. 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.

9.5 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 and batch training
- Batch

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

10 Conclusions

The six filter 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 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 is not 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 primarily deal with quantities and qualities from information retrieval; precision, recall and efficiency.

11 Comments

11.1 End-User Benefit

Very little beyond vague hints reflect how the end users use or experience the filters. The reasons for this may be that no-one has had the opportunity to perform user studies, in combination with the difficulty to have enough people signed up for the weeks needed to reach solid results.

There is also the difficulty how the benefit of the filter should be measured. Does it really suffice if the user claims to be relieved of noise and how to quantify this in order to compare different systems? How much of possible frustration is caused by a novel user interface and not attributable to the filtering mechanism per se?

11.2 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=.

[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.

[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.