Experimental Investigation of High Performance Cognitive and Interactive Text Filtering *

Douglas W. Oard and Nicholas DeClaris, Department of Electrical Engineering
Bonnie J. Dorr and Christos Faloutsos, Department of Computer Science
Gary Marchionini, College of Library and Information Services
University of Maryland, College Park, MD 20742
{oard|declaris|bonnie1|christos|march}@glue.umd.edu

Abstract

Text filtering has become increasingly important as the volume of networked information has exploded in recent years. This paper reviews recent progress in that field and reports on the development of a testbed for experimental investigation of cognitive and interactive text selection based on a history of user evaluations. An interactive filtering system model is presented and a new cognitive filtering technique which we call the Gaussian User Model is described. Because development of analytic measures of text selection effectiveness has proven intractable, we have modified the Cornell SMART text retrieval system to create a flexible text filtering testbed for experimental determination of filtering effectiveness. The paper concludes with a description of the design of this testbed system.

1 Introduction

Automatic filtering (or "selective dissemination") of information from text sources has become increasingly important as the volume of electronically accessible texts has exploded in recent years. Among these sources of electronically accessible texts are news stories, journal articles and electronic conference systems such as USENET. Text filtering systems are designed to sift through large quantities of dynamically generated texts and identify those which may be relevant to a user's interests.

Malone, et. al. present a taxonomy of filtering approaches, defining cognitive, social and economic filtering [6]. Cognitive filtering is "characterizing the contents of a message and the information needs of potential message recipients and then using these representations to intelligently match messages to receivers," while social filtering is based on "the personal and organizational interrelationships of individuals in a community." While both cognitive and social filtering are grounded in content, cognitive techniques work by modeling the user's interest based on direct evidence about content, while social filtering (often called "collaborative filtering") relies on indirect evidence gathered by observing other users' reactions to the texts they read. While we expect that the combination of cognitive and social filtering will often be superior to application of either technique in isolation, some small-scale experimental evaluations of social filtering have suggested that a critical mass must be reached before the benefits of this technique can be effective. [7] We believe that effective cognitive filtering provides a means to achieve this critical mass of users; thus the goal of our present research program is to experimentally evaluate the performance of cognitive filtering techniques.

Thirty years of research on text retrieval, in which the objective has been to select documents from relatively static collections, has produced a number of effective content-based text selection techniques [1]. Although some implementation details will differ, similar performance can be achieved when these techniques are applied to filtering applications [12]. Recently reported work has led us to conclude, however, that techniques which automatically develop a user model based on both document content and a significant history of observing the user's reactions to those documents have the potential to achieve excellent performance without extensive knowledge engineering [5, 13]. For this reason, we are particularly interested in considering the interactive application of cognitive filtering techniques.

*This work has been supported in part by NIH grant 1S10RR06460-01 (Medical Informatics Network), NSF award IRI-9357731, ARPA and ONR contract N00014-92-J-1929, ARPA contract DACA76-92-C009 and the Logos Corporation.

In order to provide a common frame of reference for evaluating cognitive filtering techniques we will adopt the following interactive filtering system model. Initially, documents may be sorted by source and/or category to facilitate browsing, or they may be initially filtered based on stereotypes [10] or the use of social filtering techniques [7, 9]. The user's response to each article is then observed and that information is used to organize the display of newly arrived texts. Because the computation for this retraining step can be expensive, we are interested in characterizing the impact of batching updates to the user model on both effectiveness and efficiency. To avoid confounding our training and evaluation data with incorrect inferences, we plan to rely on explicit binary ("like/hate") feedback in our experiment design and disregard documents which the user declines to score. We leave to future work the integration of less intrusive approaches based on indirect evidence [8, 11] once the effectiveness of the filtering algorithms have been demonstrated.

The past several years have seen a tremendous amount of activity in information filtering research.[1] In the next section we describe three promising cognitive text filtering techniques which might be practically applied to interactive applications and propose a new cognitive filtering technique which we call the Gaussian User Model. While analytic evaluation of computational efficiency is feasible, we have adopted an experimental approach to evaluating effectiveness because effectiveness has shown substantial variation when the same technique is applied to different test collections. Our experimental testbed is designed to perform side-by-side comparison of the effectiveness of competing algorithms, using similar parameters and the same test collection.

2 Text Filtering Algorithms

Text retrieval approaches applicable to information filtering include boolean, vector space, probabilistic, spreading activation and natural language processing techniques [1]. In the boolean approach, queries are expressed as boolean expressions with the indexed terms as the set of terminal symbols, and an unranked set of documents which satisfy the expression is returned. The vector space approach returns a ranked list of documents based on the similarity of each document to the query using a similarity function based on vectors which are constructed from information about the terms which occur in the documents. The probabilistic approach also returns a ranked list, but determines that ranking by estimating the probability that each document will satisfy the information need. Spreading activation methods rank documents using connectionist networks with weighted links. Natural language processing techniques perform sophisticated linguistic analysis to determine syntactic word class and semantic content. Although many of the approaches are intuitively appealing, the competitive performance and relative simplicity of the vector space approach have led to fairly wide acceptance and continued vector space application development. In particular, the four text filtering techniques we describe below are all based on the vector space approach. Accordingly, we begin with a brief discussion of that approach and a specific algorithm, Latent Semantic Indexing (LSI), which is common to three of the four techniques.

In the vector space approach the set of terms in a document is represented as a vector, where each component of the vector is some function of the frequency with which that term appears in the document. Step functions, logarithms and functions which account for the frequency of the term in the entire collection are all common. Queries for vector space information retrieval systems are typically expressed in natural language and their vector representation is computed in the same manner as that used for documents. One of the most widely used similarity measures is the cosine of the angle between two vectors, computed as the inner product of two normalized vectors.

LSI uses a more sophisticated approach to create the document vectors, but their similarity is computed in the same way [2]. From the complete collection of documents a term-document matrix in which each entry is some function of the number of occurrences of a specific term in a specific document is first formed. The Singular Value Decomposition (SVD) of this matrix is then computed and small singular values (thought to represent the contribution of term usage variations) are eliminated. The number of singular values to retain is a design parameter which can be decided using an inductive algorithm in which the retrieval effectiveness on an evaluation set is used to measure the effect of varying that parameter. Once the SVD is completed and the number of singular values to retain decided, the singular values and singular vectors can be used to map term frequency vectors for documents and queries into a subspace in which semantic relationships from the term-document matrix are preserved while term usage variations are suppressed. The span of this subspace is uniquely defined

[1] Links to every networked resource that we are aware of are at http://www.ee.umd.edu/medlab/filter/filter.html

by the set of singular vectors.

Foltz was the first to apply LSI to the text filtering problem [4]. He tried three cognitive filtering techniques on a small USENET news collection: closest match, average match, and clustering. Dumais has evaluated Foltz's average match technique in the first three Text REtrieval Conferences (TREC) [3]. In the average match technique, the vector representation of the information need is built as the mean of the concept vectors for relevant documents in the training set. Dumais reports that the performance of the average match technique exceeds that of a query-based LSI implementation, and the TREC-3 results show that its overall performance is competitive with techniques requiring more sophisticated linguistic analysis.

Hull has recently built on this work by applying LSI a second time to the reduced-dimensional vectors which represent relevant documents in a training collection, a technique which he calls Text-based Discriminant Analysis (TDA) [5]. His objective is to identify a small set of factors which explain the majority of the variance in the concept distribution of relevant documents. He then performs a modified version of discriminant analysis to compute the probability of relevance for newly arrived documents in the evaluation set. As with the original LSI step, the number of factors to retain is determined with an inductive algorithm. Hull's results on the Cranfield collection of 1,400 aeronautical abstracts[2] show substantial improvement over the average match technique.

Yang and Chute have also recently reported similar performance with a text categorization technique [13] which they call Linear Least Squares Fit (LLSF). They construct an optimal linear operator which minimizes the magnitude of the difference between the transformed document vectors and a given vector of category judgments on a training collection using only the term-document frequency matrix and a vector of relevance judgments. If relevance is viewed as a binary-valued category, LLSF can be applied to cognitive text filtering. Although Yang and Chute use a SVD to construct a pseudoinverse of the term-document frequency matrix when calculating the linear operator, their technique is fundamentally different from Hull's because no singular values are suppressed.

The Gaussian User Model

Our technique combines aspects of both Hull's TDA technique and Yang and Chute's LLSF. Like Hull, we base our model of the user's interest on the set of vectors in the reduced-dimensional LSI "concept" space associated with documents in the training collection which are judged by the user to be relevant. But like Yang and Chute's LLSF technique, we intend to maintain the contribution from every singular value and singular vector in that set rather than suppressing the smaller ones. The structure of our Gaussian User Model is based on the intuitive observation that when describing an interest, the range of acceptable "values" for specific features of that interest depends strongly on the feature. For example, the interest may be specific to the location, but insensitive to the time (or specific to the time, but insensitive to the location). Since Deerwester, et. al. [2] claim that the singular vectors retained by LSI represent concepts, it seems reasonable to consider interest representations which allow greater variation in one dimension than another.

Sample mean vectors and sample covariance matrices can be used to predict relevance by applying a threshold to the Mahalanobis distance between a statistical interest representation and each vector representing a new article. Mahalanobis distance is a distance measure which computes the difference between a deterministic vector and a random vector in an intuitively meaningful way based on the first and second moments of the random vector's distribution. The formal definition is

$$r^2 = (\bar{x} - \bar{\mu})^T \Sigma^{-1} (\bar{x} - \bar{\mu})$$

where r is the Mahalanobis distance, $\bar{x}$ is the deterministic vector and $\bar{\mu}$ and Σ are the mean vector and the covariance matrix for the random vector.

The first and second moments of a random vector uniquely describe a Gaussian distribution, and it is this observation which has led us to name this a "Gaussian User Model." For a multidimensional Gaussian distribution, the surfaces of equal probability density form hyperellipses, so the Mahalanobis distance is the probability density at which the deterministic vector is found. In directions with small variance, small Euclidean distances equate to large Mahalanobis distances. For directions with large variances, even large Euclidean distances will equate to relatively small Mahalanobis distances. To the extent that a distribution is well characterized by its first and second moments, Mahalanobis distance is essentially a measure of our "surprise" at encountering a specific instance of a random vector.

[2] Available at ftp://ftp.cs.cornell.edu/pub/smart/

Figure 1 provides a simple example in two dimensions. In that figure the ellipse depicts the contour described by a constant Mahalanobis distance, while the circle depicts a contour of constant cosine similarity with the same mean vector. Because the cosine measure is spherically symmetric, it is not possible to simultaneously discriminate tightly in one direction while accepting a large variance in another when that measure is used.

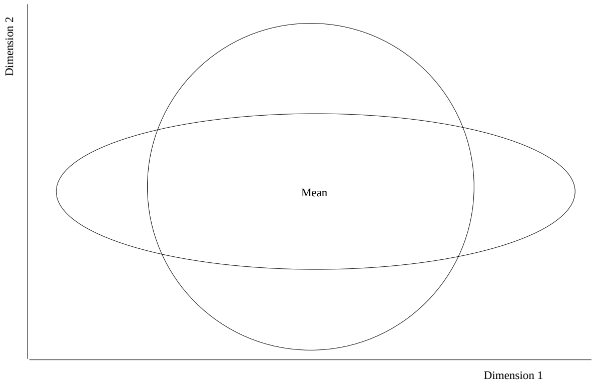

Figure 1: Contours of constant cosine similarity and Mahalanobis distance

Until the number of relevant documents in the training set exceeds the number of dimensions in the reduced-dimensional LSI space, the sample covariance matrix will not be invertible. Since between 100 and 300 dimensions are commonly retained in the reduced-dimensional LSI space, direct computation of the Mahalanobis distance will not be possible during the initial iterations of the interactive filtering process. We are evaluating two alternatives to avoid this difficulty. When the number of missing dimensions is small, we are considering heuristically extending the eigenvalues of the sample covariance matrix to achieve full rank. For example, we could replace every zero eigenvalue with a value equal to one half of the smallest non-zero eigenvalue. An alternative, which we expect will be useful when even fewer relevant documents are available, is to reduce the dimensionality of the LSI representation. Because singular values are, by convention, sorted in decreasing order, this is easily implemented by truncating the LSI document vectors.

3 Performance Evaluation

Two fundamental measures of effectiveness for a text filtering system are effectiveness and efficiency. Precision and recall are commonly used to characterize effectiveness. Precision is the ratio of relevant documents that are retrieved to the total number of documents that are retrieved. High precision results when a low false alarm rate is achieved. Recall is the ratio of retrieved relevant documents to available relevant documents. High recall is equivalent to a low rate of false dismissals. Precision and recall must be measured concurrently in a ranked text selection system because it is possible to increase one at the expense of the other by adjusting the number of documents examined. Often the time required for indexing and retrieval for a standard test collection are reported as measures of efficiency for text retrieval systems. For cognitive text filtering systems, the time required to exploit new evidence and to process newly arrived documents are the analogous measures. Because our emphasis is on interactive applications, the “retraining time” needed to exploit additional observations is of particular significance.

Evaluation of recall and precision requires knowledge of the actual relevance judgment for each document. Our insistence that the user provide explicit binary relevance judgments will make it possible to measure these parameters for the set of documents which the user chooses to evaluate. Unfortunately, the very interaction we wish to foster and evaluate would create significant difficulties for other researchers wishing to duplicate reported results. We believe that interactive evaluations will be essential to evaluate usability, but that repeatable experiments using standard document collections are important when developing and evaluating algorithms.

Because the cognitive filtering systems we are considering will benefit from a large set of interest evidence in a “training collection”, a straightforward partition of a standard document collection into training and an evaluation subsets would require a very large document collection. This is the approach taken in TREC. Because resource constraints make it impractical to score every document in such a large collection against every possible interest, TREC uses a pooled relevance methodology in which only documents selected by at least one of the participating systems are scored and the remaining documents are assumed not to be relevant. The pooled relevance methodology results in accurate calculation of precision, but calculates only an upper bound for recall. Because all participating systems are subject to the same limitations with regard to recall calculations, however, this issue does not impede comparisons between participating systems.

An alternative which is practical for smaller collections in which every document can be judged, but in which there may now be relatively few documents relevant to a specific interest, is the cross-validation technique used by Hull [5]. In cross-validation, a single relevant document is omitted from the training collection, the remainder of the documents (including the omitted document) are ranked, and the rank of the omitted document is recorded. By repeating this procedure with each document omitted individually, a set of rankings is obtained which can be used to compute precision and recall. Essentially, cross-validation determines the typical performance of a routing algorithm for the last document to arrive by using multiple trials in which each document has the opportunity to be "last." Because the pooled relevance methodology used in TREC is relatively expensive and requires that evaluation efforts be coordinated across development teams, cross-validation is a useful technique, particularly for initial development work on new filtering algorithms.

Evaluation Resources

Because text retrieval effectiveness must be studied empirically, a number of retrieval evaluation systems and standard test collections have been made available by researchers in that field. The SMART information retrieval system is one of the most flexible systems for evaluation of vector space techniques. Since we are not aware of any similar widely distributed system for evaluation of vector space text filtering techniques, we have modified the SMART version 11.0 distribution[3] to incorporate the necessary capabilities. SMART is particularly well suited to experimental evaluation of vector space text selection algorithms because it contains extensive functionality for experiment control in addition to a full set of vector manipulation capabilities and a modest interactive interface.

Because the techniques we are interested in comparing all require a singular value decomposition on a large sparse matrix, our first step was to integrate the C language version of SVDPACK,[4] a SVD package which was developed with LSI applications in mind. Because Hull's TDA technique requires computation of two singular value decompositions (although the second matrix is not sparse), so we have included provisions for reusing this code. We achieved some savings in integration effort (at the expense of execution efficiency) by also using the same SVD code to invert the sample covariance matrix when implementing our Gaussian User Model technique. The efficiency penalty is not particularly significant because we plan to use our testbed only for effectiveness measurements and to determine efficiency analytically. Our experiments with the Gaussian User Model also requires that that the singular vectors span the complete space when inverting the sample covariance matrix. We have added this capability to SVDPACK using the Gramm-Schmidt algorithm.

Because SMART was developed to evaluate vector space information retrieval techniques, it contains capabilities for reading and processing relevance information which we explot to obtain and store the relevance evaluations required to train our cognitive filtering algorithms. We have implemented the training portion of our algorithms with cross-validation in mind, including provisions to remove any individual document from the training collection using a run-time parameter. SMART's run-time parameter mechanism is quite capable, and we also use it to select file for SVD data in order to avoid unnecessary SVD recomputation when performing multiple evaluation runs.

To date we have implemented Foltz's average match technique and our Gaussian User Model and we are presently using the system to evaluate alternative techniques for inverting the sample covariance matrix. SMART allows run-time subroutine selection, making performance comparisons straightforward. We expect to be able to implement LLSF with the components we have completed, but some additional code will be required for the discriminant analysis component of TDA.

4 Conclusion

The excellent performance reported recently been reported for vector space techniques with obvious applications to interactive cognitive filtering has led us to develop a capability for evaluating these techniques under controlled conditions. We hope to be able to demonstrate a level of performance which will encourage more widespread use of text filtering technology in areas such as selective dissemination of literature in medicine, business and academia, construction of personalized digital newspapers, library collection development, and internet resource discovery.

References

[1] N. J. Belkin and W. B. Croft. Information filtering and information retrieval: Two sides of the

[3] Available at ftp://ftp.cs.cornell.edu/pub/smart/

[4] Available at http://www.netlib.org/svdpack/

same coin? *Commun. ACM*, 35(12):29–38, Dec. 1992.

[2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[3] S. T. Dumais. Latent semantic indexing (LSI): TREC-3 report. In D. Harman, editor, *Overview of the Third Text REtrieval Conference*, pages 219–230. NIST, Nov. 1994. http://potomac.ncsl.nist.gov/TREC/.

[4] P. W. Foltz. Using latent semantic indexing for information filtering. In F. H. Lochovsky and R. B. Allen, editors, *Conference on Office Information Systems*, pages 40–47. ACM, April 1990.

[5] D. Hull. Improving text retrieval for the routing problem using latent semantic indexing. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 282–291. Springer-Verlag, July 1994.

[6] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen. Intelligent information sharing systems. *Commun. ACM*, 30(5):390–402, May 1987.

[7] D. A. Maltz. *Distributing Information for Collaborative Filtering on Usenet Net News.* PhD thesis, MIT, Department of Electrical Engineering and Computer Science, May 1994. http://www.xerox.com/PARC/dlbx/tapestry-papers/maltz.ps.

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

[9] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An open architecture for collaborative filtering of netnews. In R. K. Faruta and C. M. Neuwirth, editors, *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 175–186. ACM, Oct. 1994. http://www-sloan.mit.edu/ccs/CCSWP165.html.

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

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

[12] T. W. Yan and H. Garcia-Molina. Index structures for selective dissemination of information under the boolean model. *ACM Transactions on Database Systems*, 19(2), June 1994.

[13] Y. Yang and C. G. Chute. An example-based mapping method for text categorization and retrieval. *ACM Transactions on Information Systems*, 12(3):252–277, July 1994.