Keywords: collaborative filtering, content-based filtering, profile learning, routing.

Related, but not Relevant: Content-Based Collaborative Filtering in TREC-8

Ian M. Soboroff and Charles K. Nicholas
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
{ian,nicholas}@cs.umbc.edu

Abstract

Historically, solutions to the TREC filtering tasks have focused exclusively on the content of documents and search topic descriptions as training data. These approaches are well-known for their ability to focus on those salient concepts in the document stream which are most useful for separating relevant documents from irrelevant ones. However, one kind of information that has not been used is the relationships among the topics themselves. In our TREC-8 routing experiments, we employed a collaborative (or social) filtering algorithm, based on latent semantic indexing which highlights common term usage patterns among groups of filtering profiles. Our hypothesis was that this would allow related topics to share common relevant documents. We found, however, that the algorithm also recommends many documents of related, yet irrelevant interest. As a result of this process, many similar search topics are "linked" together by common sets of documents recommended to them. We visualize these topic relationships using graphs where topics are nodes and edges exist where two topics share a recommended document.

1 Introduction

In describing their Information Lens system, Malone et al. (1987) identified three possible methods for the automated filtering of documents: *cognitive*, that is, based on the content of the message and information needs of the recipients; *social*, based on knowledge about the sender and endorsements by colleagues; and *economic*, based on the cost to both the sender and receiver.

Solutions to the filtering problem in the TREC workshops have chiefly employed cognitive, more commonly called content-based, techniques for learning user profiles, examining new documents, and making filtering recommendations. Some examples of the algorithms which have been used are relevance feedback (Singhal et al., 1998), K-nearest neighbors (Ault and Yang, 2000), and latent semantic indexing (Dumais, 1994). Schütze et al. (1995) examined several text classification algorithms and their performance in

the routing task using the TREC-2 and 3 collections. All of these algorithms attempt to identify the best textual document features for correctly classifying a document as relevant or irrelevant to a user's information need.

In contrast, social or collaborative techniques have not been tried in the TREC domain. The reasons for this are largely historical, in that the TREC filtering tasks grew out of the ad hoc retrieval community, where the experimental methodology assumes that users' information needs occur in isolation. Additionally, because the TREC topics are contrived specifically for TREC and are not designed to interact with each other, there may also be good data-dependent reasons to assume that collaborative approaches would not be applicable.

In practice, however, TREC topic sets often contain clusters of interests in such domains as health and medicine, foreign affairs, economics, and education. Topics within these clusters are often closely related. While they do not share a great number of relevant documents, it is possible that more relevant documents may be found by incorporating common features from the cluster in each search profile. Further, collaborative approaches may help find related documents not strictly relevant to the topic, but that help the user to gain a greater understanding of the broader collection and the relationship of his interest to it. Related documents don't directly improve a TREC score (which is based on finding relevant documents only), but in actual retrieval and filtering settings, these documents help to support the final information result of a user's search process (O'Day and Jeffries, 1993).

This paper has two goals. First, we describe our experience with a collaborative filtering algorithm in the TREC-8 routing task. We designed this algorithm to discover conceptual relationships occurring in a community of content-based filtering profiles. The collaborative algorithm yielded a lower average precision score compared to using the content-based profiles alone. However, a visualization of topic relationships based on patterns of filtering recommendations shows that the collaborative algorithm clusters related topics together. Second, we place that experience in the larger context of both collaborative filtering and classical information retrieval experiments. The concept of "relatedness" is known to exist, but is even harder to define than relevance itself; our analysis takes a small step toward quantifying that relationship.

The first part of the paper introduces collaborative filtering, both pure and hybrid approaches. The second part describes our content-based collaborative filtering algorithm which is based on latent semantic indexing. The third part presents our experimental results and analysis of topic relationships and collaborative potential. We conclude with a summary of our findings and thoughts for the future.

2 Collaborative Filtering

Collaborative filtering is the name given to an assortment of techniques that attempt to leverage the activity of an entire community of information seekers. Classical or pure collaborative filtering systems collect ratings of documents (which are as often books, movies, or music CDs as news articles), and compute correlations between users in order to predict how those users will rate documents they have not yet seen. The prediction for user u regarding document d is given by

$$pred(u, d) = \sum_v corr(u, v) \cdot r_{v,d}$$

where $r_{v,d}$ is the ratings by user v for document d. In practice, the prediction may be computed from only the highest-correlated users who have rated document d. The approach is collaborative because predictions are generated on the basis of similarity among users; document content is ignored. This particular approach to collaborative filtering was pioneered by the GroupLens project at the University of Minnesota (Resnick et al., 1994) and by Ringo at MIT (Shardanand and Maes, 1995), and has been used commercially by e-commerce web sites such as Amazon.com.

The highest predictions are presented to the user as recommendations, hence these systems are often called "recommender systems", although in truth most information search, browsing, and filtering systems can be thought of as recommender systems. It is most useful to think of filtering as the task domain, and collaborative filtering as a class of solutions applied in that domain.

2.1 Combining Content-based and Collaborative Filtering

Both collaborative and content-based filtering have inherent strengths and weaknesses. Content-based filtering is good at identifying new documents that are on topics similar to what has been seen before. A user represents his information need with example documents on his topic of interest, and a content-based filtering system finds new documents that are similar to them. Collaborative filtering can recommend anything that users can provide ratings for, so documents need not be analyzed. A collaborative filter recommends based on quality, in the sense that the system will recommend items that have been rated highly by many like-minded users.

On the other hand, it can be difficult for users to understand collaborative filtering recommendations. The system can identify those neighboring users or their profiles that had a high influence on the recommendation, but this may not be much help if those users are not well-known. Furthermore, a purely collaborative system can't handle unrated documents or new users. Conversely, content-based systems are limited by

their representation of content; they can't recommend a document that's like nothing seen before. The respective strengths and weaknesses of each filtering approach are complementary. A system that combines content-based and collaborative filtering should exploit the strengths and limit the weaknesses of both approaches.

Content-based filtering and collaborative filtering can be combined at two different levels. At an algorithmic level, content and collaborative information can be combined within the model for computing predictions; this is the approach we applied in TREC-8 and which we describe below.

At an application level, content or collaborative information can be used to supplement a system based on the other, in order to provide a more useful information system. For example, user ratings can be used to supplement text representations for partially-textual or multimedia documents, with ratings treated similarly to text features in search algorithms. In this example, a basic search algorithm that compares document features to an ad hoc query or other documents would remain unchanged. This was the approach taken in the Tapestry system (Goldberg et al., 1992), where documents were annotated with reviews and other collaborative features which could then be queried in an ad hoc fashion. Content information can likewise be used to give context and ensure topicality for collaborative recommendations.

3 Content-based Collaborative Filtering with LSI

In this section, we introduce the filtering algorithm that we used in the TREC-8 routing task. The algorithm is a variant of latent semantic indexing (LSI), a content-based technique that improves retrieval effectiveness by exploiting term co-occurrence patterns to reduce the dimensionality of the feature space. Our application of LSI is a little unusual; rather than computing the latent semantic index from a collection of documents, we compute it from a collection of filtering profiles. By building a reduced-dimensional representation of the space of user profiles based on term co-occurrence between them, we hoped to improve filtering effectiveness. We begin with a short overview of latent semantic indexing and its applications in the filtering domain, and then present our specific variation of the algorithm and our experimental results.

3.1 Latent Semantic Indexing

Latent semantic indexing is an enhancement to the familiar vector-space model of information retrieval (Deerwester et al., 1990). Typically, authors will use many words to describe the same idea, and those words will appear in only a few contexts. LSI attempts to highlight these patterns of how words are used within a document collection. By grouping together the word co-occurrence patterns that characterize groups of

documents, the "latent semantics" of the collection terms are described. Themes in the document collection arise from subsets of documents with similar word co-occurrences.

Specifically, each document is represented by a vector of terms, whose values are weights related to their importance or frequency of occurrence. The collection of documents, called the *term-document matrix*, is decomposed using the singular value decomposition or SVD

$$M = T\Sigma D^T$$

The columns of T and D are orthonormal, and are called the *left* and *right singular vectors*. Σ is a diagonal matrix containing the *singular values* σ, ordered by size. The singular values of M are the eigenvalues of the matrix MM^T. If M is $t \times d$ and of rank r, T is a $t \times r$ matrix, D is $d \times r$, and Σ is $r \times r$.

The SVD projects the documents in the collection into an r-dimensional space, in contrast to their t-dimensional representation in the term-document matrix. This LSI space is spanned by the columns of T, and it is useful to think of $T\Sigma^{-1}$ as a projection matrix for casting arbitrary document vectors into the LSI space. Specifically, multiplying document vector i from the original term-document matrix by $T\Sigma^{-1}$ yields the ith column of D, the document's representation in the LSI space. We can map any document vector into the LSI space in this way, and compare documents by taking the dot product of their LSI representations. This operation is called "folding in". We think of it as projecting the new document into the LSI space using the T matrix, because it is the same as projecting points using an affine transformation matrix in geometry or computer graphics.

In comparing documents with LSI, Deerwester et al. show that one uses $D\Sigma$ as the space for comparison, because the matrix of dot products between all documents in M, $M^TM = D\Sigma\Sigma D^T$. Since folding in a document involves a multiplication by Σ^{-1} and we usually compare it to a document in $D\Sigma$, in practice the Σ terms are dropped since they cancel each other out.

An important feature of the SVD is that the singular values are ordered by magnitude, and give an indication of the relative importance of each dimension. One can choose how many dimensions to retain by eliminating low-valued dimensions, in other words, setting some of the singular values on the diagonal of Σ to zero. If all dimensions are kept, then document similarities computed in the LSI space using T or D are the same as they were using the original term-document matrix. If one keeps k dimensions ($1 \leq k < r$), then the matrix product

$$M_k = T_k \Sigma_k D_k^T$$

is the closest rank-k approximation to the original term-document matrix M (Berry et al., 1995). Document

comparisons in this truncated LSI space should be more meaningful because unimportant term relationships are disregarded. Choosing the best value of k is an open problem, the solution to which is usually approximated by testing several values of k with a training collection.

The most common application of LSI has been in retrieval. The work at Bellcore described by Deerwester et al. (1990) and Berry et al. (1995) showed improvements in retrieval effectiveness using a variety of small, topically-focused datasets such as Cranfield, CISI, and CACM. In TREC-3, the 199-dimensional representation used yielded only small performance gains (Dumais, 1994). This was due at least in part to the term weighting strategy chosen, but may have also been a result of computing an SVD from a random sample of the entire collection of TREC newswire stories.

The eigenanalysis approach of LSI has also been applied in the domain of hypertext ranking. Kleinberg's HITS algorithm is an iterative algorithm which identifies web pages that are authorities (good reference pages) or hubs (pages that point to many good references). The hub and authority values resulting from the algorithm are the first left and right singular vectors of the web graph adjacency matrix (Kleinberg, 1999). In other words, authoritative web pages are those which dominate co-occurrence patterns of hyperlinks on the web, much as stopword usage dominates term co-occurrence patterns in texts.

The principal challenge in applying LSI to large data collections is the cost of computing and storing the SVD, prohibitively high in the days of the early LSI experiments. Straightforward computation of SVDs for large dense matrices can be quite expensive, but several fast, low-memory algorithms exist for the sparse matrices usually found in retrieval applications (Berry, 1992)[1]. To make things more complex, in filtering applications the document collection and the filtering profiles change over time, so SVDs of either or both need to be recomputed or updated if LSI is used for filtering. Berry et al. (1995) described the basic SVD updating algorithms, which can approach the complexity of fully recomputing the SVD. Zha and Simon (1999) developed several more efficient updating algorithms for LSI. Kolda and O'Leary (1996) developed an alternative matrix decomposition, the semi-discrete decomposition, which is more expensive to compute initially but is much cheaper to update than the SVD. In terms of alternatives to the SVD, probabilistic models with LSI-like behavior and better explanatory properties have recently been proposed (Hofmann, 1999; Ding, 1999).

[1] Briefly, the complexity of these algorithms is polynomial in the number of nonzeros in the matrix, but due to their iterative nature the costs are more complicated than that. See (Golub and Van Loan, 1996; Berry, 1992; Berry et al., 1995) for more details.

3.2 LSI in Content Filtering

Latent semantic indexing was applied to routing as well as ad-hoc retrieval in TREC-3 (Dumais, 1994). The LSI space was first computed from a collection of documents, and then profiles were constructed as centroids of document representations from the LSI space. The gain in performance in routing was greater than in retrieval.

Hull (1994), comparing LSI to other dimension reduction algorithms for filtering and routing applications, computed the LSI from a set of documents known to be relevant to the profiles. In later experiments, a "local LSI" was built from documents similar to a given profile (Schütze et al., 1995). This is different from the random sampling used by Dumais. The key insight here is that the LSI projection can be trained from an arbitrary set of document vectors, as long as they adequately cover the set of terms we expect to occur in future documents. As Hull observed, the LSI can describe the occurrences of terms across only the documents which are used in the SVD computation. Thus, just as appropriate training data is necessary in machine learning applications, one should compute the SVD from a document collection containing a distribution of terms across documents that is relevant to the task.

3.3 LSI in Collaborative Filtering

Latent semantic indexing has not been widely applied in the field of collaborative filtering. Billsus and Pazzani (1998) proposed a machine-learning approach to collaborative filtering which used an SVD of a user-by-document matrix of ratings to compute a lower-dimensional representation of the rated documents. In their experiment with the EachMovie collection, they first converted the ratings in the matrix to a Boolean "like" or "dislike" value by setting a threshold rating, and then computed the SVD of this matrix. The left singular vectors corresponding to the training movies for a user were then fed to an artificial neural network, which learned the difference between the user's rating for an item and its average rating.

In the Jester system (Gupta et al., 1999), the principal components of the ratings matrix are computed and used in a hierarchical clustering algorithm. Principal components analysis is closely related to singular value decomposition and eigenvalue analysis. Gupta et al. used their algorithm to recommend jokes, but did not apply it to any standard test collections.

3.4 A Content-based Collaborative Filtering Algorithm using LSI

The specific filtering algorithm we used in TREC-8 is as follows. First, we obtain a content-based profile for each user, with each profile being a single term-weight vector. In our experiments, these vectors are created using Rocchio's relevance feedback algorithm, but in practice any similar algorithm could be used.

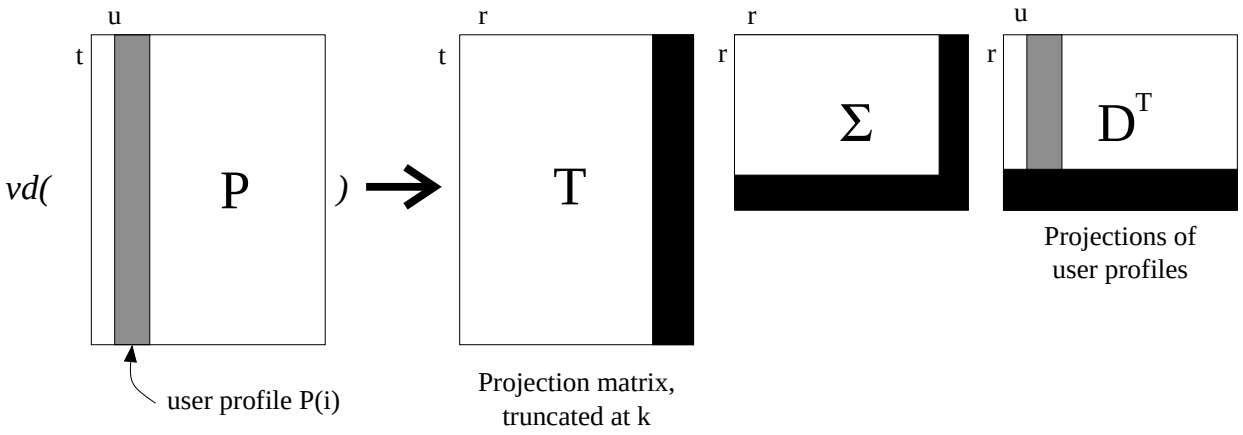

Figure 1: The profile matrix, and the result of computing its SVD. The profile matrix P (left) is a $t \times u$ matrix for t terms and u user profiles. The SVD matrix T is at most $t \times r, r = rank(P)$, but has been truncated to k dimensions. Likewise Σ and D.

We then collect these profiles into a term-by-user matrix P, each profile becoming a column of the matrix as illustrated in Figure 1. This profile collection is analogous to the document collection used in LSI retrieval, in that this matrix of profiles will be our input to the SVD algorithm.

The result of computing the SVD of the term-by-user matrix of profiles is a projection of the profile matrix which highlights common term co-occurrence patterns among the profiles. More specifically, the T matrix can be used to project new documents into the profile LSI space. The D matrix contains the given user profiles already projected into this space. We reduce the dimensionality by selecting a subrange of the singular values $\sigma_1 \ldots \sigma_k$ and setting the remaining values along the diagonal of Σ to 0. The resulting truncated T_k will eliminate "noisy", low-frequency term co-occurrence patterns found among the profiles while retaining the most prominent ones.

The T_k matrix is used to filter incoming documents as follows. An incoming document is represented as a vector of term weights like a profile, as if we were doing straightforward content-based filtering. But instead of comparing the document vector v directly to the profile vectors, we first project the document into the LSI space by multiplying it by T_k, and then take the dot products between $v^T T_k$ and the LSI profile vectors in D_k.

Our approach differs from the LSI content filtering approaches in that the LSI space is computed from the collection of profiles, rather than a collection of documents. This means that commonalities between profiles guide the construction of the LSI space. In the traditional use of LSI for retrieval, only overlap among documents is able to affect the SVD. Collaboration occurs as the LSI describes the relationships of important concepts across profiles. Projecting a new document into this space should allow us to better view

the document's relation to the group of profiles.

This algorithm is similar to Hull's local LSI, in that we compute the SVD from the specific data we hope to differentiate. It is also related to applying LSI to the ratings matrix directly, in that the content-based profile vectors are closely related to the document-by-user ratings matrix. We can transform the ratings matrix such that an entry contains a 1 if the user's rating for that document exceeds some minimum threshold. Each column is then divided by the number of documents exceeding the threshold for that user. Multiplying this profile-construction matrix by the term-document matrix for the document collection produces the profile matrix.

Both of these matrices, the ratings matrix and the content-based profile matrix, model the universe of user interests. One considers document objects explicitly and separately, while the other pools the document contents to give "ratings" of actual content terms. Neither representation is necessarily collaborative in any way. With the ratings matrix, collaboration occurs when user-user correlations are used to predict new ratings. For collaboration within the content profile matrix, we compute the LSI of the content-based profile matrix.

We conducted initial experiments with this technique in the Cranfield collection, with LSI giving between 4 and 40% improvement in performance over content-based profiles alone (Soboroff and Nicholas, 1999; Soboroff, 2000).

4 TREC-8 Routing Experiment

In the TREC-8 filtering track, there were three tasks: routing, batch filtering, and adaptive filtering (Hull and Robertson, 1999). Each task had a slightly different training and test document set. We participated in the routing task in order to focus on profile construction and finding communities of information interests, rather than on profile or filtering threshold adaptation. Had we done adaptive filtering, it might have been difficult to judge the impact of the LSI technique over whatever profile adaptation technique we might have used.

In the routing task, the test collection was articles from the Financial Times between 1993-4, and profiles could be trained using any other documents and judgments the participant chose. All of the test documents were ranked against the topic queries, and the top 100 documents were evaluated using precision and recall.

TREC topics are not designed to overlap, either in information interest or in actual relevant document sets. However, just from a reading of the topic descriptions, several topics in the TREC-8 Filtering task seem closely related, as can be seen in Figure 2. These groups might have documents in common, for example, in the case of the fuels and education groups; or they might indeed be "false friends", containing common

- Medicine:
  - postmenopausal estrogen Britain (356)
  - in vitro fertilization (368)
  - anorexia nervosa bulimia (369)
  - health insurance holistic (371)
  - obesity medical treatment (380)
  - alternative medicine (381)
  - mercy killing (393)
- Alternative fuels:
  - hydrogen energy (375)
  - hydrogen fuel automobiles (382)
  - hybrid fuel cars (385)
- Exploited labor:
  - clothing sweatshops (361)
  - human smuggling (362)
- Pharmaceuticals:
  - food/drug laws (370)
  - mental illness drugs (383)
  - orphan drugs (390)
  - R&D drug prices (391)
- Education:
  - mainstreaming (379)
  - teaching disabled children (386)
  - home schooling (394)

Figure 2: A sampling of topics used in the TREC-8 Filtering track, grouped manually into families of related interest.

terms but not common relevant documents, probably the case in the other three groups. In fact, because of the strict definitions of relevance in TREC topics, and how they explicitly seek to limit how far relevance carries to related documents (see Figure 3 for an example), collaborative filtering techniques might actually harm performance.

That said, we felt it would be worthwhile to explore collaborative filtering in the TREC data set. The collaborative filtering community does not have many large data sets freely available to researchers, and the data sets that are available predominantly feature ratings of movies. This has caused the collaborative filtering literature to focus overly much on recommending movies (a task domain with vague utility and a relatively low cost for making a bad recommendation) at the expense of other domains. Thus, we chose to try our techniques in TREC in order to add to the collections used in collaborative filtering experiments, as well as to consider the possibilities of existing text collections and the requirements for future ones.

4.1 Profile Construction

To build our profiles, we adopted a technique similar to that used by the AT&T group in TREC-6 (Singhal, 1997) and TREC-7 (Singhal et al., 1998). First, a training collection was constructed from TREC discs 4 and 5 using the Financial Times documents from 1992, all documents from the Foreign Broadcast Information

```
<top>

<num> Number: 351
<title> Falkland petroleum exploration

<desc> Description:
What information is available on petroleum exploration in
the South Atlantic near the Falkland Islands?

<narr> Narrative:
Any document discussing petroleum exploration in the
South Atlantic near the Falkland Islands is considered
relevant.  Documents discussing petroleum exploration in
continental South America are not relevant.

</top>
```

Figure 3: A TREC topic. Note than relevance is strictly defined. Some related but irrelevant documents describe disputes between Argentina and the UK over other natural resources in the Falklands.

Service, and the Los Angeles Times documents. We gathered collection statistics here for all future IDF weights. The training document vectors were weighted log-tfidf, and normalized using pivoted unique-term document normalization (Singhal et al., 1996).

Pivoted document length normalization is an improvement over the more commonly-used cosine normalization. Vector normalization is done in general because longer documents, having more terms, will dominate the similarity calculation otherwise. The cosine normalization does a fairly good job of ensuring that probability of relevance does not increase with length, but still manages to favor very long documents. Pivoted normalization repairs this by more severely normalizing longer documents. This helps in filtering and routing, where the varying amount of training data per profile causes profiles to differ widely in length.

We then built a routing query using Rocchio's formula for relevance feedback (Salton, 1971):

$$Q' = \alpha Q + \beta \left(\frac{1}{|rel|} \sum_{r \in rel} D_r \right) + \gamma \left(\frac{1}{|nrel|} \sum_{n \in nrel} D_n \right)$$

An initial query Q is made from the short topic description, and using it the top 1000 documents are retrieved from the training collection. The results from this retrieval are used to build a feedback query, using:

- Q, the initial short-description query (weighted $\alpha = 3$)

- D_r, all documents known to be relevant to the query in the training collection (weighted $\beta = 2$)
- D_n, retrieved documents 501-1000, assumed to be irrelevant (weighted $\gamma = -2$)

The set of documents retrieved with the initial query Q is called the "query zone" (Singhal et al., 1996), and this blind feedback is a kind of unsupervised learning technique. One can also use the top documents from the query zone as unsupervised positive examples, but we found this did not perform as well against the training set. Also, we looked at using the known irrelevant judgments as supervised negatives, but these did not perform as well at retrieving the training set.

The choice of α, β, and γ are somewhat arbitrary, but reflect our intention to weigh the original query terms most highly, and use the weights of the blind-irrelevant documents to eliminate distracting terms in the positive training documents. Singhal et al. (1997) used $\alpha = 8, \beta = 64, \gamma = -64$. This would give terms from the query zone examples much more importance than the original query terms. Their rationale for using these specific values is not given.

4.2 Software Architecture

For our experiments, we used the SMART system with several modifications to support latent semantic indexing, storing feedback profile collections, and pivoted length normalization. The LSI code is based on software written at the University of Maryland,[2] and on SVDPACKC from the NETLIB archive.[3]. Query zoning was implemented using SMART's retrieval routines and standard UNIX shell tools such as AWK. The experiments were carried out on a Intel Pentium II-based system running Linux 2.2 with 512MB of physical RAM and 36GB of local SCSI-II disk space.

4.3 Results

Two routing runs, or lists of 1000 retrieved documents per topic, were submitted for TREC-8. The first run, labeled `umrqz`, used only the routing queries built with Rocchio's algorithm as described above. The second, `umrlsi`, computed an LSI from the collection of these routing queries, and routed the test documents in the resulting LSI space.

As we have already mentioned, for LSI to improve performance, the dimensionality must be reduced below the rank of the profile matrix (in this case, 50 dimensions). We evaluated our LSI profiles' ability to retrieve their constituent documents in the training collection at several dimensions, and found that no

[2] Available from `http://www.glue.umd.edu/~oard/`
[3] SVDPACKC is available from `http://www.netlib.org/`

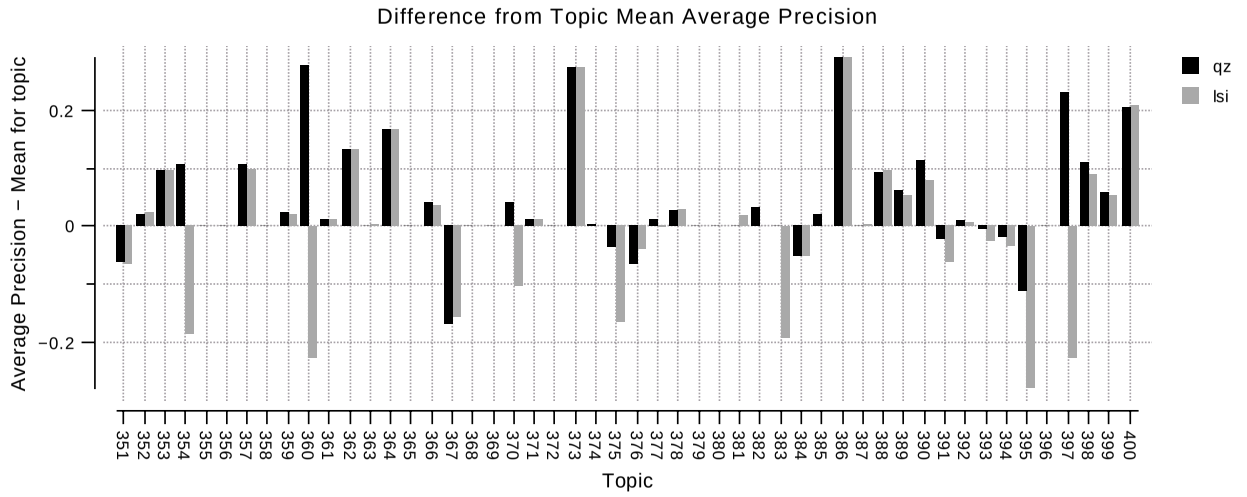

Figure 4: Difference in average precision from mean of all participating systems for each topic. For some topics, LSI made a dramatic difference in performance; for most topics, we achieved similar performance whether LSI is used or not.

dimensionality choice seemed to show any improvement for LSI. For the official submission, we arbitrarily chose 45 dimensions. The official average precision was 0.405 for `umrqz` and 0.364 for `umrlsi`.

Overall, both runs performed around the middle of the pack, with `umrqz` above the median average precision for 27 topics, and `umrlsi` for 23. For five topics, our runs produced the best performance of any participant, and for four of those topics, the LSI gave the maximum score.

For the majority of queries, however, there was only a very small difference in performance between the two runs. This indicates that good overall performance was mostly due to the routing query construction, which used a combination of approaches shown to work well in previous TRECs. Figure 4 shows the difference in average precision from the mean score of all participating systems for each topic, illustrating the similarity of the results.

We expected that LSI might not increase performance for most topics. Since most of the topics are quite different in their particular subject, with little opportunity for overlap, the LSI should have been unable to help. However, for the example candidate topic "clusters" described above, the difference in average precision from using LSI was negligible. For some of the drug-related topics, performance was much lower with LSI.

In 18 topics, the difference in average precision between the non-LSI and LSI routing was more than 0.009. In 11 of these cases the difference was quite small relative to the whole span of scores. In the other seven, the difference was more marked. The difference was against LSI in all but one (topic 381). For one

topic (360), LSI gave the minimum performance and the non-LSI query gave the maximum.

Furthermore, in the twenty topics where average precision in the `umrlsi` run was high (> 0.5), precision without LSI was either the same or slightly higher.

In eight topics, the LSI average precision was less than 60% of that achieved without LSI. These topics have a fair range of relevant document set sizes and in only one of these topics was performance across all systems poor. One topic in this group was 375, "hydrogen energy", and three were drug-related (drug legalization, food/drug laws, mental illness drugs). It may be that the drug-related topics contained a lot of shared terms, but this caused LSI to bring out false friends. We will explore this possibility further in the next section.

5 Discussion

The results indicate that, for the topics and documents here, LSI overall does not improve precision over non-transformed profiles, and if anything may degrade precision among manually-identified clusters of interest.

One explanation for this might be that the topics have no overlap in relevant documents. If the topics were truly orthogonal, so that there was no overlap among highly-weighted terms among profiles, then we would expect the LSI to give results that are identical to the non-transformed queries, or nearly so. This agrees with the results as shown in Figure 4. However, we know from Figure 2 that there are groups of topics which share the same subject area or have a closely related focus. It may be that collaborating queries are "sharing documents", that is, LSI is boosting the ranking of certain common documents among sets of topics. These documents relate to the general interest of the topic, but are not actually *relevant* to the topic as determined by the TREC evaluation.

Alternatively, our profile vectors may not give a good representation of the topic. Perhaps we are using too many negative example documents, or should be more selective about which terms to retain after the Rocchio expansion. To analyze this, we could look at overlap in terms, training documents, and test documents among the topics. This should give us a better view of where to expect LSI to make gains, but on the other hand this is what the LSI is supposed to do for us. It might be instructive to look at the LSI dimensions and the terms which characterize them, to see exactly what patterns the LSI is finding.

As another hypothesis, it may be that there are topics which have collaborative potential, and in fact there are term co-occurrence patterns across their profiles which we'd expect the LSI to find, but these patterns aren't sufficiently prominent relative to the rest of the collection. This might happen because there aren't enough terms co-occurring, or the pattern doesn't span enough profiles. In our three example groups, only drug-related topics represent a large segment of the topic collection, and this grouping is broadly-defined. An

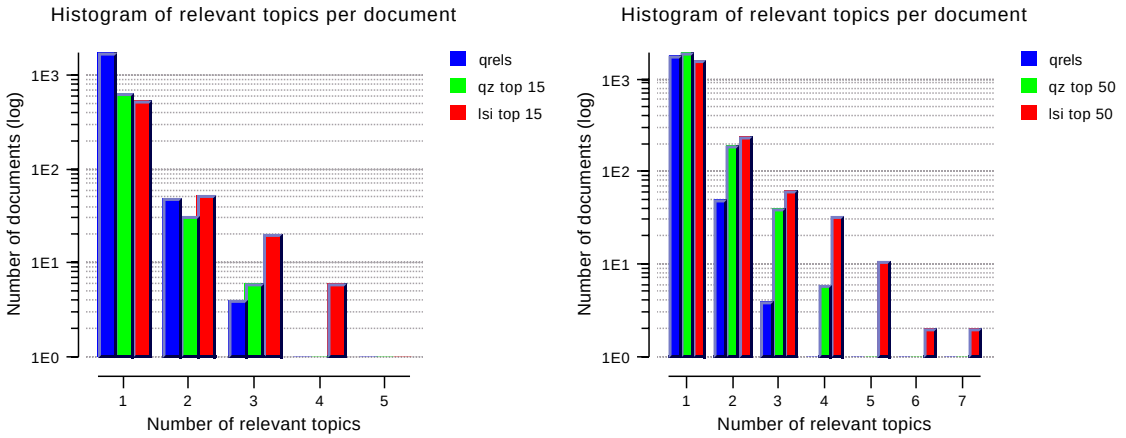

Figure 5: Histograms showing how many topics are relevant to each document, as dictated by the TREC-8 Filtering relevance judgments ("qrels"), and as predicted by the submitted runs. The horizontal axis is the number of relevant topics; the vertical axis is a log scale of the number of documents which are relevant to only that many topics. The chart on the left uses the top 15 submitted documents in each run; the right uses the top 50.

alternative approach might be to augment the matrix used to compute the LSI with more example profiles (perhaps from older TREC topics), or with a sample of documents.

In the following discussion, we will explore how documents are distributed among topics, and then how specific topics are related by the documents that are relevant or recommended to them. In particular, we have constructed graphs which illustrate the relationships between topics by linking topic nodes with edges indicating the number of documents recommended to both topics. This supports our hypothesis that documents are being recommended to related but irrelevant (by TREC standards) topics. These related documents degraded filtering precision, and this is in fact what we should expect collaboration to do in the TREC setting.

5.1 Document Overlap among Topics

Although Figure 2 implies families of topics that seem to be of related interest, the fact of the matter is that these are separate topics with specific guidelines as to what is and what is not relevant to the topic. Thus, one might expect that documents that are relevant to multiple topics, and thus collaboration, would be rare.

One measure of this is that a document might be recommended to more topics than are truly relevant. Figure 5 illustrates how many topics for which a document is relevant or recommended. It shows, how many topics were predicted by our runs for a document, in comparison to the official relevance judgments. The "qrels" bars show topics per document in the relevance judgments, while the `umrqz` and `umrlsi` bars show

the recommendations of the submitted runs. One can see that TREC defines relevance quite narrowly; the vast majority of documents are relevant to only one topic, and less than sixty documents are relevant to more than one topic. If a pure collaborative algorithm were used to predict relevance for these topics, and these relevance judgments were sampled for training data, it would fail miserably because the matrix would be too sparse. The probability of any useful quantity of overlap occurring is very small.

The two charts in Figure 5 differ in the method for predicting which documents in the `umrqz` and `umrlsi` runs are relevant. A routing run contains the highest-scored 1000 documents for each topic, but clearly the system does not expect that all 1000 documents are relevant. Thus, we only predict as relevant some of the documents in each run. The left-hand chart uses the top 15 ranked documents, because 15 is the median number of relevant documents per topic in the actual relevance judgments. The right-hand one uses the top 50.

We can see that our runs tend to spread documents across more topics than are actually relevant. Within the top 15, the `umrqz` run distribution is similar to the qrels, and the `umrlsi` run gives slightly more overlap. At 50 documents per topic the difference is much greater; however, for documents that are shared among only two or three topics, the runs are close to each other in overlap. Recommending documents to more topics than they are relevant for will obviously result in decreased precision for those topics. If the increases occur among related topics, this is an indication that the LSI technique is fostering collaboration by broadening the scope of the topics in a direction of related interest.

5.2 Visualizing Topic Clusters with Graphs

The histograms above group all the topics together, but we expect that the topics collaborate and share differently. Figure 6 shows how the topics share relevant documents, according to the relevance judgments. An edge between two topic nodes indicates a number of documents which are relevant to both topics. The style of line is related to the number of shared documents, as a visual aid; thicker lines indicate more documents. In this diagram, we can see the alternative fuels and pharmaceuticals clusters which we predicted from just reading the topics. These are also loosely linked to other topics, such as "ocean remote sensing", "robotics" and "obesity medical treatment". Another strong link exists between "territorial waters dispute" and "Falkland petroleum exploration", and this group also contains links to "piracy", "illegal technology transfer", and "World Court". Some of these topics are more closely tied together than others, for example, "mental illness drugs" and "R&D drug prices" with 15 documents, while the links between others are more tenuous.

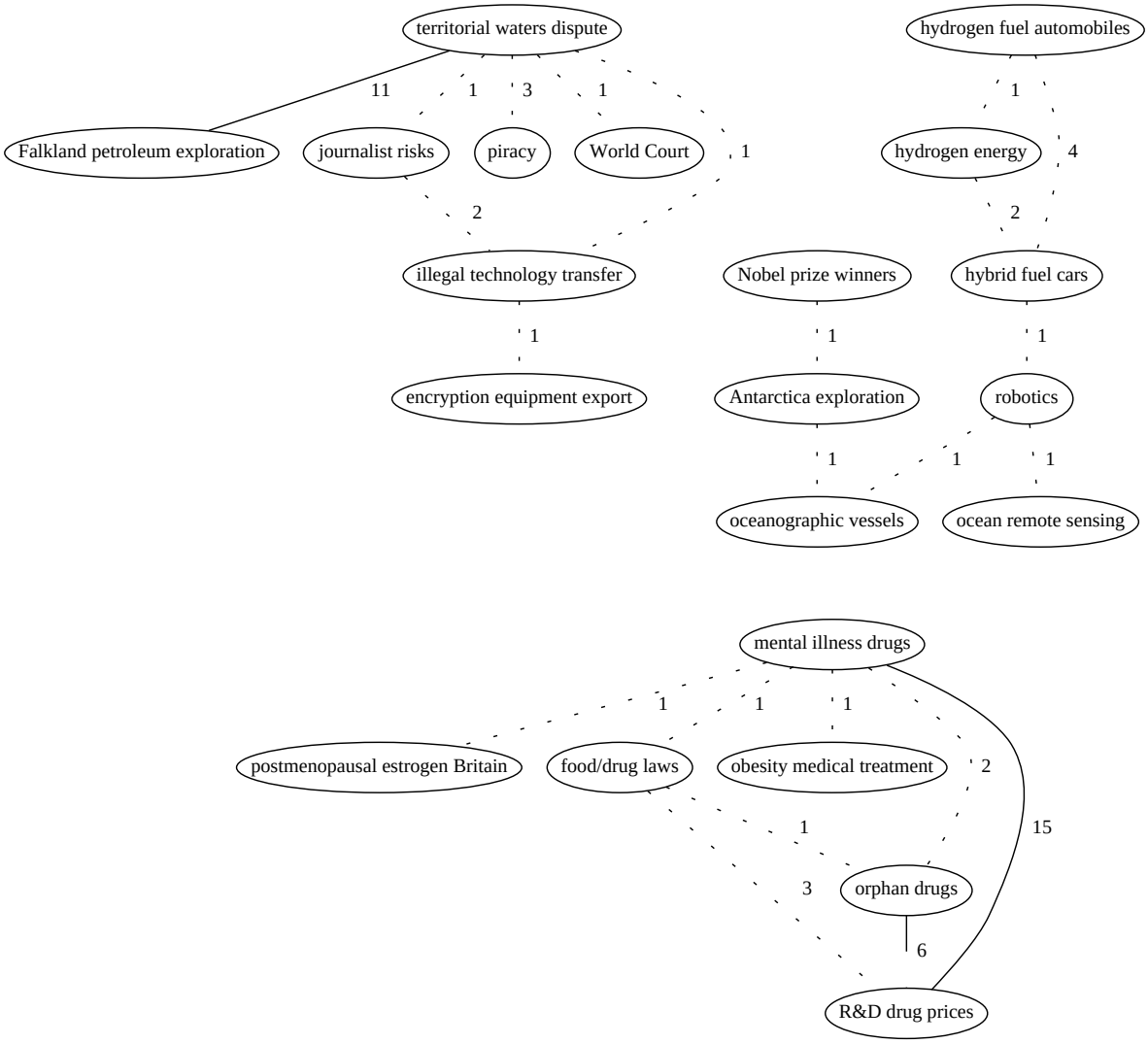

Figure 6: Number of relevant documents shared between topics, according to the official TREC judgments.

5.2.1 Topic Clusters from the Routing Profiles

Figure 7 shows the topic relationships as recommended by the query-zone (non-LSI) profiles. The documents represented by these links are in the top 50 for each topic in the submitted run. The graph of the entire recommendation set contains a large number of links with few documents; for clarity, only links of five or more documents are shown, which in Figure 7 is only 18% of the edge set. We can see that many of the links in the relevance judgments graph are predicted here, although with much larger shared documents sets.

Although not containing any relevant documents, our education cluster appears, with a link between "teaching disabled children" and "mainstreaming". It turns out that the documents along this link are of related interest (specifically, education studies and opinions on education policy) but are not actually topically relevant. The query-zone profiles also recommend some odd and probably unrelated links, for example the links among "cigar smoking", "health insurance holistic" and "clothing sweatshops". These links are likely due to co-occurrence among distracting terms in the profiles. Another example (not strong enough to show on this graph) is a predicted link between "transportation tunnel disasters" and "British Chunnel impact".

Finally, Figure 8 shows document sharing in the top 50 recommendations made by the LSI profiles. Again, this graph only shows links of five or more documents (in this case, 36% of the total edges). The LSI makes some links stronger, bringing them to our attention when they didn't appear in the graph for the query zone profiles. One example is the set of topics linked to the Falklands group; most of these links were not strong enough to be visible in Figure 7. Another example is in the hybrid fuel cars group; automobile recalls wasn't linked heavily before, but it is now. Also, note that the pharmaceuticals and medicine topics are more closely linked in the LSI recommendations.

The LSI also is lessening the impact of some relationships in the feedback profiles. The links between "hybrid fuel cars" and "hydrogen fuel automobiles" is slightly stronger while the links to both of these from "hydrogen energy" is slightly weaker. This effect is not as strong as we had hoped.

5.2.2 Topic Cluster Changes with LSI

Now that we have visualized the changes caused by using an LSI projection of the profiles, we will look at those changes more closely to try to understand how the LSI recommendations differ.

Figure 9 quantifies how the links change with LSI. Many more documents are added than deleted. More significantly, the vast majority of documents added and deleted from the links are irrelevant (but possibly related) to either topic. In fact, as the second half of this table shows, a majority of the documents added to the graph are on links that did not exist in the `umrqz` profiles, but emerged with LSI.

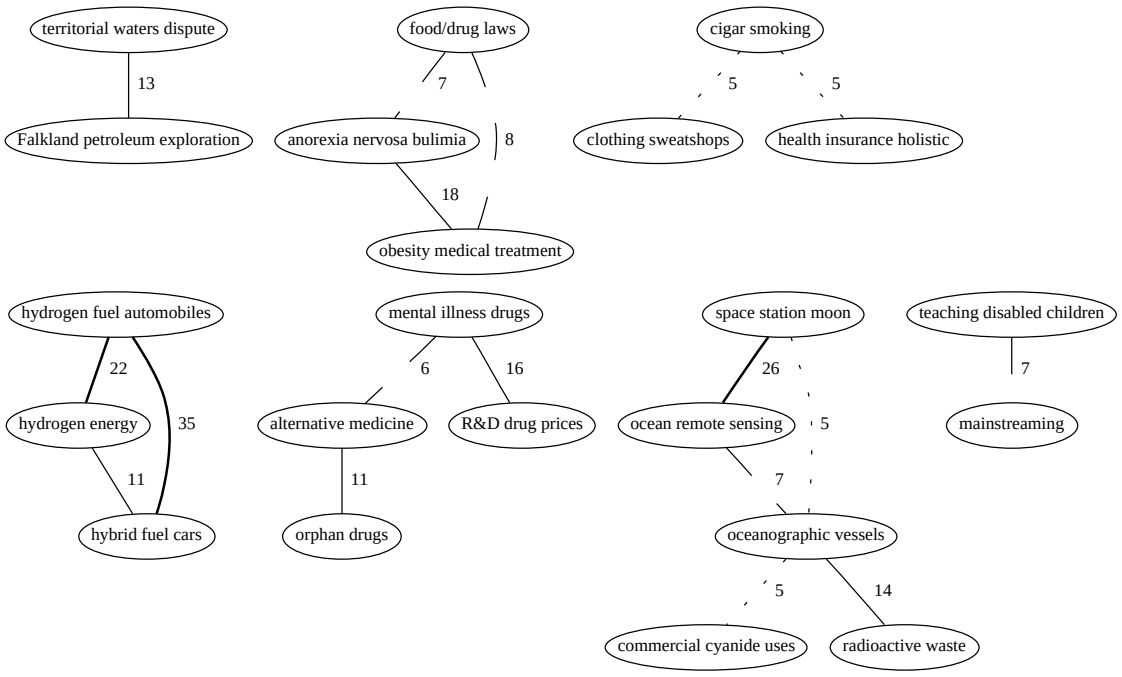

Figure 7: Document sharing among recommendations made by the query-zoned (non-LSI) profiles. Only links of five or more documents are shown.

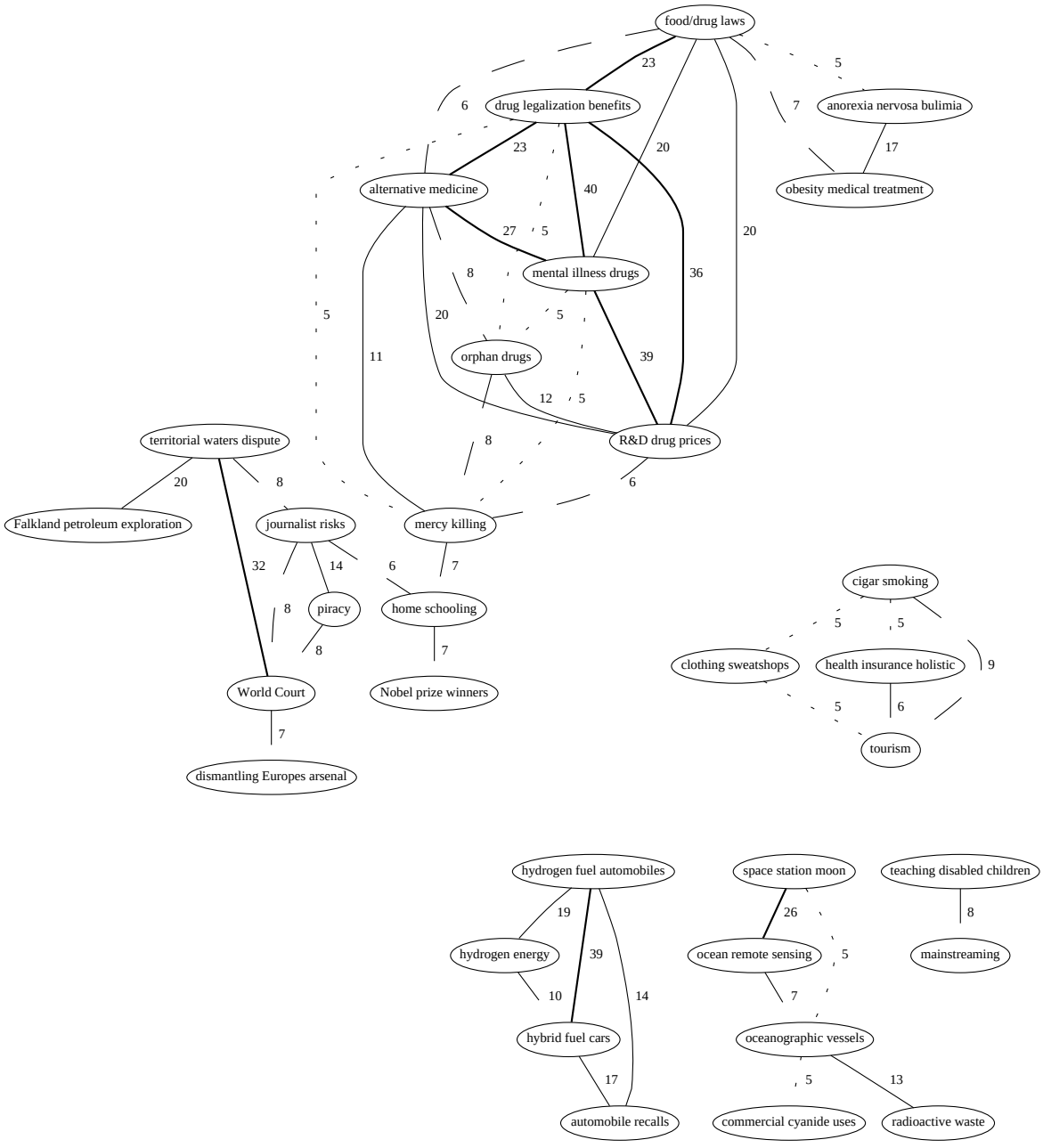

Figure 8: Document sharing among recommendations made by the LSI profiles. Only links of five documents or more are shown.

Relevant to...	added	deleted	added	deleted
one topic	180	9	73	7
both topics	7	0	6	0
neither	323	43	124	33
total	510	54	203	42

Figure 9: How many of the documents added to the intertopic links by LSI are actually relevant? This table shows how many that are added and deleted are relevant to one, both, or neither topic on the link. The first column pair includes completely new links added by LSI but not present before; the second pair only considers links present before LSI.

Upon examining the documents added and removed by LSI, some patterns emerge. For links which were new in the LSI recommendations, most documents are not relevant or even related to either topic. We suspect that these topic links are the result of patterns of term co-occurrence caused by the noisy nature of the query-zoned profiles. Examples of these kinds of topics links are those between several of the drug-related topics; the documents along the LSI links contain many common terms but the content is unrelated.

When a link existed in the recommendations from the original content-based profiles and was "revised" by LSI, the circumstances are different. If a document along the link is relevant to either topic, it is often related to the other. An example of this is the link between "territorial waters dispute" and "Falklands petroleum exploration": documents along this link concern disputes in the Falklands, mostly regarding fishing rights, and the related diplomatic sparring between the UK and Argentina. These documents are not about territorial waters disputes in general, nor are they about the particular disputes that arose from the discovery of petroleum around the Falklands, but about characteristic maritime disputes in the Falklands.

If the documents along the revised link were not relevant to either topic, then usually they were also not related. Likewise, documents removed from links were nearly always unrelated to either topic.

We found that topics with concise, specific short description fields tended to get linked to related topics better than those which were more broad. Also, many topics are difficult to form automated queries for using our methods; these topics usually have elaborate needs spelled out in the topic narrative field. For these topics, we believe that our methods form queries badly to begin with, and the LSI seems to make them worse. It is likely that our naive term selection in the profile expansion step, especially with respect to negative examples, is calling forth more distracting terms than we might find with a better approach.

6 Conclusion

We have conducted a TREC routing experiment using a collaborative algorithm to try to improve the performance for individual profiles by leveraging the collective information present in the entire group of

profiles. This idea is motivated by our observation of groups or clusters of information interests within the TREC topics; for example several topics deal with drug legalization, marine disputes, and the like.

A purely collaborative filtering approach, such as those used in the GroupLens project (Resnick et al., 1994; Konstan et al., 1997) would not work in the TREC setting, because very few relevant documents are shared in common among topics as a whole, much less within the TREC routing training data. Hence, our approach was to use a content-based collaborative algorithm based on latent semantic indexing. This algorithm realigns the collection of profiles according to patterns of term co-occurrence among the profiles, thus producing a collaborative space for routing text.

In practice, the collaborative LSI technique serves to recommend documents to pairs of topics within related subject areas, and thus to enhance ties among clusters of topics. But these ties consist of related, but mostly non-relevant documents, as can be seen by comparing the document co-recommendation graphs of the runs to the relevance judgments graph. While yielding poor performance by TREC measures, this is exactly what adding collaboration to a content approach should do: expand the user's definition of his information need. The usefulness of these ties cannot be assessed using the standard evaluation methodology; precision- and recall-based measures assume a static definition of relevance.

This problem has implications for future assessment of collaborative algorithms, because it implies that standard test collection methodology used in information retrieval may not be as useful for collaborative filtering. Test collections for collaborative filtering, such as EachMovie and MovieLens, typically consist of log data from live recommender systems, and most evaluations that use them focus on error rate (Breese et al., 1998). The problem with error rate in a log-based collection is that, like in TREC-style evaluations, it assumes that user preferences are completely stable, since in the experimental run documents will be "recommended" to users out of the order of the original presentation, and yet we compare the predictions to the original user ratings. This tends not to be troubling because the concept of "preference" for non-critical information like movies is less rigidly defined. TREC assumes this same stability but arrives at the assumption by a different path, by defining relevance narrowly and employing a methodology that examines the ranked list as the end product. The design of an experimental methodology providing the same repeatability and comparability as TREC, but which accounts for an evolving notion of relevance, is an open problem.

Acknowledgments

This work was supported by a research contract from the U.S. Department of Defense.

References

Ault T and Yang Y (2000). kNN at TREC-9: A failure analysis. In: Voorhees EM and Harman DK, eds., The Ninth Text REtrieval Conference. National Institute of Standards and Technology, Gaithersburg, MD. To appear.

Berry MW (1992). Large scale singular value computations. International Journal of Supercomputer Applications, 6(1):13–49.

Berry MW, Dumais ST and O'Brien GW (1995). Using linear algebra for intelligent information retrieval. SIAM Review, 37(4):573–595.

Billsus D and Pazzani MJ (1998). Learning collaborative information filters. In: Kautz H, ed., Proceedings from the AAAI 1998 Workshop on Recommender Systems. Madison, WI.

Breese JS, Heckerman D and Kadie C (1998). Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. Morgan Kaufman, Madison, WI.

Deerwester S, Dumais ST, Furnas GW, Landauer TK and Harshman R (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6):391–407.

Ding CHQ (1999). A similarity-based probability model for latent semantic indexing. In: Hearst et al. (1999), pp. 58–65.

Dumais ST (1994). Using LSI for information filtering: TREC-3 experiments. In: Harman DK, ed., Proceedings of the Third Text REtrieval Conference (TREC-3). Gaithersburg, MD. Also titled "Latent Semantic Indexing (LSI): TREC-3 Report".

Goldberg D, Nichols D, Oki BM and Terry D (1992). Using collaborative filtering to weave an information tapestry. Communications of the ACM, 35(12):61–70.

Golub GH and Van Loan CF (1996). Matrix Computations. The Johns Hopkins University Press, Baltimore, third edition.

Gupta D, DiGiovanni M, Narita H and Goldberg K (1999). Jester 2.0: A new linear time collaborative filtering algorithm applied to jokes. In: Soboroff I, Nicholas C and Pazzani M, eds., Proceedings of the 1999 SIGIR Workshop on Recommender Systems. Berkeley, CA.

Hearst M, Gey F and Tong R, eds. (1999). Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '99). ACM Press, Berkeley, California.

Hofmann T (1999). Probabilistic latent semantic indexing. In: Hearst et al. (1999), pp. 50–57.

Hull D (1994). Improving text retrieval for the routing problem using latent semantic indexing. In: Proceedings of the Seventeenth Annual International ACM SIGIR Conference (SIGIR '94). Dublin, Ireland.

Hull DA and Robertson S (1999). The trec-8 filtering track final report. In: Voorhees EM and Harman DK, eds., Proceedings of the Eighth Text REtrieval Conference (TREC-8), NIST Special Publication 500-246. National Institute of Standards and Technology, Gaithersburg, MD.

Kleinberg JM (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM, 46(5):604–632.

Kolda TG and O'Leary DP (1996). A semi-discrete matrix decomposition for latent semantic indexing in information retrieval. Technical Report UMCP-CSD CS-TR-3724, Department of Computer Science, University of Maryland, College Park and the UM Institute for Advanced Computer Studies.

Konstan JA, Miller BN, Maltz D, Herlocker JL, Gordon LR and Riedl J (1997). GroupLens: Applying collaborative filtering to Usenet news. Communications of the ACM, 40(3):77–87.

Malone TW, Grant KR, Turbak FA, Brobst SA and Cohen MD (1987). Intelligent information-sharing systems. Communications of the ACM, 30(5):390–402.

O'Day VL and Jeffries R (1993). Orienteering in an information landscape: How information seekers get from here to there. In: Proceedings of INTERCHI'93. Amsterdam, Netherlands.

Resnick P, Iacovou N, Suchak M, Bergstrom P and Riedl J (1994). GroupLens: An open architecture for collaborative filtering of netnews. In: Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work. ACM, Chapel Hill, NC.

Salton G, ed. (1971). The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall.

Schütze H, Hull DA and Pedersen JO (1995). A comparison of classifiers and document representations for the routing problem. In: Proceedings of the Eighteenth Annual International ACM SIGIR Conference (SIGIR '95). Seattle, WA, USA.

Shardanand U and Maes P (1995). Social information filtering: Algorithms for automating "word of mouth". In: Proceedings of CHI'95 – Human Factors in Computing Systems. Denver, CO, USA.

Singhal A (1997). AT&T at TREC-6. In: Voorhees EM and Harman DK, eds., The Sixth Text REtrieval Conference, NIST Special Publication 500-240. National Institute of Standards and Technology, Gaithersburg, MD.

Singhal A, Buckley C and Mitra M (1996). Pivoted document length normalization. In: Croft WB and van Rijsbergen CJ, eds., Proceedings of the Nineteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.

Singhal A, Choi J, Hindle D, Lewis DD and Pereira F (1998). AT&T at TREC-7. In: Voorhees EM and Harman DK, eds., The Seventh Text REtrieval Conference, NIST Special Publication 500-242. National Institute of Standards and Technology, Gaithersburg, MD.

Singhal A, Mitra M and Buckley C (1997). Learning routing queries in a query zone. In: Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 97). Philadelphia, PA.

Soboroff IM (2000). Combining Content-based and Collaborative Text Filtering. Ph.D. thesis, University of Maryland, Baltimore County, Baltimore, MD.

Soboroff IM and Nicholas CK (1999). Combining content and collaboration in text filtering. In: Joachims T, ed., Proceedings of the IJCAI'99 Workshop on Machine Learning in Information Filtering. Stockholm, Sweden.

Zha H and Simon HD (1999). On updating problems in latent semantic indexing. SIAM Journal on Scientific Computing, 21(2):782–791.