Combining evidence for effective information filtering

Susan T. Dumais
Bellcore
445 South St.
Morristown, NJ 07960
email: std@bellcore.com

Abstract

As part of NIST/ARPA's TREC Workshop, we used Latent Semantic Indexing (LSI) for filtering 336k incoming documents from diverse sources (newswires, patents technical abstracts) for 50 topics of interest. We developed representations of user interests, or filters, for these topics using two sources of training information. A *Word Filter* used just the words in the topic statements, and a *RelDocs Filter* used just the known relevant training documents and ignored the topic statement. Using the relevant training documents (a variant of relevance feedback) was more effective than using a detailed natural language description of interests. Combining these two vectors provided some additional improvements in filtering. On average, 7 of the top 10 documents and 44 of the top 100 documents are relevant using the combined vector method. Data combination using the results of the Word and RelDocs retrieval sets was not generally successful in improving performance compared to the best individual method, although we believe it might be if additional sources are used. These combination methods are quite general and applicable to a variety of filtering and feedback applications.

Introduction

As part of NIST/ARPA's TREC Workshop, we used Latent Semantic Indexing (LSI) for filtering 336k documents from diverse sources for 50 topics of interest. We examined how different sources of information (e.g., a natural language description of interests, feedback about previous documents) can be used to best predict which new objects will be of interest. An LSI model which combines the initial topic description with relevant training documents is quite effective.

Latent Semantic Indexing (LSI)

LSI is a variant of vector retrieval in which the dependencies between terms are explicitly taken into account (see Deerwester et al., 1990 for mathematical details and examples). Most retrieval models (e.g., Boolean, standard vector, probabilistic) treat words as if they are independent, although it is quite obvious that they are not (Salton & McGill, 1983). The central theme of LSI is that important inter-relationships among terms can be automatically derived, explicitly modeled, and used to improve retrieval.

LSI uses singular-value decomposition (SVD), a statistical technique closely related to eigenvector decomposition, to model the associative relationships. A large term-document matrix is decomposed into a set of k, typically 100 to 300, orthogonal factors. These derived indexing dimensions, rather than individual words, are the basis of the vector space used for retrieval. Each term and document is represented by a vector in the resulting k-dimensional LSI space. Terms that are used in similar contexts (documents) will have similar vectors in this space. One important consequence of the LSI analysis is that users' queries can retrieve documents that do not share any words with the query - e.g., a query about "automobiles" would also retrieve articles about "cars" and even articles about "drivers" to a lesser extent.

Retrieval operates in the same way in the reduced dimension vector space as it does in standard vector models. A query vector is located at the weighted vector sum of its constituent term vectors. Documents are ranked by their similarity to the query vector and the most similar documents are returned. Since both term and document vectors are represented in the same space, similarities between any combination of terms and documents can be easily obtained. This makes it easy to use LSI for relevance feedback and information filtering.

The LSI method has been applied to many of the standard information retrieval test collections with favorable results. Using the same tokenization and term weightings, the LSI method has equaled or outperformed standard vector methods in almost every case, and was as much as 30% better in some cases (Deerwester et al., 1990; Dumais, 1995). As with the standard vector method, differential term weighting and relevance feedback both improve LSI performance substantially (Dumais, 1991).

TREC - Information Filtering

We used LSI in NIST/ARPA's TREC-3 Workshop (Dumais, 1995; Harman, 1995). For the TREC *filtering (or routing) task*, we were given 50 topics of interest and asked to find articles relevant to these interests from a new stream of 336,306 documents (1.2 gig of ascii text). The 1000 documents most similar to each of the 50 topics of interest are returned, and performance is evaluated using precision and recall.

The TREC topic statements were quite detailed, structured and specific. They are representative of the profiles that an information analysts might develop over time for standing interests. An example filtering topic is given below.

<num> Number: 108
<dom> Domain: International Economics
<title> Topic: Japanese Protectionist Measures
<desc> Description: Document will report on Japanese policies or practices which help protect Japan's domestic market from foreign competition.
<narr> Narrative: A relevant document will identify a Japanese law or regulation, a governmental policy or administrative procedure, a corporate custom, or a business practice which discourages, or even prevents, entry into the Japanese market by foreign goods and services. A document which reports generally on market penetration difficulties but which does not identify a specific Japanese barrier to trade is NOT relevant.
<con> Concept(s):
1. Japan
2. Ministry of International Trade and Industry, MITI, Ministry of Foreign Affairs
3. protectionism, protect
4. tariff, subsidy, quota, dumping, obstruction, retaliation
5. structural impediment, product standard
6. trade dispute, barrier, tension, imbalance, practice
7. market access, free trade, liberalize, reciprocity
8. Super 301, 301 clause
<nat> Nationality: Japan

Training information was available for each topic. Known relevant and non-relevant documents from a different (although related) corpus of documents were identified. On average, there were 215 known relevant documents and 896 non-relevant documents for each topic. The topic statement and the training data were to be used to develop profiles or filters for each topic.

Basic Word and RelDocs filters

LSI has previously been applied to information filtering with promising results (Foltz & Dumais, 1992), and we extended this work to the large, diverse TREC corpus. We used the training corpus to construct a 346-dimensional LSI representation. For information filtering, we begin by identifying a vector for each topic of interest in the LSI space. New documents are "folded in" to the space, and suggested as relevant to a topic if they are near enough the filter vector. Folding in works just like query formation. Each new document is located at the weighted vector sum of its constituent term vectors.

We compared two basic methods for creating filters - one using only the text of the topic statement (***Word Filter***); the other using only relevant documents from the training set (***RelDocs Filter***). The ***Word Filter*** is located at the weighted vector sum of all the words in the topic statement. On average, topics contain 192 words, of which 52 are unique content words. This method ignores all the training data about relevant and non-relevant documents. The ***RelDocs Filter*** is located at the centroid or vector sum of the relevant training documents, and ignores the topic statement. This is a somewhat unusual variant of *relevance feedback*. Typically users' queries are modified by adding words from relevant documents and omitting words from non-relevant documents. We *replaced* the users' topic statement with relevant documents.

For each of the 50 filtering topics we created two filters or profiles - a Word Filter and a RelDocs Filter. Figure 1 illustrates how these vectors would be formed for one topic using a 2-dimensional LSI space. The vectors for the words in the topic statement are labeled w, and the vectors for the relevant training documents are labeled R. As can be seen in Figure 1, the Word Vector is located at the weighted sum of the vectors for topic words, and the RelDocs Vector is located at the sum of the relevant training document vectors. In both cases, the filter was a single vector. New documents were "folded in" to this LSI space, as described above, and ranked in decreasing order of similarity to the filter vector.

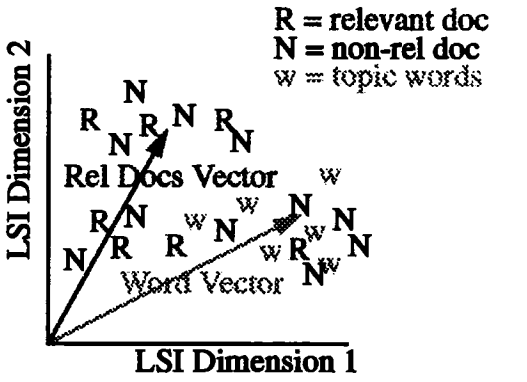

FIGURE 1. Baseline RelDocs and Word Vectors

These two methods provide baselines for examining various combinations of information from the users' topic statement and from feedback using relevant training documents.

Combining Information - Query Combination and Data Combination

The filtering methods described above constitute the main sources of information. We also examined methods for combining evidence from these sources. Belkin et al. (1993, 1994) have described the methods we used as *query combination* and *data combination (or data fusion).*

For ***query combination***, we combine two (or more) representations into a single new query vector for retrieval. In the work described in this paper we explored linear combinations of the basic Words and RelDocs vectors. The combined vector, with equal weight given to Words and RelDocs, is shown in Figure 2. A new ranking of documents was produced for the Combined vector.

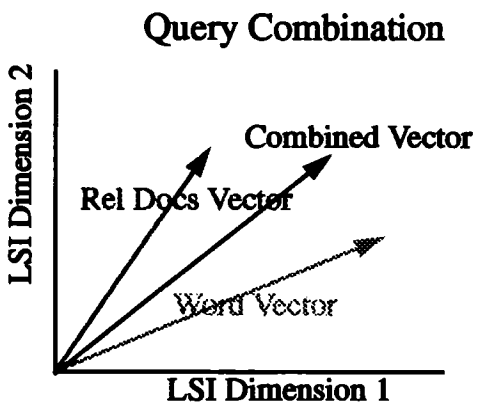

FIGURE 2. Query Combination

Data combination (or data fusion, as Belkin et al. call it) combines information about the results of two (or more) systems which rank the same documents in response to the same requests. We examined various ways of combining the top 1000 items from the Word and RelDocs matches to produced a new ranking of documents for each topic. We used information about the ranks and cosines, and combined them using max, min, and sum operators. Although we did not examine the parameter space as systematically as Bartell et al. (1994) did, we did choose data combination methods which had previously been successful in the context of TREC (Belkin et al., 1994; Fox & Shaw, 1994) .

Figure 3 shows geometrically what the combination looks like for the *min(cos(doc,Word), cos(doc,RelDocs))* measure. The cross-hatched region shows documents that have a high value on the *min(cos(doc,Word), cos(doc,RelDocs))* measure - these documents are near either the Word or RelDocs filter and would be top ranked for this data combination method.

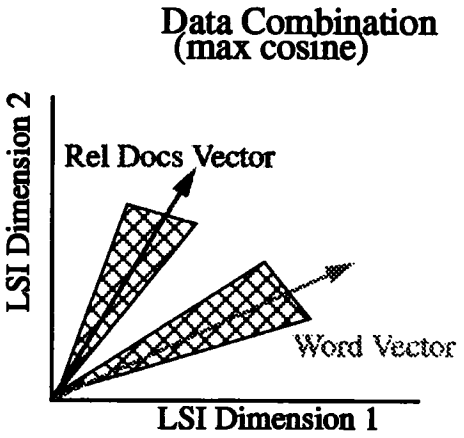

FIGURE 3. Data Combination - max cosine measure

In this paper we explored combinations of document rankings which were derived from the same LSI space and can thus be represented geometrically in the same figure. The method is much more general than this, however, and could be used to combine rank and/or similarity information from very different retrieval systems in order to determine a final document ranking. Data combination is thus much more general than query combination.

Results

For the TREC filtering tasks performance was evaluated using the top 1000 documents returned for each topic. We report the total number of relevant documents returned, precision at 10 and 100 documents, and average precision.

Query combination

Table 1 presents the results for the two basic LSI filters as well as several combinations of them. Not surprisingly, the RelDocs filter vectors which take advantage of the known relevant documents are better than the Word vectors. The improvement in average precision is 30% (.3737 vs. .2880). RelDocs was one of the best TREC filtering methods. Users get an average of 2 additional relevant documents in the top 10 using the RelDocs method for filtering (6.7 vs. 4.6). Even though the topic statements are quite rich, using known relevant training documents still provides sizeable retrieval benefits. Recall that the RelDocs method ignores the topic description! It is also important to note how much information was filtered out by these methods. By looking at just

100 documents per topic or 1.4% of the data (5000/336306), 24% of the known relevant documents are found. By looking at less than 15% of the data (50000/336306), 74% of the relevant documents are found using RelDocs.

TABLE 1. Query Combinations of LSI Word (W) and RelDocs (R) vectors

	Rel Docs	Prec at 10	Prec at 100	Avg Prec
Words (W)	6252	.4620	.3532	.2880
RelDocs (R)	6878	.6720	.4544	.3737
.5W+.5R	7078	.6820	.4400	.3792
.25W+.75R	7010	.6500	.4590	.3827
.75W+.25R	6930	.5540	.4118	.3561

We examined three mixtures varying the amount that the Word and RelDocs sources contribute to the combined vector (.25, .50, .75). Small improvements in performance are found when the RelDocs vector is given equal or more weight than the Word vector. This same pattern of results was observed for a different set of 50 routing topics in a previous TREC, so we have reason to believe these results will generalize to other filtering applications.

The RelDocs methods described above used all known relevant documents from the *training* collection. On average, there were 216 relevant documents for a topic. Table 2 shows how much training data it takes to achieve this level of performance. With only 5 relevant documents performance is poor. It is comparable to the Words filter in precision, and somewhat worse in the total number of relevant documents. (Knowing 5 relevant items is as good as a 192 word description of the topic of interest.) Performance improves steadily and with only 20 relevant documents is within 6% of the maximum. With relatively small amounts of training data, filtering performance is quite good.

TABLE 2. Performance as a function of number of training documents used in RelDocs vector

	Rel Docs	Prec at 10	Prec at 100	Ave Prec
5 Rel	5436	.5840	.2564	.2777
10 Rel	6313	.5860	.3934	.3096
20 Rel	6669	.6400	.4290	.3526
30 Rel	6642	.6560	.4428	.3565
50 Rel	6806	.6640	.4470	.3663
100 Rel	6822	.6800	.4514	.3706
max Rel	6878	.6720	.4544	.3737

The best performance one can achieve for this LSI representation is obtained by locating the filter vector at the centroid of all the relevant *test* document. Clearly this could not achieve this in practice since the relevant test documents are not known ahead of time, but it does set an upper bound on performance. Placing the filter vector here would increase average precision by 28% to .4776, so there is room for improvement given the current representation! We can move in this direction by combining the RelDocs vector with a vector based on new relevant test documents as they arrive. Combining the RelDocs vector with only 1 new relevant test documents improves average precision 7% to .4017, and adding 10 new relevant documents improves average precision 16% to .4353.

Data Combination

Data combination begins with the top 1000 documents returned for the Word and RelDocs filters and combines results (not the vectors) in various ways to arrive at a new ranking. We used information about the ranks and cosines, and combined them using max, min, and sum operators. Similar methods have previously been tested in the context of TREC although they do not provide the kind of systematic parameter analysis used by Bartell et al. (1994).

Consider, for example, the sum of cosines. For each document, a new measure of similarity (its cosine in the Word set plus its cosine in the RelDocs set) is computed and used to derive a new ranking. Retrieval performance is evaluated using the new ranking. Table 3 summarizes the performance of six data combination methods along with the Word and RelDocs vector baselines.

TABLE 3. Data Combinations of LSI Word (W) and RelDocs (R) return sets

	RelDocs	Avg Prec
Words	6252	.2880
RelDoc	6878	.3737
Ranks - sum(W,R)	6960	.3687
Ranks - min(W,R)	6938	.3498
Ranks - max(W,R)	6764	.3538
Cosines - sum(W,R)	6891	.3660
Cosines - min(W,R)	6764	.3229
Cosines - max(W,R)	6885	.3741

As can be seen in Table 3, there are only small inconsistent improvements in performance when different sources of data are combined. Differentially weighting the contributions of the Words and RelDocs measures does not help either.

Another way of combining data from the two basic filters is to pick the best filter for each topic. Although the RelDocs filter is best on average, there are 14 topics for which the Words filter is better. If we use the training data to select which method to use for each topic, performance is 6890 relevant docs and .3731 average precision, and this is no better than using just the RelDocs filter. (If we select optimally using the test data, small improvements are seen - 7058 relevant docs and .3962 average precision.)

Others (Belkin et al., 1994; Fox & Shaw, 1994) have reported some success with data fusion methods, and it is not entirely clear why we were not successful. Perhaps our two sources are not sufficiently different from each other. Both the Word and RelDocs vectors were represented in the same LSI space and are likely to reflect many of the same derived indexing features. (However, remember that a query combination method does improve performance.)

A related possibility is that two sources of information may not be sufficient. Additional sources of information are easy to include (e.g., dot product similarity, keyword matching, phrase matching), but it would be nice to do so in a principled manner. Finally, it may be that non-linear combinations would lead to performance improvements.

Summary and Conclusions

We used LSI for filtering 336k documents from diverse sources for 50 interest profiles. Using relevant training documents was more effective than using a detailed natural language description of interests. Combining these two vectors provided some additional improvements in filtering. On average, 7 of the top 10 documents are relevant using this method. Performance can be improved by continually incorporating new relevant documents. Data combination using these two methods was not successful, although we believe it might be if additional sources of information are used. These combination methods are quite general and applicable for a wide variety of filtering tasks.

References

Bartell, B.T., Cottrell, G.W., and Belew, R.K. Automatic combination of multiple ranked retrieval systems. In *Proceedings of SIGIR94*, ACM Press, 1994.

Belkin, N. J., Kantor, P., Cool, C. and Quatrain, R. Combining evidence for information retrieval. In D. Harman (Ed.), *The Second Text REtreival Conference (TREC-2), NIST Special Publication 500-215*, 1994, pp. 35-44.

Belkin, N. J., Cool, C., Croft, W. B. and Callan, J. P. The effect of multiple query representations on information retrieval performance. In *Proceedings SIGIR'93*, 1993, pp. 339-346.

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

Dumais, S.T. Using LSI for information filtering: TREC-3 experiments. In: D. Harman (Ed.), *The Third Text REtrieval Conference (TREC3), National Institute of Standards and Technology Special Publication, 500-225*, 1995, pp.219-230.

Foltz, P. W. and Dumais, S. T. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 192, *35(12)*, 51-60.

Fox, E. A. and Shaw, J. A. Combination of multiple searches. In D. Harman (Ed.), *The Second Text REtreival Conference (TREC-2), NIST Special Publication 500-215*, 1994, pp. 243-252.

D. Harman (Ed.), *The Third Text REtrieval Conference (TREC3), National Institute of Standards and Technology Special Publication, 500-225*, 1995, pp.219-230.

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