Improvement of Vector Space Information Retrieval Model based on Supervised Learning

Xiaoying Tai*1 Minoru Sasaki*1 Yasuhito Tanaka*2 Kenji Kita*1

*1 Faculty of Engineering, Tokushima University
2-1, Minami-josanjima, Tokushima 770-8506, Japan
*2 Department of Economics & Information Science, Hyogo University
2301 Shinzaike Hiraoka-cho Kakogawa, Hyogo 675-01, Japan
{xytai, kita, sasaki}@is.tokushima-u.ac.jp
vyasuhito@humans-kc.hyogo-dai.ac.jp

Abstract

This paper proposes a method to improve retrieval performance of the vector space model (VSM) by utilizing user-supplied information of those documents that are relevant to the query in question. In addition to the user's relevance feedback information, incorporated into the retrieval model, which is built by using a sequence of linear transformations, is information such as inter-document similarity values. Then, the high-dimensional and sparse vectors are reduced by SVD (Singular Value Decomposition) and transformed into the low-dimensional vector space, namely the space representing the latent semantic meanings of the words. The method was experimented on through two test collections, Medline collection and Cranfield collection. Improvement of average precision compared with LSI (Latent Semantic Indexing) model were 4.03% (Medline) and 24.87% (Cranfield) for the two training data sets, and 0.01% (Medline) and 4.89% (Cranfield) for the test data, respectively. The proposed method provides an approach that makes it possible to preserve the user-supplied relevance information for a long term in the system and to use the information later.

Keywords: supervised learning; vector space model; relevance feedback; singular value decomposition; linear transformation

1 Introduction

The Vector Space Model (VSM) is a conventional information retrieval model that represents documents and queries by vectors in a multidimensional space. The basic idea is that when indexing terms are extracted from a document collection, each document or query is represented as a vector of weighted term frequencies. Similarity comparisons among documents and/or between documents and queries are made via the similarity between two vectors (e.g. cosine similarity).

The technique to improve the VSM's retrieval performance using feedback information from the user is called relevance feedback. Relevance feedback identifies and utilizes, among documents from a retrieved set, those that are most relevant to the original query to redefine/clarify the query by adjusting models, documents or retrieval queries.

Generally, use of relevance feedback information by means of modifying a single query to improve a system's retrieval is becoming a common feature of many IR systems. However, as relevance feedback works only to adjust retrieval queries, information retrieved in a relevance feedback process can hardly be preserved for a long term in the system and be used later.

Another technique of using the user's feedback information is User Lens[2, 6]. By applying a criterion based on Guttman's Point Alienation statistic, the User Lens is used to reweight the vectors which represent documents and/or queries in information retrieval systems, so that the vectors get automatically adjusted according to the user's relevance feedback. As a result, relevance feedback information from the user can be preserved for a long term in the system.

To improve retrieval performance based on the VSM, we propose a new method that utilizes information of documents that are relevant to the query in question. In addition to the user's feedback information, information such as inter-document similarity values are incorporated into the retrieval model built by using a sequence of linear transformations. Similar to the technique User Lens, the proposed method preserves feedback information for a long term in the system. But it differs from the User Lens in the way it adjusts the model by using a sequence of linear transformations.

2 The Retrieval Model

Proceedings of the 5th International Wordshop Information Retrieval with Asian Languages
Copyright ACM 1-58113-300-6/00/009 ...$5.00

In this section, specification of the vector space model (VSM) is briefly explained, and based on VSM a new method is proposed to improve the precision of the information retrieval system.

2.1 Vector Space Model

Within the VSM, each document $d_j (1 \leq j \leq n)$ is represented by a weighted vector,

$$d_j = (w_{1j}, ...; w_{tj})^T \tag{1}$$

where w_{kj} is the weight (or importance) of term k in the document d_j, and t is the size of the indexing term set. A collection of n documents is then represented by a $t \times n$ term-document matrix D:

$$D = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{t1} & w_{t2} & \dots & w_{tn} \end{bmatrix}$$

For the queries, a similar matrix representation is applied:

$$Q = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1l} \\ q_{21} & q_{22} & \dots & q_{2l} \\ \vdots & \vdots & \ddots & \vdots \\ q_{t1} & q_{t2} & \dots & q_{tl} \end{bmatrix}$$

where l is the size of the query vector set. When we are given a query q_i:

$$q_i = (q_{1i}, ..., q_{ti})^T \ (1 \leq i \leq l) \tag{2}$$

retrieval is achieved by measuring the similarity between a document and a query in the underlying vector space:

$$\text{sim}(d_j, q_i) = \sum_{k=1}^{t} (w_{kj} \times q_{ki}). \tag{3}$$

In addition, when we are given a collection of documents D, and queries Q, the resulting similarity matrix is given by:

$$S = D^T Q. \tag{4}$$

2.2 Building of a Retrieval Model based on Linear Transformations

Here, we consider a supervised approach. Given the query matrix Q and the document matrix D, we are told which documents are relevant for each query. We use this information to construct a relevant matrix $A = a_{ij}$, where a_{ij} equals 1 if document j is relevant to query i, and 0 otherwise. We assume that the query matrix Q and the relevance matrix A will be related by a matrix X. For the finding matrix X, we need the following linear transformation L:

$$A = L[Q] = D^T X Q. \tag{5}$$

That is, we let the matrix Q and A be related by the L. For this reason, first we introduce linear transformation g to transform the query matrix Q to a matrix M:

$$M = g[Q] = XQ. \tag{6}$$

Then, in order to make it related to A, we also introduce a linear transformation f, so L is:

$$\begin{aligned} A &= L[Q] \\ &= f \circ g[Q] \\ &= f[g[Q]] \\ &= f[M] \\ &= D^T M \\ &= D^T X Q. \end{aligned} \tag{7}$$

The formula can be illustrated in Figure . As we show now, the bases of the query vector space can be translated to the bases of relevance documents vector space by L.

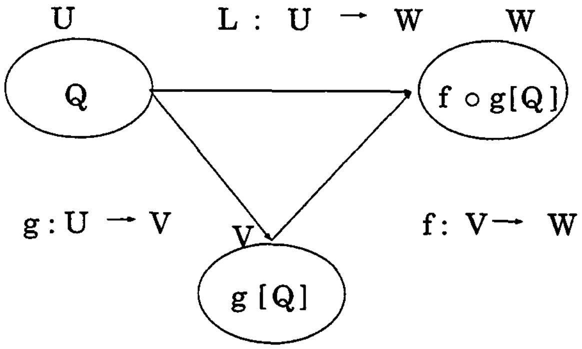

Figure 1: *Linear transformation on the retrieval model*

Let U, V, and W be finite dimensional vector spaces with bases $(e_1, ..., e_t)$, $(e'_1, ..., e'_t)$, and $(e''_1, ..., e''_n)$, respectively. Let g and f be linear transformations from U to V and from V to W respectively, so that transformation L of U into W is defined by $L = f \circ g$. The X is called the matrix of g with respect to the bases $(e_1, ..., e_t)$, $(e'_1, ..., e'_t)$, and D is called the matrix of f with $(e'_1, ..., e'_t)$ and $(e''_1, ..., e''_n)$.

With respect to these bases, given Q, M, A are uniquely determined by its coefficients. Therefore each query vector q within query vector space U is transformed to representing relevance documents vector space W which are relevance documents to questions, from given query vector space U, by $L = f \circ g$ linear transform . The following formula shows the transformational process:

$$L[Q] \quad = \quad f \circ g[Q]$$

$$\begin{aligned}
&= f \circ g\left[\sum_{k=1}^{t} Q_k e_k\right] \\
&= f\left[\sum_{k=1}^{t} Q_k \Big(g(e_k)\Big)\right] \\
&= f\left[\sum_{j=1}^{t} \Big(\sum_{k=1}^{t} x_{jk} Q_k\Big) e_j^{'}\right] \\
&= f\left[\sum_{j=1}^{t} M_j e_j^{'}\right] \\
&= \sum_{j=1}^{t} M_j \Big(f(e_j^{'})\Big) \\
&= \sum_{i=1}^{n} \Big(\sum_{j=1}^{t} d_{ik} M_j\Big) e_i^{''} \\
&= \sum_{i=1}^{n} A_i e_i^{''}. \qquad (8)
\end{aligned}$$

2.3 Finding Matrix X

We find X in two steps. First, we find M satisfying the following equation:

$$A = D^T M. \qquad (9)$$

Second, we find X satisfying equation 6. However, in general, the solution of the equation 9 does not exist. Therefore, M is to be determined by solving the following least squares problem:

$$M^* = \underset{M}{\operatorname{argmin}} \|A - D^T M\|_F^2 \qquad (10)$$

where $\|\cdot\|_F$ indicates the Frobenius norm. We can use M^* as the best approximation to M.

Computationally, we use the QR-decomposition of the matrix D, that is, $D^T =$ QR. Then, we obtain M^* as follows:

$$\begin{aligned} y &= \mathrm{Q}^T A \\ M^* &= \mathrm{R}^{-1} y \qquad (11) \end{aligned}$$

After computing the M^*, we find X^* by solving the following least squares problem:

$$X^* = \underset{X}{\operatorname{argmin}} \|M^* - XQ\|_F^2 \qquad (12)$$

2.4 Introducing Inter-document Similarity Values

After we have found X in the equation 5, a trial test is carried out to examine the performance of the model:

$$S = D^T X Q \qquad (13)$$

on the Medline collection.

The model is first trained with the training data. Then, the effictiveness of the model is tested through both the training data and the test data. As a result, the performance on the training data set improves and it produces 100% precision, but the performance on the test data set degrades.

In order to improve the precision of the test data set, we try to incorporate inter-document similarity values into the model in addition to the relevance documents information. Here, the inter-document similarity values can be represented by using self-correlation matrix $D^T D$. To incorporate the matrix $D^T D$ into our model, we use the arrangement operation. We suppose that we arrange orderly the elements of matrixes X and Y into the matrix Z, and the Z is called the arrangement of X and Y.

$$Z = X \circ Y \qquad (14)$$

For example, we let X and Y are:

$$X = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$Y = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

Then, Z is

$$Z = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

To incorporate $D^T D$ into the equation 5, we try to extend 4 as follows:

$$\begin{aligned} S \circ (D^T D) &= (D^T Q) \circ (D^T D) \\ &= D^T (Q \circ D) \qquad (15) \end{aligned}$$

Consequently, the equation 5 is extended as follows, with introduction of the matrix X.

$$A \circ (D^T D) = D^T X (Q \circ D) \qquad (16)$$

2.5 Dimensionality Reduction by SVD

In the VSM, the elements of the term-document matrix are occurrences of each word in a particular document. Since every word does not normally appear in each document, the matrix D is usually high-dimensional and sparse. High-dimensional and sparse vectors are susceptible to noise, and have difficulty capturing the underlying semantic structure. Additionally, the storage and processing of such data places great demands on computing resources. The dimensionality reduction of the model is a method to solve these problems. Using the singular value decomposition (SVD)[3], one can take advantage of the implicit higher-order structure in the association of terms with documents by 100-300 of the largest singular vectors, then match them against user queries.

The vectors representing the documents are projected in a new, low-dimensional space obtained by SVD. The D representing the term-document matrix is expressed as follows by using SVD.

$$D = U_{(t,r)}\Sigma_{(r,r)}V^T_{(r,n)} \quad (r = \text{rank}(D)) \qquad (17)$$

where U and V are orthonormal matrices ($U^TU = V^TV = I$) and Σ is a diagonal matrix in which the singular values of D are listed in descending order. SVD works by omitting all but the $k(k < r)$ largest singular values in the above decomposition for some appropriate k. Here k is the dimension of the low-dimensional semantic space alluded to in the informal description above. It should be small enough to enable fast retrieval, and large enough to adequately capture the structure of the collection. The reduced-dimension representation is then given by the rank-k approximation:

$$D_k = U_k\Sigma_k V_k^T \quad (k < rank(D)) \qquad (18)$$

Here, we let $k = 100$. The D_k, in one sense, captures most of the important underlying structure, yet at the same time removes the noise. This results not only in great savings in storage and query time, but also in improvements of information retrieval.

At last, we find the X from the equation:

$$A \circ (D_k^T D_k) = D_k^T X_k(Q_k \circ D_k) \qquad (19)$$

and get the following model:

$$S = D_k^T X_k(Q_k \circ D_k). \qquad (20)$$

3 Experiments

3.1 Document Collections

Experiments for the evaluation using two standard collections are conducted to validate the model 20. The two collections used are the Medline collection of medical abstracts, and the Cranfield collection of aeronautics abstracts. Included with each of the test collections is a set of evaluation queries. The performance of an arbitrary retrieval method can be estimated by using these queries, since all documents in the collection have been identified as either relevant or irrelevant to each query. In our experiments, the queries and the relevance documents are changed into the query matrix Q and the relevance documents matrix A, respectively. Then, Q and A are divided into two disjoint sets, the training data set and the test data set. A training data set is used to construct a retrieval model 20, and the test data set is used to evaluate the resulting model. The query set and the relevance document set are partitioned so that 2/3 (Medline) and 3/4 (Cranfield) of each collection are placed in the training data set, and the remaining 1/3 (Medline) and 1/4 (Cranfield) are placed in the test data set, respectively. In this way, the Medline collection has 30 sample queries; thus each partition has 20 training queries and 10 test queries, and the cranfield has 225 queries, partitioned into 169 training queries and 56 test queries.

3.2 Extracting Indexing Terms

Numbers, marks and terms which only occur in one document are deleted from each collection. We also eliminate non-content-bearing stopwords such as "a", "able" etc, using a stop list of 439 common English words. Then, remaining terms are stemmed using the Porter algorithm[5]. The results of extraction for indexing term with the preprocessing step are summarized in Table 1.

Table 1: *Size of the two test collections*

	Medline	Cranfield
number of documents n	1033	1400
number of queries l	30	225
number of indexing terms m	4329	4991

3.3 Term Weighting

There are two different types of term weighting: Global and Local. Local weights are functioned to determine how many times each term appears in a document; global weights are functioned to determine how many times each term appears in the entire collection. The d_{ij}, i-th element of the document vectors d_j is given by

$$d_{ij} = L_{ij}G_i, \qquad (21)$$

where L_{ij} is the weight for term i in the document d_j, G_j is the global weight for term i. As a term weighting scheme, we used Log-Entropy, is given by[4]:

$$L_{ij} = \begin{cases} 1 + \log f_{ij} & (f_{ij} > 0) \\ 0 & (f_{ij} = 0) \end{cases} \qquad (22)$$

$$G_i = 1 + \sum_{j=1}^{n} \frac{\frac{f_{ij}}{F_i}\log\frac{f_{ij}}{F_i}}{\log n} \qquad (23)$$

where n is the number of documents in the collection, f_{ij} is the frequency of the i-th term in the j-th document, and F_i is the frequency of the i-th term throughout the entire document collection.

3.4 Training Model

In the section 2.2, we let the elements of the relevance documents matrix A be 0 or 1. In the evaluation experiment, the elements of the relevance documents matrix A are multiplied by the weight w_R. Mathematically, this is:

$$A = w_R \times A. \qquad (24)$$

Table 2: *Average precision according to the change of the w_R of the relevance documents matrix A on Cranfield training data (127 and 42)*

	Average Precision							
Training query data	w_R=3.0	w_R=4.0	w_R=5.0	w_R=6.0	w_R=7.0	w_R=8.0	w_R=9.0	w_R=10.0
127	0.4503	0.4617	0.4732	0.4843	0.4941	0.5027	0.5014	0.5168
42	0.3657	0.3687	0.3687	0.3687	0.3718	0.3771	**0.3779**	0.3759

Table 3: *Average precision of LSI model vs. proposed model*

		Medline		Cranfield	
		Weight w_R	Average Precision	Weight w_R	Average Precision
LSI Model	Training Data	-	0.6747	-	0.4001
	Testing Data	-	0.6927	-	0.4434
Proposed Model	Training Data	1.0	0.7019	9.0	0.4996
	Testing Data	1.0	0.6928	9.0	0.4651

Because relevance to the training data set is risen generally according to weight w_R increase, retrieval precision to the training data set is risen too. However, when making undue excessive improve the precision of the training data set simply, (making weight w_R increase simply) the retrieval precision to the evaluation data (the strange data set) results in having fallen oppositely. To decide the optimal weight w_R, the heldout method can be used. That is, we divide training data set into two, using the one as the known data set, and the other as the strange data set. Then weight w_R, which can raise the most retrieval precision to the strange data set, will be decided. In this experiment, because there are a few queries for Medline collection, we have made weight w_R = 1.0, but we find the optimal weight w_R for Cranfield collection , using the heldout method. As a result, it is shown by Table 2, the optimal weight to Cranfield collection became w_R = 9.0.

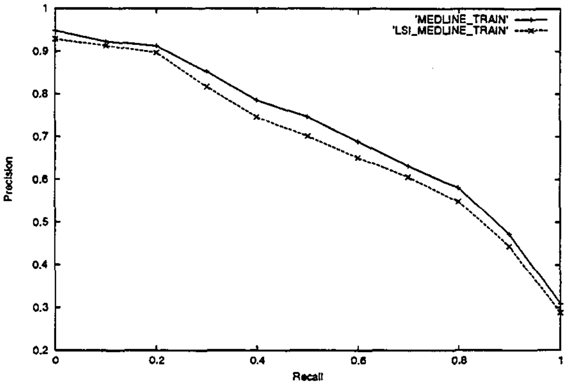

Figure 2: *Recall-Precision performance curve on the Medline training data*

3.5 Experiment Results

The Table 3 shows the average precision of the retrieval model, a model based on the LSI (Latent Semantic Indexing)[1], and the proposed model. The rises of retrieval precision compared with LSI model are 4.03% (Medline) and 24.87% (Cranfield) for two training data, and 0.01% (Medline) and 4.89% (Cranfield) for two test data, respectively.

Figure 2 to Figure 5 show the Recall-Precision performance curve to the training data set and the test data set of two collections. The solid line is the proposed model, and the broken line is the LSI model. For the training data set, the retrieval effictiveness of the proposed model is improved compared with that of the LSI model. This is because the proposed model against each question in the training set is learning the information which are relevant documents. But also demonstrated is that the improvement of the precision to the test data set of Medline collection is very little. For the training data set or the test set of Cranfield collection, the improvement of average precision is gotten. This seems to be resulted from the size difference of the training data set. The scale of Medline collection is small and we only used retrieval queries 20 as the training data set. In the case of Cranfield collection, we used 169 retrieval questions as the training data set. It may be taken into consideration that if we use a larger training data set, the retrieval model improvement could be greater.

4 Conclusions

The method proposed in this paper to improve

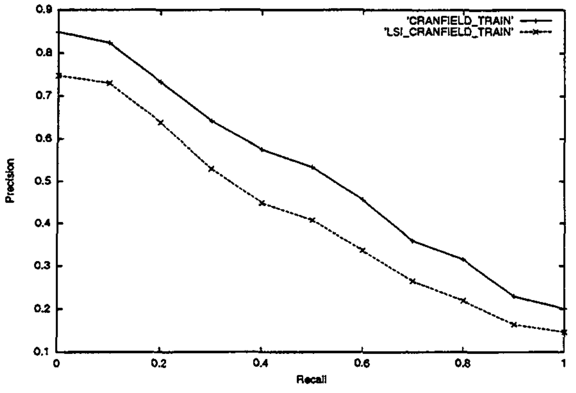

Figure 3: *Recall-Precision performance curve on the Cranfield training data*

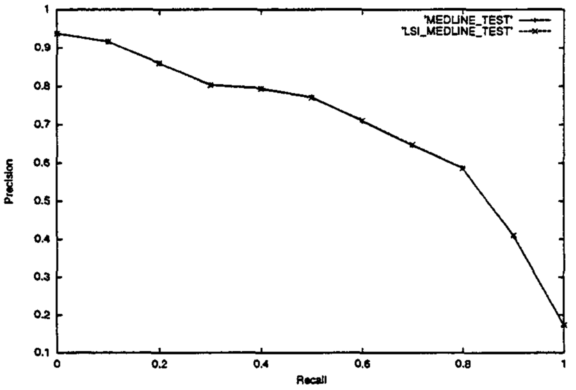

Figure 4: *Recall-Precision performance curve on the Medline testing data*

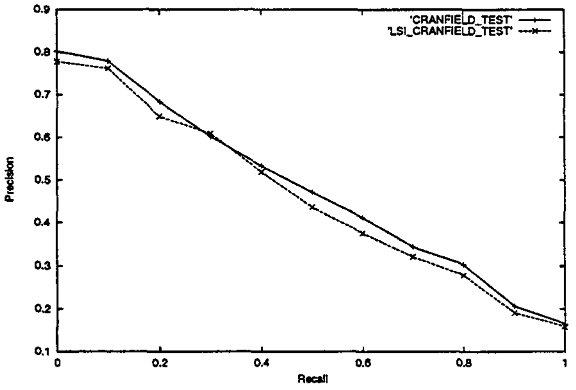

Figure 5: *Recall-Precision performance curve on the Cranfield testing data*

retrieval performance is based on the VSM (Vector Space Model) by utilizing user-supplied information of those documents that are relevance documents to the query in question. In addition to feedback information from the user, information such as inter-document similarity values are incorporated into the retrieval model, which is built by using a sequence of linear transformations. The model was experimented on through two test collection (Medline and Cranfield) and the effectiveness of the proposed method is shown.

In usual relevance feedback technique, it isn't possible to preserve the feedback information from the user in system for a long term to adjust retrieval model against each retrieval query.

The new method introduced in this paper has the advantage over the previous ones. It provides an approach that makes it possible to preserve information having been incorporated into the retrieval model for a long term in the system and to use them later.

Reference

1 Berry, M. W., Dumais, S. T., O'Brien, G. W. Using linear algebra for intelligent information retrieval.*SIAM Review, 37(4)*,1994, pp. 573-595.

2 Bartell, B. T., Cottrell, G. W. and Belew, R. K. Optimizing parameters in a ranked retrieval system using multi-query relevance feedback. *Proceedings of the Symposium on Document Analysis and Information Retrieval,* Las Vegas, 1994.

3 Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K. and Harshman, R. Indexing by latent semantic analysis.*Journal of the American Society for Information Science,* Vol. 41, No. 6, 1990, pp. 391-407.

4 Erica, C. and Tamara, G. K. New term weighting formulas for the vector space method in information retrieval.*Technical Memorandum ORNL-13756, Oak Ridge National Laboratory, Oak Ridge, Tennessee*,1998.

5 Frakes, W. B. and Baeza-Yates, R. Information retrieval: Data structures and algorithms.: Prentice Hall, 1992.

6 Vogt, C. C., Cottrell, G. W., Belew, R. K. and Bartell, B. T..User lenses-achieving 100% precision on frequently asked questions. *Proceedings of User Modeling'99,* Banff,1999.