Experiments on Routing, Filtering and Chinese Text Retrieval in TREC-5

Chong-Wah Ngo [1] Kok F. Lai [2]

Information Technology Institute
11 Science Park Road, Singapore 117685.

Abstract

We describes our experiments in the routing, filtering and Chinese text retrieval. We based our routing and filtering experiments on our discriminant project algorithm. The algorithm sequentially constructs a series of orthogonal axis from the training documents using the Gram-Schmidt procedure. It then rotates the resulting subspace using principal component analysis so that the axis are ordered by their importance. For Chinese text retrieval, we experimented both with an automatic method and a manual method. For the automatic method, we use all phrases in the description field and compute the aggregate scores using the simple *tf.idf* formula. We then manually construct boolean phrase queries which are thought to improve the results.

1 Introduction

This paper describes the routing, filtering and Chinese text retrieval experiments performed by the Information Technology Institute for TREC-5.

We based our routing and filtering experiments on our *discriminant project algorithm* as described in TREC-4 [1]. The algorithm sequentially constructs a series of orthogonal axis from the training documents using the Gram-Schmidt procedure. It then rotates the resulting subspace using principal component analysis so that the axis are ordered by their importance. One can then discard insignificant asis to obtain compact and efficient representation of relevance information.

For the routing task, we experimented with a two-phase training algorithm. In the first phase, we used the centroid of all relevant documents to produce an initial ranking for all training documents. We then select the top X % of non-relevant documents from the initial ranking, and then supply them as negative examples into the discriminant projection algorithm. We constructed the two routing queries, *itidp*1 and *itidp*2, using $X = 10$ % and $X = 30$ % respectively.

For the filtering task, we used *itidp*1 as the filtering query and attempted to find the best threshold to maximise the utility measures. The training documents are first divided equally into two sets. We first construct the routing queries from the first set using the discriminant projection algorithm. We then use these queries to obtain the scores for all documents in the second set. For each query, we find the respective thresholds that best optimise the utility measures. We then apply these thresholds,

[1] Currently with Computer Science, Hong Kong University of Science and Technology

[2] Please direct all enquiries and correspondence to Kok F. Lai, Information Technology Institute, 11 Science Park Road, Singapore 117685. Email address : kflai@iti.gov.sg.

together with routing query constructed from the entire training set, to filter new documents in the test set.

For Chinese text retrieval, we experimented both with an automatic method and a manual method. For the automatic method, we use all *phrases* in the description field and compute the aggregate scores using the simple $tf.idf$ formula. We then examine the results and manually construct boolean phrase queries which are thought to improve the results.

The following sections explain in more details the experiments which were performed.

2 The Discriminant Projection Algorithm

Denote a document as follows :

$$\mathbf{d} = [f_1, f_2, \ldots f_n]^T \tag{1}$$

where f_j is a function of term frequency for term j. n is the total number of unique indexed terms, and is typically very large. We shall assume that $\mathbf{d}$ is normalized, *i.e.*, $|\mathbf{d}| = 1$, in subsequent discussions.

Denote the collection of training documents as $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m]$. For a typical routing task, $\mathbf{D}$ consists of two separate groups : $\mathbf{D}_r$ and $\mathbf{D}_x$. $\mathbf{D}_r$ contains the set of m_r relevant documents while $\mathbf{D}_x$ contains the set of m_x non-relevant documents.

Note that the total number of training documents, $m = m_r + m_x$. Without loss of generality, let

$$\mathbf{D} = [\mathbf{D}_r : \mathbf{D}_x] \tag{2}$$

We want to find a *routing query* $\mathbf{q}$, such that $\mathbf{D}_r^T\mathbf{q} = \mathbf{1}$ and $\mathbf{D}_x^T\mathbf{q} = \mathbf{0}$. Thus, we write

$$[\mathbf{D}_r : \mathbf{D}_x]^T\mathbf{q} = [\mathbf{1} : \mathbf{0}]^T \tag{3}$$

Using the method of least squares, it can be shown [1] that solution to (3) is given by

$$\mathbf{q} = \mathbf{R}^{-1} \sum_{i=1}^{m_r} \mathbf{d}_{r_i} \tag{4}$$

The solution involves the inversion of a very large nxn correlation matrix, $\mathbf{R} = \mathbf{D}\mathbf{D}^T$. Unfortunately, unlike standard pattern recognition problems, document routing tasks are peculiar as they typically create very sparse, or *empty* feature spaces. That is, the number of training documents, m, is usually very small compared to the number of dimensions, n. Thus, $\mathbf{R}$ is usually not invertible.

Consequently, it is necessary to project the raw document space into a lower dimensional space where each axis possesses higher representational richness. We can project the raw document space $\mathbf{D}$ into a lower dimensional space $\mathbf{V}$ as follows:

$$\mathbf{V} = \mathbf{U}^T\mathbf{D} \tag{5}$$

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$ are the first k orthonormal basis that account for most information in the raw space. The solution for routing query $\mathbf{q}$ is then given by (see [1]) :

$$\mathbf{q} = \mathbf{U}\mathbf{\Lambda}^{-1} \sum_{i=1}^{m_r} \mathbf{U}^T\mathbf{d}_{r_i} \tag{6}$$

where

$$\mathbf{\Lambda} = \mathbf{V}\mathbf{V}^T = diag(\lambda_1, \lambda_2, \ldots, \lambda_k) \tag{7}$$

is a diagonal matrix containing the eigenvalues of $\mathbf{R}$ in descending order.

Now, $\mathbf{U}$ and $\mathbf{\Lambda}$ can be computed by performing principal component analysis on matrix $\mathbf{R}$ [3]. This method is equivalent to singular value decomposition[3] (SVD) employed by Latent Semantic Indexing [4]. However, the order of complexity of the standard SVD algorithm [5] is $O(n^2 + k^3)$. Since frequently $n >> m$, it will be advantageous to find an alternative method that exploits this characteristics.

The next two subsections describe the discriminant projection algorithm, which uses only the m documents with relevance information to compute the optimum projection. It first contructs a subspace based on the training documents using the Gram-Schmidth procedure. The dimension of this subspace will be at most m. It then performs principal component analysis on this subspace to extract orthonormal basis which are ordered by their importance in capturing the subspace information. By specifying the desirable representation richness, only $k < m$ orthonormal basis needs to be extracted.

2.1 Gram-Schmidt Procedure

The procedure makes use of training documents $\mathbf{d} \in \mathbf{D}$. We begin the procedure with the initialization

$$\mathbf{y}_1 = \mathbf{d}_1 \tag{8}$$

We then construct $\mathbf{y}_2$ as follows :

$$\begin{aligned} \mathbf{x}_2 &= \mathbf{d}_2 - \mathbf{d}_2^T \mathbf{y}_1 \mathbf{y}_1 \\ \mathbf{y}_2 &= \frac{\mathbf{x}_2}{|\mathbf{x}_2|} \end{aligned} \tag{9}$$

The vector $\mathbf{y}_2 = \mathbf{0}$ if and only if $\mathbf{d}_2 = a\mathbf{d}_1$, meaning that $\mathbf{d}_2$ is linearly dependent upon $\mathbf{d}_1$. Assume that $\mathbf{y}_2 \neq \mathbf{0}$, then $\mathbf{y}_1$ and $\mathbf{y}_2$ are linearly independent and orthogonal :

$$\mathbf{y}_2^T \mathbf{y}_1 = \mathbf{0} \tag{10}$$

We continue this procedure, generating each new vector $\mathbf{y}_j$ as follows :

$$\begin{aligned} \mathbf{x}_j &= \mathbf{d}_j - \sum_{i=1}^{j-1} (\mathbf{d}_j^T \mathbf{y}_i)\mathbf{y}_i \\ \mathbf{y}_j &= \frac{\mathbf{x}_j}{|\mathbf{x}_j|} \end{aligned} \tag{11}$$

This procedure maps the vectors $(\mathbf{d}_1, \mathbf{d}_2, \ldots \mathbf{d}_m)$ into vectors $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_m)$ vectors. In the latter set, there will be $r \leq M$ nonzero vectors that the linearly independent which are retained. The discarded vectors are linearly dependent on the r linearly independent vectors and do not carry additional information on the training documents.

Note that the complexity of this algorithm is $O(nr^2)$ and is linear with respect to the number of terms n. In the worst case, the complexity will be $O(nm^2)$.

[3]Please refer to [3] for the relationships between singular values and eigenvalues.

2.2 Eigen-Decomposition of the Resultant Subspace

Without loss of generality, we write the resultant nonzero vectors as follows : $(\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_r)$. We can obtain the rxr correlation matrix $\mathbf{G}$ in this space by simply computing the projection on each training documents on the $\mathbf{y}$ vectors :

$$\mathbf{G}(p,q) = \sum_{i=1}^{m} \mathbf{d}_i^T \mathbf{y}_p \cdot \mathbf{d}_i^T \mathbf{y}_q \tag{12}$$

We can then apply principal component analysis to extract eigenvectors of this r-dimensional subspace. In our implementation, we use the Hotelling's Iterative Procedure [7] to solve the eigen-decomposition problem. The procedure iteratively obtain eigenvectors $\mathbf{w}_1, \mathbf{w}_2, \ldots$ that corresponds to eigenvalues $\lambda_1, \lambda_2, \ldots$ in decending order. One can simply terminate the procedure if a pre-specified threshold t has been exceeded :

$$\sum_{i=1}^{k} \lambda_i / tr(\mathbf{G}) \geq t \tag{13}$$

$tr(\mathbf{G})$ is the trace of the matrix $\mathbf{G}$ that denotes the *total* variance available in the training documents. For example, if $t = 0.75$, then the first k eigenvectors account for more than 75% of the information in the training documents.

In the Hotelling's procedure, the iteration of each eigenvectors typically take only a few steps to converge. Since the algorithm operates on $\mathbf{G}$, its complexity is $O(m^2)$ in the worst case.

Finally, the orthonormal basis $\mathbf{u}$ can be obtained via

$$\mathbf{u}_i = \sum_{j=1}^{k} w_{ij} \mathbf{y}_j \tag{14}$$

where w_{ij} are the coefficients from eigenvectors $\mathbf{w}_i$. Each $\mathbf{u}_i$, in diminishing importance, captures an orthogonal axis containing the semantics of the training documents. Thus $\mathbf{u}_1$ is a linear combination of terms containing the highest information contents in the training documents. $\mathbf{u}_1$ is orthogonal to $\mathbf{u}_2$ *i.e.*, $\mathbf{u}_1^T \mathbf{u}_2 = 0$: it captures information contents missed by $\mathbf{u}_1$, and so on.

2.3 Applying Discriminant Projection Sequentially

Now, suppose that we have applied semantic projection on a set of $m-1$ documents to obtain a set of k orthogonal basis in $\mathbf{U}_{m-1} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k]$. Correspondingly, we have a kxk correlation matrix $\mathbf{\Lambda}_{m-1}$ which contains the eigenvalues in its diagonal. We wish now to expand our semantic projection to encode information of a new document $\mathbf{d}_m$.

To achieve this purpose, we simply restart the Gram-Schmidt procedure by initializing

$$\mathbf{y}_i = \mathbf{u}_i \tag{15}$$

for $i = 1, 2, \ldots, k$. We then obtain the new Gram-Schmidt vector $\mathbf{y}_{k+1}$ using (11) and the corresponding correlation matrix $\mathbf{G}_m$ using (12). With a little book-keeping, it can easily be shown that

$$\mathbf{G}_m = \begin{bmatrix} \Lambda_{m-1} & \mathbf{U}_{m-1}^T \mathbf{y}_{k+1} \\ \mathbf{y}_{k+1}^T \mathbf{U}_{m-1} & \epsilon \end{bmatrix} \tag{16}$$

where $\epsilon = 1 - \mathbf{y}_{k+1}^T \mathbf{U}_{m-1} \mathbf{U}_{m-1}^T \mathbf{y}_{k+1}$ is a measure of *residual* projection. We can now obtain a new set of orthogonal basis $\mathbf{U}_m$ by performing principal component analysis on $\mathbf{G}_m$.

3 Routing Experiments

We experimented only with documents that appear on the relevant list of at least one query. All other documents that do not belong to this group are ignored. Consequently our training documents are drawn only from approximately 18,000 documents, rather than from the entire TREC collection.

Each document is automatically parsed and index terms representing the document content are retrieved. A simple token recognizer written in flex is used to identify words within the document structure that contributes to the content. Most words are indexed except very common words, maintained in a stopword list, and numbers are removed. The remaining words are then stemmed using the Porter's stemming algorithm. The document indexing process next creates the inverted files using Faircom's c-tree Plus file management product. Currently, without compression, the overhead for the indices amounts to about 80% of the original database size.

Using the training documents, we compute the *idf* weights for each term, *i.e.* $idf_i = log(N/n_i)$. Weights of terms seen only in test documents but not in training documents are set to 1.

We construct the initial routing query $\mathbf{q}_i$ by using all relevant documents. It can be shown that $\mathbf{q}_i = \mathbf{m}$, the mean (centroid) of the relevant documents. Using $\mathbf{q}_i$, we ranked all training documents $\mathbf{d}$ by computing their projection score $\mathbf{q}_i^T\mathbf{d}$.

From the initial ranking, we select the top X % of the non-relevant documents, and compute the new routing query using the sequential formula. For *itidp*1, we set X to 10 %, *i.e.*, we use $0.1 * m_r$, where m_r is the number of relevant documents. For *itidp*2, we set X to 30 %. The results of the experiments are shown in Figure 1.

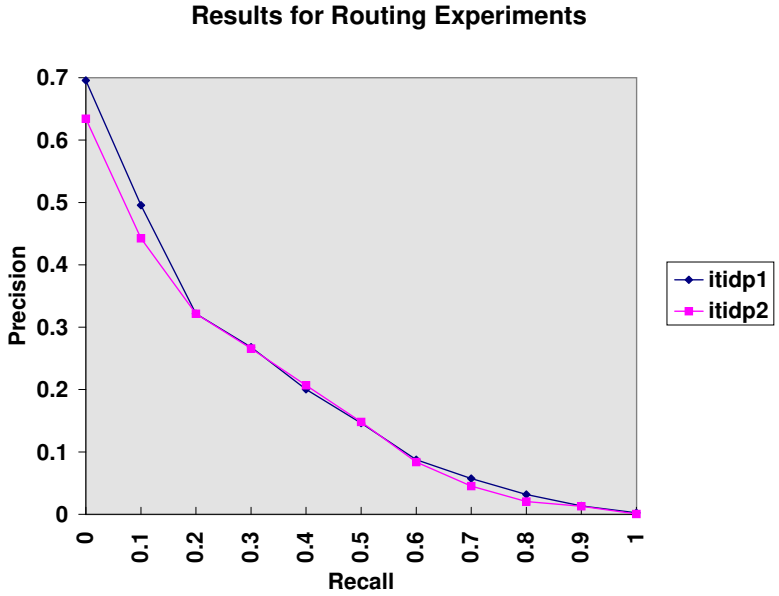

Figure 1: Results for Routing Experiments

From the experience gathered through our previous experiments, we have expected that *itidp*2 to be a high precision run, out-performing *itidp*1 in the high precision region. This is because more non-relevant documents were used in constructing *itidp*2. However, the results obtained show that *itidp*1 consistently out-performs *itidp*2 at all recall levels.

In our routing experiments, we have not used the original query topics, but instead relied completely on the training documents. On hindsight, this is probably a mistake. In our post-mortem, we have found that we can significantly improve the results by assigning heavy weights to the original query topics.

4 Filtering Experiments

We used *itidp*1 as the filtering query and attempted to find the best threshold to maximise the utility measures. We first divide the training documents a equally into two sets, and construct the routing queries from the first set using the discriminant projection algorithm. We then use these queries to obtain the scores for all documents in the second set. For each query, we find the respective thresholds that best optimise the utility measures. We then apply these thresholds, together with routing query constructed from the entire training set, to filter new documents in the test set.

5 Chinese Text Retrieval

We experimented both with an automatic method and a manual method.

For the automatic method (*iticn*1), we use all *phrases* in the description field and compute the aggregate scores using the following formula :

$$s_k(\mathbf{d}_i) = \frac{\sum_{j=1}^{Q}(IDF_j * freq_{ij})}{C * \sqrt{\mathbf{d}^T\mathbf{d}}} \tag{17}$$

where Q is the number of matching phrases between document i and query k, IDF_j is the IDF weight for term j in the entire collection, and $freq_{ij}$ is the frequency of phrase j in document i. C is a constant which is selected to avoid heavily penalising long documents, and is set to 1000 in our experiment.

For the manual method *iticn*2, we construct boolean phrase queries which are thought to improve the results. The queries used are shown in Figure 5.

The boolean constructs are fuzzy, *i.e.*, let $s_x(\mathbf{d}_i)$ and $s_y(\mathbf{d}_i)$ are scores obtained from two phrase queries x and y, then

$$\begin{aligned} x \text{ AND } y &\rightarrow \min(s_x, s_y) \\ x \text{ OR } y &\rightarrow \max(s_x, s_y) \\ x \text{ NOT } y &\rightarrow \min(s_x, 1 - s_y) \\ x \mid y &\rightarrow s_x + s_y \end{aligned} \tag{18}$$

The results for *iticn*1 and *iticn*2 are shown in Figure 3. It can be seen that adding boolean constructs significantly improve the results.

1 最惠国待遇 and 人权 and (分离 or 脱钩)
2 中国 and 台湾 and 一国两制
3 核电站|大亚湾|秦山|安全
4 油田|天然气|油气|储量|油质
5 (知识产权法 or 商标法 or 著作权法 or 专利法) and 保护 not 贸易制裁
6 中国 and (世界贸易 and (组织 or 体系) or 关贸总协 or 市场准入 or 多边贸易) and (成员 or 加入 or 支持)
7 (南沙 or 东沙 or 西沙) and (主权 or 争端 or 建议 or 资源)
8 日本 and 地震
9 (中国 or 我国) and (毒品 or 可卡 因 or 大麻 or 海洛因)
10 (新疆 or 维吾尔) and (边境贸易 or 边贸)
11 (波斯尼亚 or 前南斯拉夫 or 巴尔干) and 联合国 and (维和 or 维持和平)
12 妇女大会 and (妇女问题 or 妇女地位)
13 中国 and (奥运 or 世界运动大会 or 奥林匹克) and 申办
14 (中国 or 云南 or 我国) and (爱滋病 or 艾滋病 or HIV)
15 海地 and (联合国 or 美国 or 多国部队 or 维和部队) and 民主
16 伊拉克 and 经济制裁
17 亚太经济合作组织|中国|关贸总协| 世界贸易组织
18 以色列|巴勒斯坦|中东|和平会 议
19 中国|希望工程|文化程度|教育
20 越南 and 失踪 and (美军 or 美国 and 军人)
21 (香港 or 特别行政区) and 彭定康
22 疟疾 and (死亡 or 感染 or 病例)
23 苏联 and (调停 or 和平建议) and (海湾战争 or 伊拉克)
24 (波黑 or 波斯尼亚 or 黑塞哥维那) and 武器禁运 and (取消 or 解除)
25 熊猫 and (生态 or 环境 or 保护 or 灭绝 or 繁殖)
26 (中国 or 蒙古 or 东北) and 森林 and (火灾 or 大火 or 防火)
27 (中国 or 我国) and 机 器人
28 (中国 or 我国) and 移动电话 and (数字 or 蜂窝式 or 网络 or 自动漫游)

Figure 2: Manual Queries used in Chinese Experiments

Through these experiments, a major difference between Chinese and English text retrieval was discovered. Typically in English text retrieval, the number of terms M, is significantly greater than the number of documents, N, *i.e.* $M >> N$). This is not so in Chinese text retrieval. There are less than $10,000$ Chinese characters encoded in the GB codeset, and this number remains constant even when N keeps increasing.

This is an important characteristics which must be taken into account when designing a Chinese text retrieval system. In our existing system which was originally designed for the English language, phrase retrieval is performed by first retrieving the position lists of the constituent terms, followed by proximity check. Since $M << N$ for large N, each term is likely to occur in many documents, making proximity verification very expensive to perform.

This problem can be alleviated by performing word segmentation or by using n-gram processings. Both methods increase M and in some cases eliminate the needs to perform proximity verification. This approach of retrofitting a text retrieval system from English language to Chinese language appears to be very popular among the Chinese track participants.

We intend to deviate significantly from this paradigm in our next participations. We believe that a different paradigm, designed by taking into considerations the unique characteristics of Chinese language, should be proposed and tested.

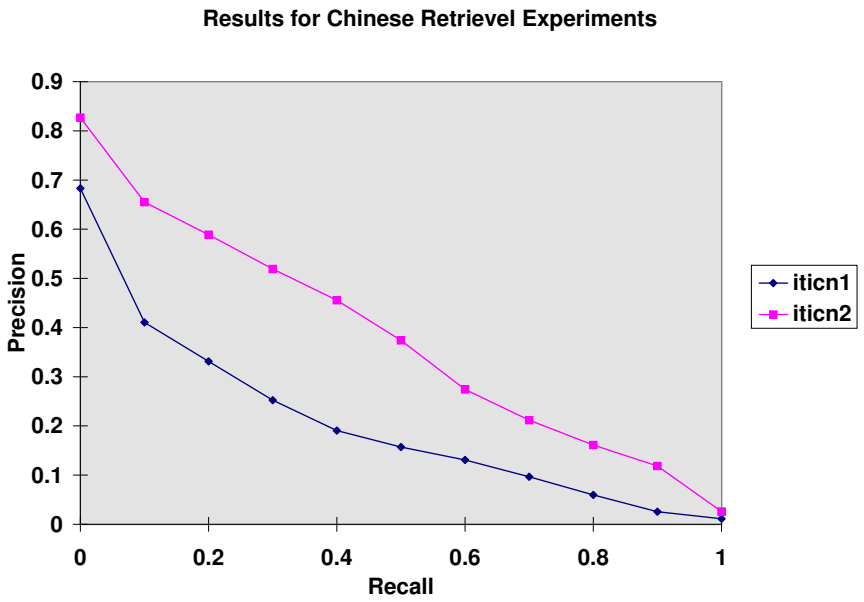

Figure 3: Results for Chinese Retrieval Experiments

6 Conclusion

We have described our experiments in the routing, filtering and Chinese text retrieval experiments. The following are our observations :

- For routing experiments, using more non-relevant documents appear to have hurt performance. It is probably a mistake to use only the training documents but ignore the original query topics.
- In our Chinese text retrieval, using less query phrases operated by Boolean conjuncts increased performance significantly.

References

[1] Kok F. Lai, Vincent Lee & Jeremy Chew, "Document Routing by Discriminant Projection," *Overview of the Fourth Text Retrieval Conference (TREC-4)*, 1995, to appear.

[2] D. Harman, "The DARPA TIPSTER Project," *SIGIR Forum*, 26(2), pp. 26-28, 1992.

[3] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis*, Addison-Wesley, 1991.

[4] S. T. Dumais, G. W. Furnas & T. K. Landauer, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, 1990, pp. 391-407.

[5] R. A. Harshman & M. E. Lundy, "Data preprocessing and the extended PARAFAC model," *Research Methods for MultiMode Data Analysis*, edited by H. G. Law et. al., 1984.

[6] D. Hull, "Improving Text Retrieval for the Routing Problem using Latent Semantic Indexing," *Proc SIGIR '94*, 1994, pp. 282-291.

[7] H. Hotelling, "Analysis of a Complex Statistical Variables into Principal Components," *Journal of Educational Psychology*, 1933, pp. 417-441, 498-520.

[8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 1986.