Pseudo Relevance Feedback Method Based on Taylor Expansion of Retrieval Function in NTCIR-3 Patent Retrieval Task

Kazuaki KISHIDA
Faculty of Cultural Information Resources
Surugadai University
698 Azu, Hanno, Saitama 357-8555 JAPAN
kishida@surugadai.ac.jp

Abstract

Pseudo relevance feedback is empirically known as a useful method for enhancing retrieval performance. For example, we can apply the Rocchio method, which is well-known relevance feedback method, to the results of an initial search by assuming that the top-ranked documents are relevant. In this paper, for searching the NTCIR-3 patent test collection through pseudo feedback, we employ two relevance feedback mechanism; (1) the Rocchio method, and (2) a new method that is based on Taylor formula of linear search functions. The test collection consists of near 700,000 records including full text of Japanese patent materials. Unfortunately, effectiveness of our pseudo feedback methods was not empirically observed at all in the experiment.

1 Introduction

Relevance feedback is widely recognized as an effective method for improving retrieval effectiveness in the context of interactive IR. As often pointed out, it is difficult for users to represent their own information needs into a well-defined set of search terms or statements. The resulting short or poor queries would bring them unsatisfactory results. However, if a few relevant documents happen to be found by the search, we could automatically or manually extract some useful terms from the documents, and add them to the initial search expression. It is obviously expected that search effectiveness of the second search using the extended query will be improved significantly. This is a basic idea of relevance feedback.

Inevitably, for executing automatic relevance feedback, the system has to obtain relevance information, i.e., relevant or irrelevant documents, from the users interactively. However, some researchers have tried to employ relevance feedback techniques with no relevance information. The objective is to enhance search performance of retrieval models such as vector space model, probabilistic model and so on, without interaction on relevance information between system and users. The technique is usually called pseudo relevance feedback, in which a standard feedback method (e.g., the Rocchio method) is applied by assuming that top-ranked documents searched by the initial search are relevant.

The purpose of this paper is to report results of retrieval experiments for examining effectiveness of pseudo relevance feedback in the case of searching a patent collection. In particular, we attempt to compare search performance of the traditional Rocchio method with that of an alternative method, which is based on Taylor approximation of retrieval function proposed by Kishida[1]. This report is based on two experiments using the NTCIR-1 test collection and the NTCIR-3 patent test collection, respectively. As to the latter, the results were obtained at the time of NTCIR-3 Workshop held in October 2002 [2].

The rest of this paper is organized as follows. In Section 2, the Rocchio method and an alternative method proposed by Kishida[1] will be introduced. In Section 3 a preliminary experiment for confirming how well the alternative method works in a normal relevance feedback situation will be described. The NTCIR-1 test collection with relevance judgment information is used for the preliminary experiment. In Section 4, results of an experiment on pseudo relevance feedback method

using the NTCIR-3 patent test collection will be shown.

2 Relevance Feedback Methods

2.1 Rocchio Method

The most typical approach to relevance feedback would be so-called the Rocchio method [3]. A basic idea of the method is to add an average weight of each term within a set of relevant documents to the original query vector, and to subtract an average weight within a set of irrelevant ones from the vector.

We denote a document vector and a query vector by $\mathbf{d}_i = (w_{i1},...,w_{iM})^T$ and $\mathbf{q} = (w_{q1},...,w_{qM})^T$, where w_{ij} is a weight of a term within a document and w_{qj} is a weight of a term within the query (M is the total number of distinct terms in the database, and T indicates transposition).

A modified query vector is obtained by a formula,

$$\tilde{\mathbf{q}} = \alpha\mathbf{q} + \frac{\beta}{|D|}\sum_{i:d_i \in D}\mathbf{d}_i - \frac{\gamma}{|\overline{D}|}\sum_{i:d_i \in \overline{D}}\mathbf{d}_i , \quad (1)$$

where D is the set of relevant documents, $\overline{D}$ is the set of irrelevant documents, and α, β, and γ are constants.

It has been empirically shown that the performance of the Rocchio method is very good [4], and in recent, many researchers have examined the method directly or indirectly [5-8]. Also, due to its effectiveness and simplicity, the Rocchio method has been widely applied in other research areas, for example, image retrieval [0] or text categorization [10].

2.2 Feedback Method Using Taylor Formula of Retrieval Function

Kishida[1] has proposed an alternative relevance feedback method, which is suitable for the situation that the degree of relevance is given as a numerical value, not dichotomous value (i.e., relevance or not), from actual users. In this section, according to Kishida[1], the method will be explained.

In vector space model [10], typical formulas for determining term weights are as follows:

$$w_{ij} = \log x_{ij} + 1.0 \quad (2)$$

$$w_{qj} = (\log x_{qj} + 1.0)\log\left(N/n_j\right) \quad (3)$$

where

x_{ij}: frequency of a term t_j in a document d_i,

x_{qj}: frequency of a term t_j in the query,

n_j: the number of documents including t_j,

N: the total number of documents in the database.

For calculating the degree of similarity between a document vector $\mathbf{d}_i$ and the query vector $\mathbf{q}$, a cosine formula is normally used:

$$s_i = \sum_{j=1}^{M} w_{ij}w_{qj} \Big/ \sqrt{\sum_{j=1}^{M} w_{ij}^2 \sum_{j=1}^{M} w_{qj}^2} \quad (4)$$

where s_i is a numerical score indicating similarity of the document given a query vector.

On the other hand, a well-known formula based on probabilistic model derived from an assumption of two-Poisson distribution [12] is

$$s_i = \sum_{j=1}^{M}\left(\frac{3.0x_{ij}}{(0.5+1.5l_i/\bar{l})+x_{ij}} \times x_{qj} \times \log\frac{N-n_j+0.5}{n_j+0.5}\right) \quad (5)$$

where

$l_i = \sum_{j=1}^{M} x_{ij}$, and $\bar{l} = N^{-1}\sum_{i=1}^{N} l_i$,

i.e., the former is a document length, and the latter is an average of the length over documents within the database. The formula (5) is a version of so-called Okapi weighting [12] under a particular setting of its parameters.

We can represent concisely the two important retrieval models as a linear function of vector,

$$\mathbf{s} = f(\mathbf{b}) = \mathbf{A}\mathbf{b}, \quad (6)$$

where $\mathbf{s}$ is a N dimensional vector of document scores, $\mathbf{s} = (s_1,...,s_N)^T$, f is a linear function of vector ($f: R^{M\times 1} \to R^{N\times 1}$), and $\mathbf{A}$ is a $N \times M$ matrix of which each element is

$$a_{ij} = (\log x_{ij} + 1.0) \Big/ \sqrt{\sum_{j=1}^{M}(\log x_{ij} + 1.0)^2}, \quad (7)$$

in the case of vector space model (see (2) and (4)), or

$$a_{ij} = \frac{3.0x_{ij}}{(0.5+1.5l_i/\bar{l})+x_{ij}} \quad (8)$$

in the case of the Okapi formula (see (5)).

Also, $\mathbf{b}$ is a M dimensional vector of which each element is defined as

$$b_j = w_{qj} \Big/ \sqrt{\sum_{j=1}^{M} w_{qj}^2} \quad (9)$$

where $w_{qj} = (\log x_{qj} + 1.0)\log\left(N/n_j\right)$ in the case of vector space model (see (3)), or

$$b_j = x_{qj} \log \frac{N - n_j + 0.5}{n_j + 0.5} \quad (10)$$

in the case of the Okapi formula (see (5)).

The most important thing is that both of two well-known formulas for estimating scores to rank documents are able to be represented by a simple form (6).

For making ranked output, documents have to be sorted in the decreasing order of scores, s_i ($i = 1,...,N$). This means that each score is assumed to indicate the degree of relevance. In other words, the score is expected to be an estimate of 'true' degree of relevance r_i.

Let $\mathbf{r} = (r_1,...,r_N)^T$ be a vector representing 'true' relevance degrees. By using this notation, we can describe operationally a purpose of retrieval system as "to estimate a vector $\mathbf{s}$ that is the closest to vector $\mathbf{r}$ when a search request is given."

Of course, $\mathbf{r}$ is unknown in real situations, but it is possible to obtain information on a part of $\mathbf{r}$ through the process of relevance feedback. For example, if a user replies a set of scores indicating the degrees of relevance for top-ranked n documents searched by the initial query, the scores allow us to estimate the part of $\mathbf{r}$ corresponding to the n documents

We denote a set of the top-ranked n documents by X and the part of $\mathbf{r}$ corresponding to the set X by $\mathbf{r}_X$, which is actually n dimensional vector reconstructed by extracting n elements of the documents from the original vector $\mathbf{r}$. According to (6), we can write that

$$\mathbf{s}_X = f_X(\mathbf{b}) = \mathbf{A}_X \mathbf{b}, \quad (11)$$

where

$\mathbf{A}_X$: an $n \times M$ matrix,

$\mathbf{s}_X$: an n dimensional vector, and

$f_X : R^{M \times 1} \to R^{n \times 1}$.

Both of the matrix and the vector are constructed by the same way with $\mathbf{r}_X$.

If we establish a distance measure ϕ between $\mathbf{r}_X$ and $\mathbf{s}_X$, the objective of relevance feedback can be formally described as follows: the relevance feedback aims at estimating a modified query vector such that

$$\tilde{\mathbf{b}} = \arg\min_{\mathbf{b}} \phi(\mathbf{r}_X, \mathbf{s}_X) = \arg\min_{\mathbf{b}} \phi(\mathbf{r}_X, f_X(\mathbf{b})). \quad (12)$$

Then we can use $\tilde{\mathbf{b}}$ for the secondary search.

An approach to estimating $\tilde{\mathbf{b}}$ is to pay our attention to a difference between initial score $f_X(\mathbf{b})$ and secondary score $f_X(\tilde{\mathbf{b}})$, and to apply so-called Taylor approximation for obtaining a vector function $f_X(\tilde{\mathbf{b}})$, i.e.,

$$f_X(\tilde{\mathbf{b}}) = f_X(\mathbf{b}) + \frac{\partial f_X(\mathbf{b})}{\partial \mathbf{b}^T}(\tilde{\mathbf{b}} - \mathbf{b}) + K, \quad (13)$$

where K is a residual term (see [13]). If we employ (11) and assume that

$\mathbf{r}_X = f_X(\tilde{\mathbf{b}})$,

according to a target condition (12), we can obtain that

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{A}_X^{-1}(\mathbf{r}_X - \mathbf{s}_X), \quad (14)$$

(see Appendix for detail calculation). It should be noted that $K = 0$ due to the linearity of Equation (11). This means (14) is not an approximation but an exact relation.

The Equation (14) contains an abnormal inverse matrix $\mathbf{A}_X^{-1}$, which is an $M \times n$ matrix and $\mathbf{A}_X^{-1}\mathbf{A}_X = \mathbf{I}_M$ where $\mathbf{I}_M$ is a $M \times M$ matrix of which all diagonal elements are 1 and others are 0. Using singular value decomposition (SVD), transpose matrix of $\mathbf{A}_X$ can be represented as

$\mathbf{A}_X^T = \mathbf{U}\Lambda\mathbf{V}^T$,

where

$\mathbf{U}$: an $M \times n$ orthogonal matrix,

Λ : an $n \times n$ diagonal matrix, and

$\mathbf{V}$: an $n \times n$ orthogonal matrix.

By employing the decomposition, we can finally represent (14) such as

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X) \quad (15)$$

(see Appendix for details). This is a final formula of our relevance feedback algorithm. For convenience, we call the algorithm "the Taylor formula based method" in this paper.

3 Preliminary Experiment with Relevance Information

3.1 Purpose and Test Data

Before applying pseudo relevance feedback based on the Equation (15) to the patent test collection, we try checking retrieval performance of the Taylor formula based method by using other test collection with relevance judgment information. To do this, we employ a well-known Japanese Test Collection NTCIR-1 (NII/NACSIS Test Collection

for Information Retrieval - 1)[1], which consists of about 330,000 bibliographic records of proceedings at conferences held in Japan. It should be noted that, in the preliminary experiment, relevance judgment information was used (i.e., not pseudo feedback).

Fifty-three topics of NTCIR-1 were employed for the experiment (from No.31 to No.83). The format of these topics is also very similar with that of TREC, i.e., a record of each topic consists of fields of <title>, <description>, <narrative> and so on. For realistic situation in which feedback methods are used, it would be more reasonable to assume that original search statements are short. Thus we employed only <title> and <description> fields for representing each topic. This means that a kind of 'short query' was used for the experiment.

3.2 Procedure and type of runs

Procedure of the preliminary experiment is as follows:

(a) Initial search: two initial search runs were carried out, i.e., the first is based on vector space model from (2) to (4) and the second is probabilistic model (5). We denote the initial search runs as ORGVEC and ORGPRB, respectively.

(b) Query modification through relevance feedback: initial queries were modified by using relevance judgment information on top-ranked n documents of each initial run. In this paper, we set 10 and 20 as the value of n,
   - In the case of vector space model, we can attempt two modification methods, i.e., the Rocchio method (1) (where $\alpha = 8$, $\beta = 16$ and $\gamma = 4$) and the Taylor formula based method (7), (9) and (15). The run using the Rocchio method is denoted as ROCCHI, and the run by the Taylor formula based method as TYLVEC.
   - In the case of probabilistic model, only the Taylor formula based method was applied using (8), (10) and (15). We denote this run as TYLPRB.

(c) Secondary search: each modified query was used for second run
   - In the case of ROCCHI, modified queries were matched with document vectors by cosine formula (4).
   - In the case of runs based on the Taylor formula, TYLVEC and TYLPRB, the linear function (6) was used for matching operation.

3.3 Conversion of binary judgment into continuous value

One of the advantages of the Taylor formula based method (15) is to allow us for making use of continuous values representing the degree to which each document is relevant. Unfortunately, in the experiment, such values were not available because only results of binary judgments are officially provided as relevance information.

Therefore, in order to testify the Taylor formula based method, we need to develop a special algorithm for converting each binary judgment to a continuous score. An easy way for converting a value of binary judgment into a continuous degree of relevance is to predict the degree from a document score in initial search by using a simple regression, $r_i = As_i + B$.

It would be straightforward that the constants A and B are determined based on maximum and minimum values of s_i and r_i for relevant and irrelevant documents independently. That is, we use a set of eight values for parameter estimation as follows.

- $s^1_{\max}$ and $s^1_{\min}$: maximum and minimum values of s_i for 'relevant' documents in top-ranked n documents,
- $s^0_{\max}$ and $s^0_{\min}$: maximum and minimum values of s_i for 'irrelevant' documents in top-ranked n documents,
- $r^1_{\max}$ and $r^1_{\min}$: maximum and minimum values of r_i for 'relevant' documents in top-ranked n documents,
- $r^0_{\max}$ and $r^0_{\min}$: maximum and minimum values of r_i for 'irrelevant' documents in top-ranked n documents.

For the set of relevant documents, we can obtain estimates of A and B by solving equations,

$$\begin{cases} r^1_{\max} = As^1_{\max} + B \\ r^1_{\min} = As^1_{\min} + B \end{cases}$$

It is easy to show that

$A = (r^1_{\max} - r^1_{\min}) / (s^1_{\max} - s^1_{\min})$ and

$B = (s^1_{\max} r^1_{\min} - r^1_{\max} s^1_{\min}) / (s^1_{\max} - s^1_{\min})$.

[1] http://research.nii.ac.jp/ntcir/

Similarly, for the set of irrelevant documents, we obtain that

$A = (r^0_{\max} - r^0_{\min}) / (s^0_{\max} - s^0_{\min})$ and

$B = (s^0_{\max} r^0_{\min} - r^0_{\max} s^0_{\min}) / (s^0_{\max} - s^0_{\min})$.

Furthermore, we have to determine *a priori* values of $r^1_{\max}$, $r^1_{\min}$, $r^0_{\max}$ and $r^0_{\min}$,

(a) For vector space model, it is reasonable that $r^1_{\max}$ is assumed to be 1.0 and $r^0_{\min}$ is 0.0 according to cosine function (4). As for $r^1_{\min}$ and $r^0_{\max}$, it is necessary to set a margin between them, i.e., amount of difference from minimum value for relevant documents to maximum value for irrelevant ones. If we take the margin as 2.0, it is automatically determined that $r^1_{\min} = 6.0$ and $r^0_{\max} = 4.0$. As a result, target values r_i for relevant and irrelevant documents are distributed from 0.6 to 1.0, and from 0.0 and 0.4, respectively.

(b) For probabilistic model, we set arbitrarily that $r^1_{\max} = 2s^1_{\max}$, $r^1_{\min} = s^1_{\max}$ and $r^0_{\min} = 0.0$ as a trial in the experiment. This means that range of document scores is enlarged doubly, and each r_i for relevant documents is to be distributed in the range from $s^1_{\max}$ to $2s^1_{\max}$. On the other hand, maximum value of r_i for irrelevant documents is complicated a little, i.e.,

$$r^0_{\max} = \min(s^1_{\min}, s^0_{\min}) + [\max(s^1_{\max}, s^0_{\max}) - \min(s^1_{\min}, s^0_{\min})]/2,$$

since there is no guarantee that $s^1_{\max}$ is always greater than $s^0_{\max}$, and $s^0_{\min}$ is always smaller than $s^1_{\min}$

3.4 Segmentation of Japanese text

The test collection NTCIR-1 basically consists of documents written in Japanese (as well as the NTCIR-3 patent test collection). We need to segment each text into a set of terms automatically for indexing the Japanese documents and queries, of which text has no explicit boundary between terms unlike English.

In the experiment, each term was identified by matching with entries in a machine-readable dictionary. We used a dictionary of the ChaSen[14], which is a well-known morphological analyzer for Japanese text, and selected each longest entry matched with a portion of text as a index term. Also, an "unknown" string was decomposed according to change of kinds of character, *Hiragana*, *Katakana*, *Kanji* and so on *(Hiragana* strings were not taken as index terms).

Also, for identifying compound words as content-bearing terms, we employed a heuristic rule that an adjacent pair of index terms identified by dictionary matching is automatically combined into a compound word.

3.5 Results

The NTCIR-1 collection includes 332,918 records, and average document length, i.e., average of the total number of terms appearing in each document, was 118.0. Table 1 shows values of mean average precision of each run.

As shown in Table 1, the Taylor formula based method outperforms the Rocchio method slightly, but clearly there is no statistically significant difference between ROCCHI and TYLVEC (.376 and .378 at top 10, and .434 and .459 at top 20).

The rate of improvement by feedback in vector space model is greater than that in probabilistic model. The run showing best performance in Table 1 is the Taylor formula based method in the vector space model (TYLVEC) using top-ranked 20 documents, which increases mean average precision at 101.6% from ORGVEC (from .228 to .459).

4 Experiment on Pseudo Relevance Feedback using NTCIR-3 Patent Test Collection

4.1 Procedure

In the previous section, the Taylor formula based method has proven to work well at the experiment using the NTCIR-1 test collection with relevance information. Next, we attempt to examine the effectiveness of pseudo relevance feedback method using the Taylor formula based feedback in the case of searching the NTCIR-3 patent test collection (with no relevance information).

The method and procedure are almost same with those in the previous section. However, in the Rocchio method, $\overline{D}$ is assumed to be empty (γ is 0 in the Equation (1)).

Table 1. Mean average precision (using the NTCIR-1 collection with relevance information)

model	Vector space		Probabilis-tic
initial search (baseline)	ORGVEC .228		ORGPRB .268
feedback	ROCCHI	TYLVEC	TYLPRB
top 10 documents	.376 (+65.2%)	.378 (+66.3%)	.396 (+48.0%)
top 20 documents	.434 (+90.4%)	.459 (+101.6%)	.450 (+68.1%)

In the experiment, only six runs were executed as shown in Table 2 (at the time of the NTCIR-3 Workshop, only the six runs were submitted). We discern two kinds of run according to query (topic) fields used for run; (I) <ARTICLE> and <SUPPLEMENT> fields and (II) <DESCRIPTION> and <NARRATIVE> fields. The <ARTICLE> field includes a news article, i.e., in the NTCIR-3 Patent Task, the participants were asked to search the document collection for a news article related to the information needs of users. The number of topics is 32.

Table 2. Runs in the experiment using patent test collection

Initial run	feedback	Topic fields	
		<A><S>*	<D><N>**
OKAPI	TAYLOR	Run1	Run2
VECTOR	ROCCHIO	Run3	Run4
OKAPI	none	Run5	Run6

*<A>:<ARTICLE>, <S>:<SUPPLEMENT>
**<D>:<DESCRIPTION>,<N>:<NARRATIVE

4.2 Results

In the indexing phase, 697,262 records were processed and average length of documents is 393.32.

Table 3 shows search performance of each run. Unfortunately, pseudo relevance feedback using relevance feedback techniques has no effect on the performance. It seems that there are no statistically significant differences between any pairs of runs. However,

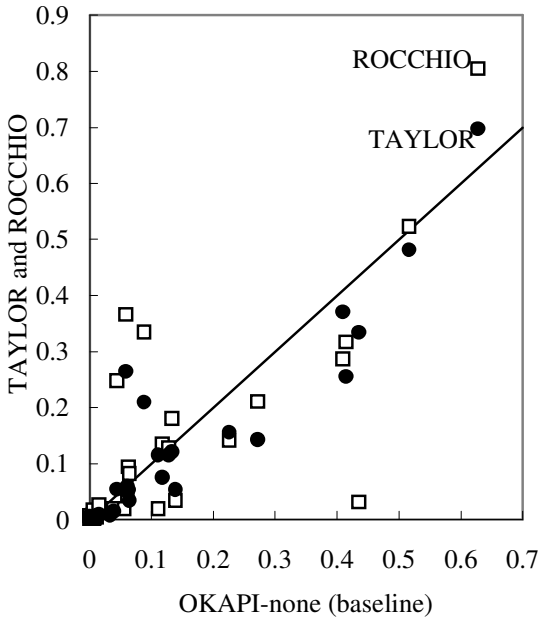

Figure 1. Topic-by-topic Analysis (in the case of using <DESCRIPTION> and <NARRATIVE>)

Figure 1 is a plot of values of average precision by topic. We can compare the Taylor formula based method (OKAPI-TAYLOR) and the Rocchio method (VECTOR-ROCCHIO) with normal Okapi formula (OKAPI-none), in level of each topic. It should be noted that, in Figure 1, square indicates ROCCHIO and circle TAYLOR.

Figure 1 shows that for most of topics, normal Okapi formula outperforms the Rocchio method and Taylor formula based method although the Rocchio method and Taylor formula based method are superior in some topics.

Table 3. Average Precision and R-precision (Using NTCIR-3 Patent Test Collection)

Topic Fields	Initial run	feedback	Average precision	R-precision
<ARTICLE> <SUPPLEMENT>	OKAPI	TAYLOR	0.1152	0.1421
	VECTOR	ROCCHIO	0.1281	0.1565
	OKAPI	none	0.1282	0.1565
<DESCRIPTION> <NARRATIVE>	OKAPI	TAYLOR	0.1370	0.1820
	VECTOR	ROCCHIO	0.1581	0.1896
	OKAPI	none	0.1583	0.1813

5 Discussion

Although the Rocchio method and Taylor formula based method have shown good performance in the preliminary experiment using the NTCIR-1 test collection with relevance judgment with relevance judgment information, unfortunately the pseudo relevance feedback was not able to show improvement of search effectiveness. A main reason for the failure may be that term selection process was omitted. In standard pseudo relevance feedback methods, better terms are usually selected from the set of top-ranked documents according to the term weights. We can expect that if the term selection process is applied, the performance is improved in the case of the Rocchio method. However, how can we select better terms in the case of the Taylor formula based method?

The behavior of the Taylor formula based method in the process of term re-weighting is a little complicated. For example, we assume that there are only 6 distinct terms (from term1 to term6) in a database, and that

$$\mathbf{b} = (0.5,0.5,0.5,0.5,0.5,0.5)^T,$$

which means that all term weights in the initial query vector are equal. The matrix of weights of terms in top-ranked 4 documents (from doc1 to doc4) is supposed to be that

$$\mathbf{A}_X = \begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 1 \\ 1 & 2 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 2 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 \end{pmatrix}. \quad (16)$$

A row of the matrix represents each document vector, e.g., $\mathbf{d}_1^T = (2,1,0,0,1,1)$

Furthermore, it is assumed that a set of numerical values indicating degree of relevance for each document was given by a user, and difference from initial document scores was calculates such that

$$\mathbf{r}_X - \mathbf{s}_X = (0.1,0.2,-0.1,-0.2)^T. \quad (17)$$

Under these assumptions, relevance feedback by the Taylor formula based method is as follows. First, by the SVD algorithm, the transpose matrix of the $\mathbf{A}_X$ can be decomposed as $\mathbf{U}\Lambda\mathbf{V}^T$, and after simple calculation, we can finally obtain that

$$\mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X) = (0.0,0.1,-0.1,0.0,0.0,0.0)^T. \quad (18)$$

This example represents well characteristics of the Taylor formula based method. From (17) we understand that scores of doc1 and doc2 have to be increased and those of doc3 and doc4 decreased. Intuitively, it seems that weights of both term1 and term2 should be augmented because they are only appearing in doc1 and doc2, neither doc3 nor doc4 at all. However, a solution by (18) indicates that the weight of term1 is unchanged (only to that of term2, 0.1 is added). This is a result so as to keep the condition (17), which means that scores of doc1 and doc2 have to be increased by 0.1 and 0.2, respectively, for reaching at an ideal situation. Actually, we can calculate from (16) such that $2\times0.0+1\times0.1=0.1$ for doc1 and that $1\times0.0+2\times0.1=0.2$ for doc2. The results indicate that the condition (17) is completely satisfied. As shown in the simple calculation, the Taylor formula based method takes the difference $\mathbf{r}_X - \mathbf{s}_X$ into consideration for re-weighting of search terms.

On the other hand, in the case of the Rocchio method, re-weighting of search terms is done by looking into only $\mathbf{A}_X$ regardless of $\mathbf{s}_X$. We suppose that doc1 and doc2 were judged as relevant documents, and doc3 and doc4 irrelevant. In the condition, the Rocchio method adds simply (1+2)/2=1.5 to weights of both of term1 and term2, not considering document scores in an initial search.

As shown in above example, in the case of the Taylor formula based method, term re-weighting is dependent on the values of $\mathbf{r}_X - \mathbf{s}_X$. Therefore, we can not use simply the vector (18) for selecting better terms. We have to consider carefully how to use the Equation (18) for term selection. Further investigation will be needed for executing term selection for pseudo relevance feedback in the case of the Taylor formula based method.

6 Concluding Remarks

In this paper, results of two experiments on relevance feedback have been reported. The purpose of first experiment is to check performance of a new feedback method proposed by Kishida[1] (the Taylor formula based method) in a normal situation with relevance information. The result has shown that the Taylor formula based method works well. The second experiment aims at examining effectiveness of pseudo relevance feedback using the Taylor formula based method for searching a patent collection. Unfortunately, the pseudo relevance feedback did not show good performance. We need to devise a technique for selecting better terms

from top-ranked documents in the case of applying the new feedback method.

References

[1] K. Kishida. 2001. Feedback method for document retrieval using numerical values on relevance given by users. IPSJ SIG Notes Fundamental Infology, 61: 189-196. (*in Japanese*)

[2] K. Kishida. 2003. Experiment on Pseudo Relevance Feedback Method Using Taylor Formula at NTCIR-3 Patent Retrieval Task. Proceedings of the Third NTCIR Workshop on Research in Information Retrieval, Automatic Text Summarization and Question Answering, NII, Tokyo. http://research.nii.ac.jp/ntcir/

[3] J. J. Rocchio, Jr. 1971. Relevance feedback in information retrieval. in G. Salton ed., The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 313-323.

[4] G. Salton and C. Buckley. 1990. Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 41: 288-297.

[5] P. Sarinivasan. 1996. Query expansion and MEDLINE. Information Processing and Management, 32: 431-443.

[6] J. H. Lee. 1998. Combining the evidence of different relevance feedback methods for information retrieval. Information Processing and Management, 34: 681-691.

[7] R. Mandala, T. Tokunaga and H. Tanaka. 2000. Query expansion using heterogeneous thesauri. Information Processing and Management, 36: 361-378.

[8] M. Iwayama. 2000. Relevance feedback with a small number of relevance judgments: incremental relevance feedback vs. Document clustering. in Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, 10-16.

[9] G. Ciocca. and R. Schettini.1999. A relevance feedback mechanism for content-based image retrieval. Information Processing and Management, 35: 605-632.

[10] M. F. Moens and J. Dumortier. 2000. Text categorization: the assignment of subject descriptors to magazine articles. Information Processing and Management, 36: 841-861.

[11] C. Buckley, J. Allan, and G. Salton. 1994. Automatic routing and ad-hoc retrieval using SMART: TREC2. in D.K. Harman ed., The Second Text Retrieval Conference (TREC2). National Institute of Standards and Technology, Gaithersburg MD, 45-55.

[12] S. E. Robertson, et al. 1995. Okapi at TERC-3. in D.K. Harman ed. Overview of the Third Text Retrieval Conference (TREC-3). National Institute of Standards and Technology, Gaithersburg MD, 109-126.

[13] D. A. Harville. 1997. Matrix Algebra from a Statistician's Perspective. Springer, New York.

[14] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka and Masayuki Asahara. 2000. Morphological Analysis System ChaSen version 2.2.1 Manual. http://chasen.aist-nara.ac.jp/

Appendix. Detail of Calculation

If we assume a linear function (11),

$$\frac{\partial f_X(\mathbf{b})}{\partial \mathbf{b}^T} = \frac{\partial (\mathbf{A}_X \mathbf{b})}{\partial \mathbf{b}^T} = \mathbf{A}_X ,$$

which is a well-known result in the field of linear algebra [13]. Therefore (13) becomes that

$$f_X(\tilde{\mathbf{b}}) = f_X(\mathbf{b}) + \mathbf{A}_X(\tilde{\mathbf{b}} - \mathbf{b})$$

(it should be noted that $K = 0$).

By following our assumption that $\mathbf{r}_X$ is equal to $f_X(\tilde{\mathbf{b}})$ and noting that $f_X(\mathbf{b}) = \mathbf{s}_X$, we obtain that

$$\mathbf{A}_X(\tilde{\mathbf{b}} - \mathbf{b}) = \mathbf{r}_X - \mathbf{s}_X . \qquad \text{(A.1)}$$

The (14) is easily derived from (A.1).

By using singular value decomposition we can obtain that $\mathbf{A}_X^T = \mathbf{U}\Lambda\mathbf{V}^T$. The transposition is that

$$\mathbf{A}_X = \left(\mathbf{A}_X^T\right)^T = \left(\mathbf{U}\Lambda\mathbf{V}^T\right)^T = \mathbf{V}\Lambda\mathbf{U}^T , \qquad \text{(A.2)}$$

because $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrixes and Λ is a diagonal matrix. Substituting (A.2) into (A.1), we finally obtain that

$$\mathbf{V}\Lambda\mathbf{U}^T(\tilde{\mathbf{b}} - \mathbf{b}) = \mathbf{r}_X - \mathbf{s}_X \cdot$$

$$\therefore \tilde{\mathbf{b}} = \mathbf{b} + \mathbf{U}\Lambda^{-1}\mathbf{V}^T(\mathbf{r}_X - \mathbf{s}_X) .$$