R^2D^2 at NTCIR 2 Ad-hoc Task: Relevance-based Superimposition Model for IR

Teruhito KANAZAWA
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
tkana@nii.ac.jp

Atsuhiro TAKASU, Jun ADACHI
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
{takasu, adachi}@nii.ac.jp

Abstract

This paper describes our evaluation experiments for NTCIR 2 ad-hoc task. We developed a retrieval system using the Relevance-based Superimposition (RS) model, in which document vectors are modified based on the relevance of the documents. The major focus of this year is on combination of the RS model and query expansion (QE). We submitted fully automatic ad-hoc results brought by different parameter settings.

Keywords: *NTCIR, information retrieval, vector space model, document vector modification, RS model, query expansion*

1. Introduction

We have proposed a method named the Relevance-based Superimposition model, in which document vectors are modified based on the relevance of the documents. In this model, relevant documents are organized into document sets when the index table is created and supplementary index terms are chosen for each document set.

For evaluation, we developed a retrieval system using the RS model, named R^2D^2(RetRieval system for Digital Documents), which is a full-text retrieval system designed based on the vector space model. Figure 1 depicts the process flow of R^2D^2.

The RS model shows better retrieval effectiveness by solving the semantic ambiguity caused by variance of expression among the documents. This ambiguity is a serious problem especially in retrieval from scientific papers written by various authors. On the other hand, query expansion has been proposed as one method of solving the semantic ambiguity of queries. At NTCIR 2, we focused on combination of the RS model and query expansion. This combination is expected to improve the effectiveness complementarily.

2. System Overview

R^2D^2 is designed as a full-text retrieval system based on the vector space model [1].

Formal definition of the vector space model is the following. The query $\boldsymbol{Q}$ consists of searching terms $\{q_1, q_2, ..., q_m\}$.

The similarity between the query $\boldsymbol{Q}$ and the document d_j is defined as follows:

$$S(\boldsymbol{Q}, \mathrm{d}_j) = \sum_{i=1}^{m} w(i, j, \boldsymbol{Q}), \tag{1}$$

$$w(i, j, \boldsymbol{Q}) \equiv f_T(j, i) \cdot f_D(i) \cdot f_C(i, \boldsymbol{Q}). \tag{2}$$

- f_T: factor based on the term frequency in a document.
- f_D: factor based on the document frequency containing the term.
- f_C: factor based on the term cooccurrence statistics.

2.1. Parsing

We employ ChaSen 1.51[2] as the Japanese morpheme analyzing program, for extracting and stemming terms. Hereafter, terms extracted from documents are called ‘index terms,’ and those extracted from queries are called ‘searching terms.’ Index terms are extracted from the titles, abstracts and free keywords given by the authors of papers. Their SGML tags in the NTCIR corpus are `TITL`, `ABST` and `KYWD` respectively. We use only the Japanese portions of

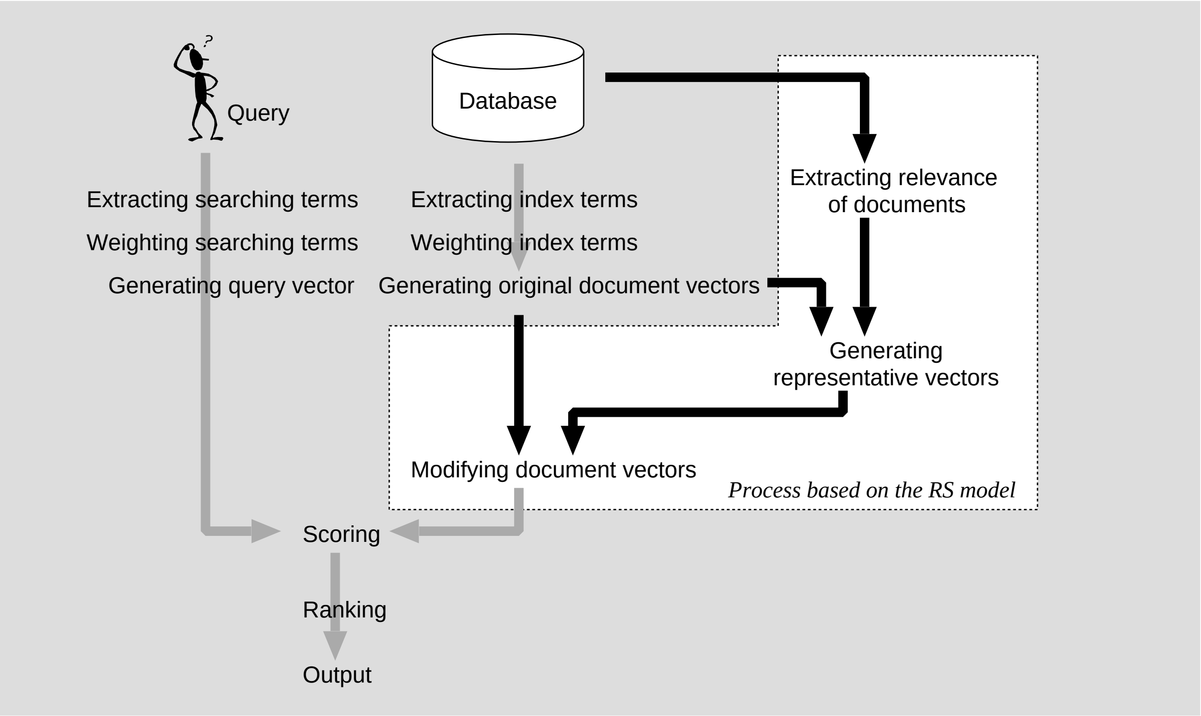
?
Query
Database
Extracting searching terms
Weighting searching terms
Generating query vector
Extracting index terms
Weighting index terms
Generating original document vectors
Extracting relevance of documents
Generating representative vectors
Modifying document vectors
Process based on the RS model
Scoring
Ranking
Output

Figure 1. The process flow of R^2D^2

records and extracted about 530,000 kind of terms from the corpus. On the other hand, we eliminated meaningless phrases such as 'I want to retrieve the papers describing ...' from queries automatically using heuristic rules.

2.2. Factor based on term frequency

We have evaluated three kinds of functions that are based on the concepts of term frequency:

$$f_{Ta} = tf_{j,i}, \quad (3)$$

$$f_{Tb} = 1 + (\log(tf_{j,i})) \cdot \log(N/df_i) \text{ and} \quad (4)$$

$$f_{Tc} = \frac{1}{\pi}\arctan(tf_{j,i}) + 0.5. \quad (5)$$

Equation (5) was identified as the most effective method in the preliminary experiment [3, 4].

Figure 2 (a) and (b) illustrates the difference between Equations 3 and 5. The factor given by Equation (5) is bounded within 1.0, while the factor given by the conventional Equation (3) grows proportionally as *tf* grows. We think that term frequency is not so important when the documents are rather short, as the NTCIR documents are. It can be generally said that documents that contain all search terms are more desirable than those documents that contain only a few of the specified terms. Thus, the conventional Equation (3) is not suitable for our purposes from this viewpoint.

Furthermore, we considered normalization using document length. Equation 5 can be generalized to:

$$f_{Tc'} = \frac{1}{\pi}\arctan\left(\alpha\frac{tf_{j,i}}{F(j)} + \beta\right) + 0.5. \quad (6)$$

We optimized the parameters and the function with TREC 3 and NTCIR 1 corpus, and used $\alpha = 100$, $\beta = -0.5$, $F(j) = \sum_i tf_{j,i}$.

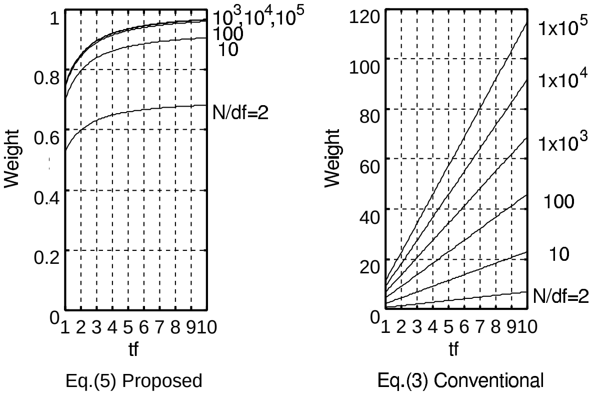

Figure 2. Term-weighting functions

2.3. Factor based on document frequency

We evaluated two factors:

$$f_{Da} = \frac{2}{\pi}\arctan(N/df_i) \quad \text{and} \tag{7}$$

$$f_{Db} = \log(N/df_i), \tag{8}$$

where N stands for the number of documents and df_i for document frequency of term t_i. Equation 8 was identified as more effective in the preliminary experiment.

2.4. Analyzing a query and weighting searching terms

R^2D^2 extracts search terms from the query using the same method as that applied to documents and weights terms by automatic relevance feedback. It is difficult to estimate the importance of search terms because a query tends not to have much information for statistical estimation. Rocchio's feedback process [5] is one effective method to weight search terms; however, there seems no assured method to tune parameters adapted to the database. We evaluated some other weighting methods in our preliminary experiments and used one described below in R^2D^2.

The weight $f_C(i, \boldsymbol{Q})$ of the query term t_i is calculated based on the cooccurrence with other query terms [6]:

$$f_C(i, \boldsymbol{Q}) \equiv \sqrt{\frac{1}{df_i}\sum_{\mathrm{d}_j \in \Delta_i}(c_j - 1)^2}, \tag{9}$$

$$\text{where} \quad c_j \equiv (\#\text{of kinds of query term appearing in document } \mathrm{d}_j),$$

$$\Delta_i \equiv (\text{document set in which term } \mathrm{t}_i \text{ appears}).$$

Let us describe this function briefly. It can be said that a document that contains many query terms is likely to be relevant to the query. The Equation (9) makes the weight greater for those query terms that appear in documents containing many of the specified query terms, in other words, the weight of an important term becomes greater.

3. RS model

3.1. Model Overview

The proposed RS model is designed according to the document vector modification approach. This model partitions the documents so that the relevant documents fall into the same cluster. However, the idea is different from traditional cluster-based methods, in which the document clusters are usually exclusive. These methods assume that documents can be classified into orthogonal clusters by the frequencies of terms, but a more natural assumption allows a document to belong to several topics. This difference in assumptions will reflect on the recall of retrieval.

For example, when there are two clusters, such as 'image processing' and 'neural networks', in an exclusive clustering model, a document on 'image processing using neural networks' will belong to one or other of them. If this document is assigned to the cluster of 'image processing', we cannot retrieve it with a query about 'neural networks'. On the other hand, in the RS model, this document can belong to both clusters; hence, this problem does not occur.

Let us define the RS model more formally. In the RS model, each document is represented by a feature vector. Term frequencies are often used as the feature. Suppose that a document database contains a set of documents $\{\mathrm{d}_1, \mathrm{d}_2, \cdots, \mathrm{d}_n\}$ and their feature vectors are $\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n$.

In the RS model, documents in the database form clusters $C_1, C_2, \cdots, C_m$. Note that one document may be contained in more than one cluster in the RS model, while clusters in other methods are often mutually exclusive. Figure 3 schematically depicts an example of document clusters in the RS model. At this point, we must decide what kind of relevance we will use to make clusters. The principle of the RS model is independent of the source of relevance information, and our choice will depend on the kind of database and the types of elements in it. For instance, the following elements can be candidates for the source of relevance information:

- keywords given by the authors,
- references, hyperlinks,
- bibliographic information, such as author name, publication date, and journal title.

In our experiments with the NII Test Collection, described in this figure and the following section, we constructed the clusters based on the free keywords given by the authors of documents. Suppose that there are two keywords A and B. Then there are two clusters corresponding to A and B, respectively. Cluster C_A consists of documents that contain the keyword A and the same relationship holds for C_B and B. In the figure, the document d_1 is in cluster C_A, since it contains only the keyword A, while the document d_3 is both in C_A and in C_B because it contains both keywords A and B.

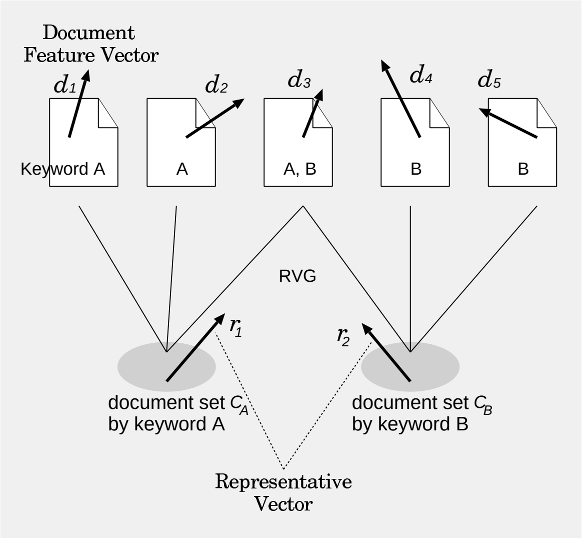

Figure 3. Representative vector generation

3.2. Representative Vector Generation

Using the clusters, the document feature vector is modified in two steps: representative vector (RV) generation and feature vector modification by RVs. The first step is to construct the RV of each cluster. The RV corresponds to the feature vector of document clusters and has the same dimension as document feature vectors. RV $\boldsymbol{r}$ of cluster C is constructed from the feature vectors of documents in C. Currently, we have proposed five kinds of representative-vector-generator (RVG) functions, based on the α-family distributions [7], which derive the ith component of RV $\boldsymbol{r}$ as follows:

[M] maximum ——— $\max\{d_{j,i}|\mathrm{d}_j \in C\}$,

[R] Root-Mean-Square — $\sqrt{(\frac{1}{|C|}\sum_{\mathrm{d}_j \in C} d_{j,i}{}^2)}$,

[A] mathematical mean —— $\frac{1}{|C|}\sum_{\mathrm{d}_j \in C} d_{j,i}$,

[S] Square-Mean-Root — $\left(\frac{1}{|C|}\sum_{\mathrm{d}_j \in C} \sqrt{d_{j,i}}\right)^2$ and

[m] minimum ——— $\min\{d_{j,i}|\mathrm{d}_j \in C\}$,

where $d_{j,i}$ stands for the ith component of the feature vector of document d_j and $|C|$ for the number of documents contained in the cluster C.

The first three functions are more 'disjunctive' and the value of the function tends to become larger when the variance of the arguments is large, so that the influence of noise appears stronger. On the other hand, the last two functions, [S] and [m] are more 'conjunctive' and the value of the function tends to become smaller when the variance of the arguments is large. If the function is too strongly conjunctive, there will be fewer supplemental terms. We must, therefore, use experiments to select and evaluate the appropriate function.

3.3. Document Vector Modification

The second step is modification of the document vector using the RVs of the clusters to which the document belongs. Figure 4 depicts this step schematically. In this case, the document vector $\boldsymbol{d}_1$ is modified using $\boldsymbol{r}_1$, because document d_1 belongs to cluster C_A, while the document vector $\boldsymbol{d}_3$ is modified using both $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$.

We assume that important index terms for a document d_j are any terms that occur frequently in any cluster to which d_j belongs, as well as terms occurring frequently in d_j itself. This characteristic is considered to be 'conjunctive'.

Currently, we propose five kinds of document-feature-vector-modifier (DVM) function. In order to define the DVM, we first define the vector of a cluster set D_j that consists of clusters to which document d_j belongs. Let S_j denote the set of RVs that belong to the clusters belonging to D_j.

Then the ith component of the vector of D_j can be defined in the following five ways:

[M] maximum ——— $\max\{r_{k,i}|\boldsymbol{r}_k \in S_j\}$,

[R] Root-Mean-Square — $\sqrt{\frac{1}{|S_j|}\sum_{\boldsymbol{r}_k \in S_j} r_{k,i}{}^2}$,

[A] mathematical mean —— $\frac{1}{|S_j|}\sum_{\boldsymbol{r}_k \in S_j} r_{k,i}$,

[S] Square-Mean-Root — $\left(\frac{1}{|S_j|}\sum_{\boldsymbol{r}_k \in S_j} \sqrt{r_{k,i}}\right)^2$ and

[m] minimum ——— $\min\{r_{k,i}|\boldsymbol{r}_k \in S_j\}$.

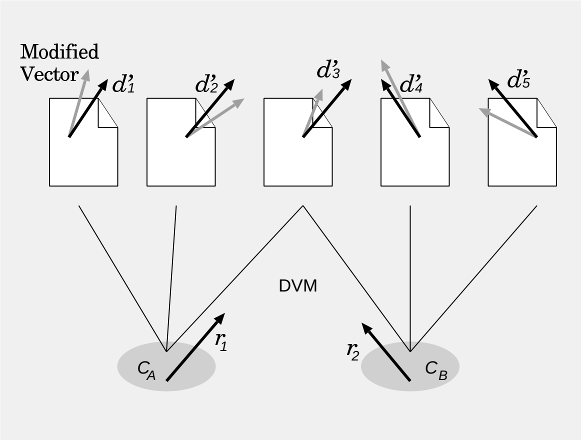

Figure 4. Document vector modification

Let $(d_{j,1}, d_{j,2}, \cdots, d_{j,I})$ represent the feature vector of a document d_j and let $(s_{j,1}, s_{j,2}, \cdots, s_{j,I})$ represent the vector of the cluster set D_j. The modified document feature vector $\boldsymbol{d}'_j$ is then defined as $(f_s(d_{j,1}, s_{j,1}), f_s(d_{j,2}, s_{j,2}), \cdots, f_s(d_{j,I}, s_{j,I}))$, where f_s is the superimposing function. We have evaluated some members of the α-family of distributions for f_s and 'maximum' is identified as the most effective method.

4. Query Expansion

We employ an automatic QE method via relevance feedback. Expansion terms are chosen from the top D document retrieved using the original query. T, the number of expansion terms is adjusted in our preliminary experiment with NTCIR 1 and TREC.

We found that most effective parameters are $D = 30, T = 10$ both for NTCIR 1 and TREC 3 SJM corpus, and D has much influence on the retrieval effectiveness. Then we examined the improvements with NTCIR 2 obtained by QE when D changed.

5. Results

There are 851,218 keywords and 90,761 of them appear in more than four documents. We do not use smaller clusters containing less than five documents since they tend to cause errors.

Table 2 shows the effectiveness of the RS model & QE. The average precision without RS nor QE is 0.2841. QE improves results by 2%, and RS does by 6%. The most effective QE parameters are $D = 40, T = 10$, however, the difference between the average precisions using $D = 40$ (the best for NTCIR 2) and $D = 30$ (the best for NTCIR 1 and TREC 3) is only 0.3%.

The combination of RS and QE achieves 9% improvement that is more than the summation of their individual effectiveness.

6. Discussion

The factor f_C based on the term cooccurrence statistics works well for most of queries, however, it fails to find important terms in some cases. For example, when it analyzes topic 118 "*Distance education support systems using TV conferencing*," it weights '*distance*' and '*TV*' twice as much as '*education*.' This estimation does not correspond to the intension.

Query 101 "*Development of hepatitis B vaccines by genetic engineering techniques*" is one of the inappropriate cases for RS. The parser does not distinguish '*B*' and '*non-B*', hence documents about "*hepatitis non-A non-B*" are recognized by the system as relevant to the query. It makes the RS model modify the feature vectors of documents containing keyword "*hepatitis non-A non-B*" improperly, and documents about hepatitis non-A non-B are retrieved. From this, we noticed the importance to treat suffixes and postfixes more carefully. We need to investigate phrasal indexing.

Query 112 "*I want papers about discharges induced by high power CO_2 lasers*" is another inappropreate case. The most relevant document cluster to the query, in other words, the cluster whose feature vector is the nearest to the query vector, is one of the documents containing keyword "*laser induced lighting*." 19 of 37 documents in this cluster are relevant to the query, hence this cluster has positive effect to improve the retrieval precision. On the other hand, the 3rd most relevant cluster is of documents containing keyword "*CO_2 laser*" that is broader concept than "*laser induced lighting*" and not reflect the content of some documents in the cluster. Only 6 of 62 documents are relevant, and it may behave as a source of noise. It is possible to screen out noisy clusters by comparing the concreteness of supplemental terms with keywords. For such a technique, a method to statistically extract a concept hierarchy is required, because no thesaurus can cover all keywords or all index terms.

Query 145 "*Papers that discuss how the locations of public libraries affect their use*," is a inappropreate case for QE. Expansion terms are '*come to the library*', '*books*', ' *building*', and so on. All of them are related to '*library*' but not to '*location*.' Then documents discussing about library generally, which are not relevant to the query, are retrieved, and it degrades the precisions of retrieval. This is the drawback of automatic QE. Interactive feedback can solve this problem. We think that the RS model makes interactive relevance feedback easy by outputting keywords as interim results and requiring the user to select appropriate keywords.

The combination of the RS model and QE seems to work well. Those two methods improve the retrieval effectiveness complementarily, in other words, QE refines the query and that enhances the effectiveness of the RS model. In some cases, the combination achieves larger gain on effectiveness than the summation of ones of the RS and QE (ex. for Q.139, the combination achieves 247% higher average precision than one of the baseline, while the RS model does +73% and QE does +33%). In some other cases, failure of one method is covered by the other (ex. for Q.101, the combination gives 8% higher precision, while the RS does 12% lower).

7. Conclusion

In this paper, we showed the effectiveness of the proposed RS model and of the combination with QE. The RS model achieves 6% superiority over the base-

Table 1. Improvements obtained by query expansion

	NTCIR 1	TREC 3 SJM
baseline	.3059	.2318
QE	.3270 (+7%)	.2578(+11%)

Table 2. Performance of the RS model and query expansion

Run	QE	RS	avg prec	R-prec	P@10 docs	P@100 docs	# q ≥ avg+0.05	# q ≤ avg−0.05
——	no	no	.2841	.3147	.5510	.1996	16	7
——	yes	no	.2886 (+2%)	.3282	.5551	.2120	18	5
R2D21	no	yes	.3020 (+6%)	.3353	.5571	.2165	22	4
R2D27	yes	yes	.3103 (+9%)	.3402	.5653	.2310	21	5

Table 3. Average precisions for some notable queries

	baseline	QE	RS	RS+QE
Positive cases for RS				
Q.115: I want papers about videostreaming techniques.				
	.1271	.1216 (−4%)	.1527 (+20%)	.1805 (+42%)
Q.139: Documents will report on the “ sick building syndrome,” which includes an allergic reaction to chemicals such as formaldehyde.				
	.3105	.4123 (+33%)	.5382 (+73%)	.7663(+247%)
Q.143: Papers that mention visually impaired persons’ use of library computer terminals or information retrieval systems				
	.4389	.4868 (+11%)	.5167 (+18%)	.5256 (+20%)
Inappropriate case for RS				
Q.101: Development of hepatitis B vaccines by genetic engineering techniques				
	.4299	.4645 (+8%)	.3801 (−12%)	.4647 (+8%)
Positive cases for QE				
Q.112: I want papers about discharges induced by high power CO_2 lasers.				
	.3912	.4119 (+5%)	.2870 (−27%)	.4215(+8%)
Q.125: I would like to learn about the antimicrobial activity of electrolytic acid water.				
	.5391	.5957 (+10%)	.5720 (+6%)	.6466 (+20%)
Inappropriate cases for QE				
Q.128: Are there any documents about coagulase-negative Staphylococci that cause infectious diseases?				
	.6070	.3771 (−38%)	.5975 (−2%)	.4069(−32%)
Q.145: Papers that discuss how the locations of public libraries affect their use				
	.4373	.2997 (−32%)	.4166 (−5%)	.2393 (−45%)

line, and the combination of RS and QE achieves 9% improvement. Those two methods improve the retrieval effectiveness complementarily.

For the future work, it is necessary to consider circumstances where databases are used for which keywords are not given. We plan to investigate automatic keyword extraction.

Acknowledgment

This research is supported by the ‘Research for the Future’ Program JSPS-RFTF96P00602 of the Japan Society for the Promotion of Science.

References

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

[2] “Japanese Morphological Analyzer ‘ChaSen’*(in Japanese)* , `http://cactus.aist-nara.ac.jp/lab/nlt/chasen.html`.

[3] T. Kanazawa, "An Information Retrieval Method using a Relevance-based Superimposition Model, Master's thesis, Graduate School of Engineering, University of Tokyo, 1999.

[4] T. Kanazawa, A. Takasu, and J. Adachi, "Effect of the Relevance-based Superimposition Model on Information Retrieval," *IPSJ Database workshop 2000 (IPSJ SIG Notes)*, Vol. 2000, No.69, 2000-DBS-122, pp. 57–64, Iwate, July 2000.

[5] C. Buckley, A. Singhal, and M. Mitra, "Using Query Zoning and Correlation Within SMART: TREC 5," *Proc. TREC 5*, Gaithersburg, MD, 1996.

[6] T. Kanazawa, "R^2D^2at ntcir: using the relevance-based superimposition model," *NTCIR Workshop 1 Proc.*, pp. 83–88, Tokyo, Aug. 1999.

[7] Y. Hayashi, "On a New Data Model suitable fpr Intellectual Accesses by Personal Preference," *IPSJ database workshop '98 (IPSJ SIG Notes)*, Vol. 98, No.58, 98-DBS-116(2), pp. 381–388, July 1998.