R^2D^2 at NTCIR: Using the Relevance-based Superimposition Model

Teruhito KANAZAWA
Graduate School of Engineering, University of Tokyo

7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, JAPAN
TEL: +81–3–3812–2111 E-mail: tkana@kyagroup.com

Abstract

Our information retrieval project submitted fully automatic ad-hoc results. We use only *description* fields as queries. R2D21 is the baseline tf·idf result, and R2D22 is the result using the proposed RS model which expands document vectors based on the relevance of documents. This method is expected to show better retrieval effectiveness than conventional methods, such as query expansion. The RS run R2D22 achieved about 12% improvement of precision over the baseline tf·idf run R2D21.

Keywords
information retrieval, vector space model, document vector expansion, RS model, relevance of documents, NTCIR

1 Introduction

We have proposed a method named the Relevance-based Superimposition (RS) model, in which document vectors are expanded based on the relevance of the documents.

For NTCIR, 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 (VSM). Figure 1 depicts the process flow of R^2D^2.

We tried to find the term-weighting methods suitable for the NTCIR test collection using the preliminary query topics (IDs are 1 to 30). In the evaluative runs (query topic IDs are 31 to 83), we used tf·idf-based weighting method for index terms, and cooccurrence-based method for searching terms. We submitted fully automatic retrieval results using the *description* fields of query topics.

2 Parsing documents and queries

We employ ANIMA 1β 3.1a [1] as the Japanese morpheme analyzing program, for extracting and stemming terms. We use the nouns, verbs, adjectives, and adverbs as terms, and use the original forms of verbs, adjectives, and adverbs. ANIMA is originally designed not to divide unknown expression of *katakana*, hence we changed this rule in order to extract term from the expression 'unknown *katakana* term + known *katakana* term'. ANIMA has no dictionary of its own, and we use the one of ChaSen 1.51[2], who contains about 110,000 words.

Hereafter, terms extracted from documents are called 'index terms', and those extracted from queries are called 'searching terms'.

Index terms in R^2D^2 are extracted from the titles, abstracts, and free keywords given by authors of papers, whose SGML tags in the corpus are TITL, ABST and KYWD. We use only the Japanese portions of records.

On the other hand, we regard such expressions as 'I want to retrieve the papers describing ...' in queries meaningless for retrieval. We eliminated those expressions automatically using heuristic rules.

3 Weighting index terms

We first have evaluated three kinds of term-weighting methods, which generates document feature vectors based on the concept of tf·idf.

$$d_{j,i} \equiv tf_{j,i} \cdot \log(N/df_i) \tag{1}$$

$$d_{j,i} \equiv 1 + (\log(tf_{j,i})) \cdot \log(N/df_i) \tag{2}$$

$$d_{j,i} \equiv \left(\frac{1}{\pi}\arctan(tf_{j,i}) + 0.5\right) \frac{2}{\pi}\arctan(N/df_i) \tag{3}$$

Equation (1) is the conventional method in which the importance of the term is proportioned to its tf. We think that term frequency is not so much important when the documents are rather short as the NTCIR documents. It can be generally said that the documents which contain all searching terms are more desirable than those documents that contains only a few of all the specified terms. Thus, (1) is not suitable for our purpose from this viewpoint.

Equation (2) is used in the SMART[3], and it makes lighter of tf than (1). And Equation (3) is the most effective method in our preliminary experiment in which we evaluated those three weighting methods of index terms as is shown in Table 1. The normalized precision of (3) is about 8% higher than the one of conventional (1), hence we adopted the method (3) to the evaluative runs.

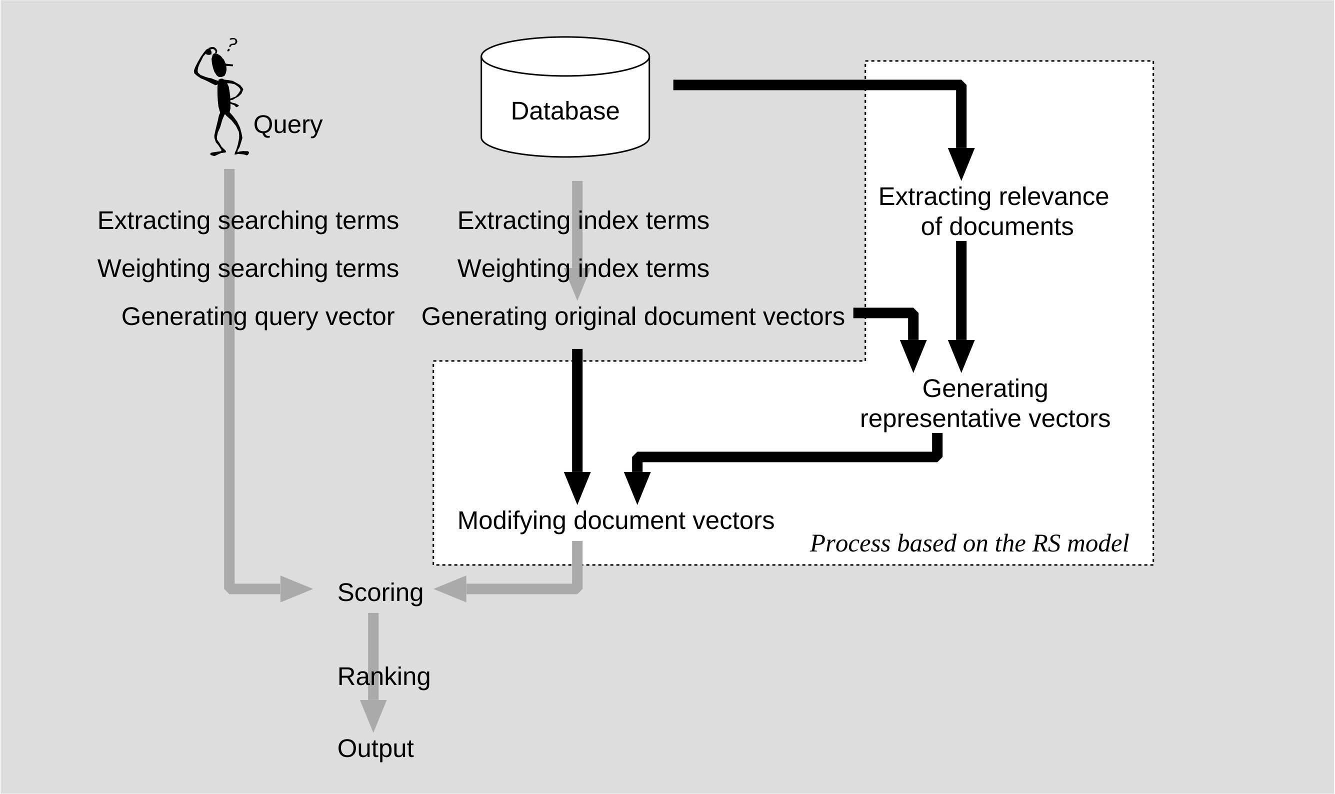

Query
Database
Extracting searching terms
Weighting searching terms
Generating query vector
Extracting index terms
Weighting index terms
Generating original document vectors
Extracting relevance of documents
Generating representative vectors
Modifying document vectors
Process based on the RS model
Scoring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

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

Table 1: preliminary experiment 1: Weighting methods of index terms

methods		Norm.Prec.	
(1)	Conventional	.3498	
(2)	SMART	.3161	(−10%)
(3)	R^2D^2	**.3772**	(+8%)

Table 2: preliminary experiment 2: Weighting methods of searching terms

methods	Norm.Prec.	
weights ignored	.3474	
R^2D^2	**.3772**	(+9%)

4 Analyzing queries and weighting searching terms

It is difficult to estimate the importance of searching term because a query tends not to have much information for statistical estimation. Rocchio's feedback process[3] is one of the effective methods to weight searching terms, however, there seems no assured method to tune parameters adapted to the database. We evaluated some other weighting methods in the preliminary experiments, and used one described below in R^2D^2.

$$q_i \equiv \sqrt{\frac{1}{\#\text{ docs in }\mathcal{D}} \sum_{d_j \in \mathcal{D}} (f(d_j) - 1)^2} \qquad (4)$$

$$\mathcal{D} \text{ is a set of docs consists of term } i. \qquad (5)$$

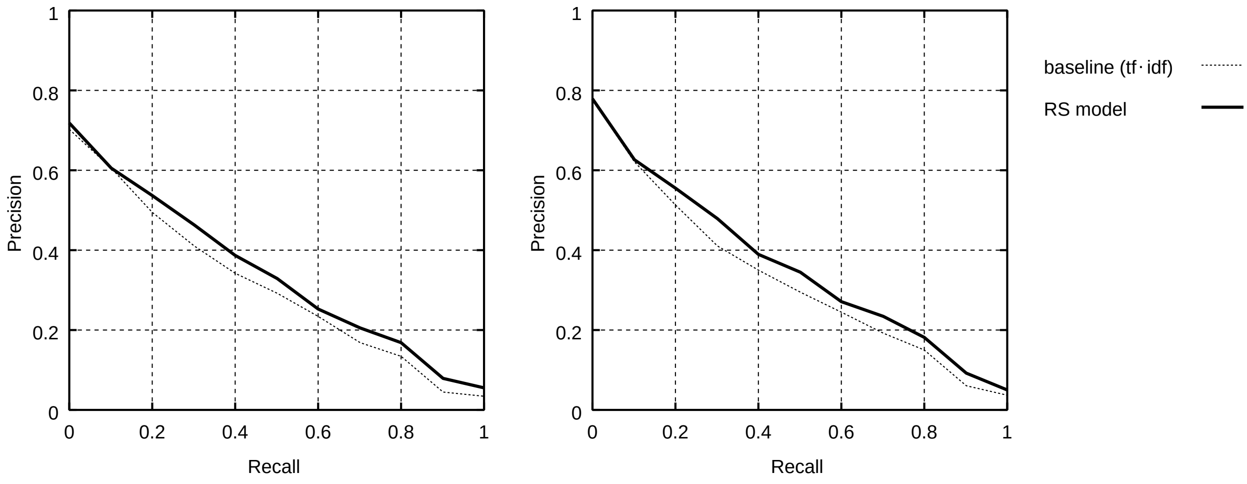
$$f(d_j) \equiv (\#\text{ kinds of term appearing in doc. } d_j) \qquad (6)$$

Table 2 shows the result of our preliminary experiment in which we evaluated the weighting method of searching terms. It achieved about 9% improvement of the normalized precision.

5 RS model

We are interested in coping with the problem of semantic ambiguity[4] in information retrieval (IR) systems. It might be difficult to recognize user's intention precisely from the query which usually provides only a restricted notation. This can degrade the effectiveness of retrieval.

Much work has been done on this problem, and these studies are categorized into three groups: query modifica-



(a) rank A

(b) rank A+B

Figure 2: Precision of the RS model and baseline tf·idf

tion [3], document space modification[5, 6], and document feature modification.

We think that document feature modification achieves higher recall without losing precision of retrieval, because document usually have much more information than a query.

The proposed RS model is designed according to the document feature modification approach that analyzes the topics of relevant document sets. This model partitions the documents so that the relevant documents fall into the same cluster.

Let us define the RS model more formally. 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 $\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n$, which are made by the method described the former section.

In the RS model, documents in the database form clusters $C_1, C_2, \cdots, C_m$. Note that clusters are mutually exclusive in cluster-based information retrieval while a document may be contained in more than one cluster in the RS model.

At this point, we must decide what kind of relevance we will use to make clusters. In R^2D^2, we use the keywords given by authors as our source of relevance information. We assume that documents that have the same keyword are relevant to each other. There are 376,503 keywords, and 77,996 of them appear in more than one document.

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 has the same dimension as document feature vectors, and RV $\boldsymbol{r}$ of cluster C is constructed from the feature vectors of documents in C. We have evaluated several kinds of representative-vector-generator (RVG) functions, and we found that the most effective RVG is Root-Mean-Square, which derive the i-th component of RV $\boldsymbol{r}$ as follows:

$$\sqrt{\frac{1}{|C|}\sum_{\mathrm{d}_j\in C}(d_{j,i})^2}, \tag{7}$$

where $d_{j,i}$ stands for the i-th component of the feature vector of document d_j.

The second step is modification of the document vector using the RVs of the clusters to which the document belongs.

We have evaluated several kinds of document-feature-vector-modifier (DVM) function, and found that the most effective DVM is Root-Mean-Square. 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 which belong to the clusters belonging to D_j.

Then the i-th component of the vector of D_j is defined as:

$$\sqrt{\frac{1}{|S_j|}\sum_{\boldsymbol{r}_k\in S_j}(r_{k,i})^2}, \tag{8}$$

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. Then, the modified document feature vector $\boldsymbol{d}'_j$ is defined as $(\max\{d_{j,1}, s_{j,1}\}, \max\{d_{j,2}, s_{j,2}\}, \cdots, \max\{d_{j,I}, s_{j,I}\})$.

Then R^2D^2 ranks all documents in the database. The score of document d_j is calculated as the inner product of the query vector and the modified vector $\boldsymbol{d}'_j$, while the baseline results of tf·idf are made with the inner product of the query vector and the original vector $\boldsymbol{d}_j$.

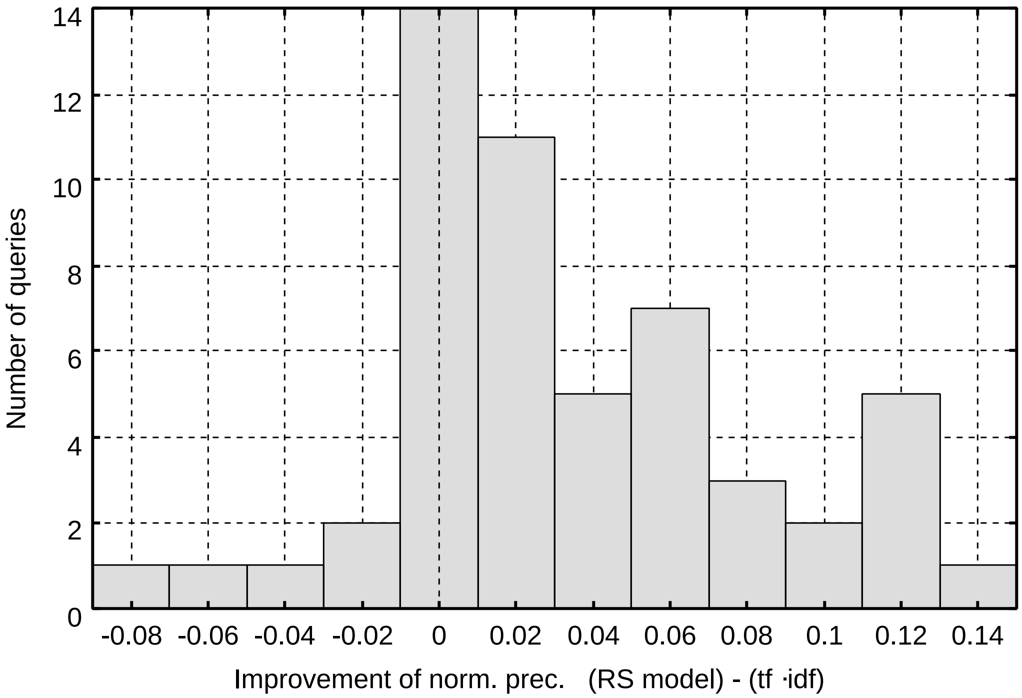


Figure 3: Histogram of precision improvement over tf·idf per query

Table 3: Normalized precision

	tf·idf (R2D21)	RS model (R2D22)
rank A	.2933	**.3289** (+.0356, 12%)
rank A+B	.3097	**.3473** (+.0376, 12%)

6 Results and discussion

We show three types of evaluation statistics.

- 11-point-average precision —— the averages of the precision of 53 queries at 11 points whose recalls are {0.0, 0.1, ..., 1.0}.

  Figure 2 shows 11-point-average precision values of baseline tf·idf and the RS model.

- Normalized precision values —— $\frac{1}{N}\sum_{i=0}^{N} P_i$.

  Table 3 shows the normalized precision values, and Figure 3 shows the histogram of the precision improvement over tf·idf per query.

- Difference of normalized precision value for each query

  Figure 4, 5 shows the differences of normalized precision value for each query.

Table 3 shows that the normalized precision of the RS model is about 12% higher than the baseline tf·idf both in rank A and in rank A+B evaluations.

Figure 2 shows that precision improved overall. This shows that the precision values of the highly-ranked documents are not improved as much as the rest. This result can be interpreted that the highly-ranked non-relevant documents have a problem other than semantic ambiguity. We guess one such problem might be structural ambiguity.

Table 4 shows some typical examples to explain the improvements achieved by the RS model. The query topic #34 requests papers describing adaptation methods of TCP to the wireless communication. In some documents, ‘mobile’ or ‘wireless channel’ is used for ‘wireless communication’, and ‘adaptation’ is expressed by various expressions. Each document listed in the Table 4 has only 3 searching terms, however, they belong to the cluster of ‘mobile computing’, ’ARQ’, and/or ‘retransmission scheme’, and they are supplemented the expected terms ‘wireless’, ‘improve’, and/or ‘adapt’.

On the other hand, Table 5 shows the examples of inappropriate supplementary terms. Generally, it can be said that leptomycin is closely connected to G1-phrase. However, the document # 81832 is not related to G1-phrase, though it has the keyword ‘Leptomycin’. This problem might be caused by the general limitation of the pure statistical method.

7 Conclusion

It is generally pointed out that automatic query expansion sometimes deteriorate the effectiveness of retrieval. Its con-

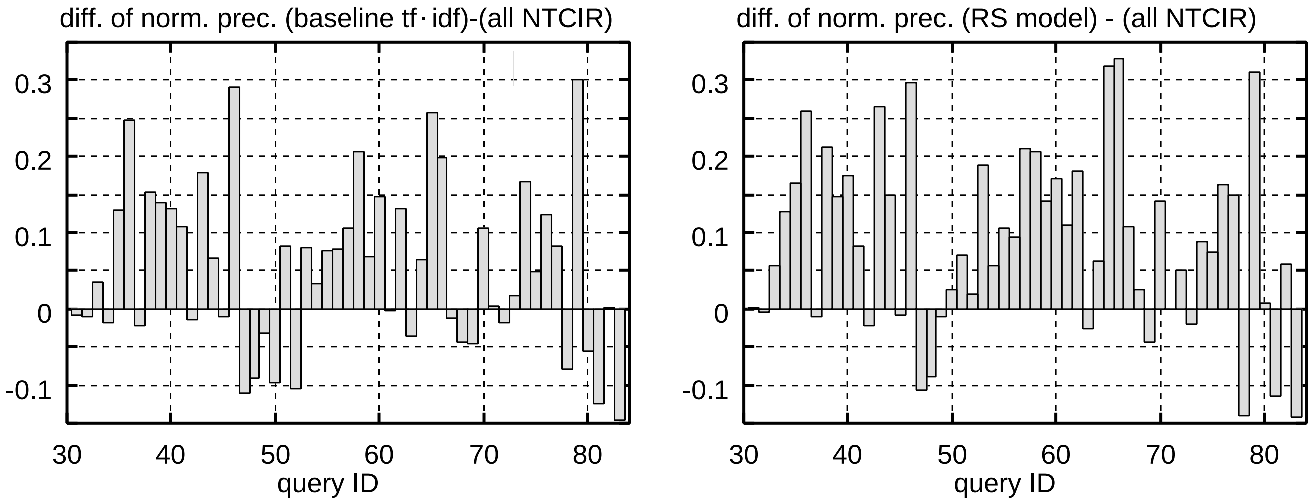


Figure 4: Difference of Precision for each query (rank A)

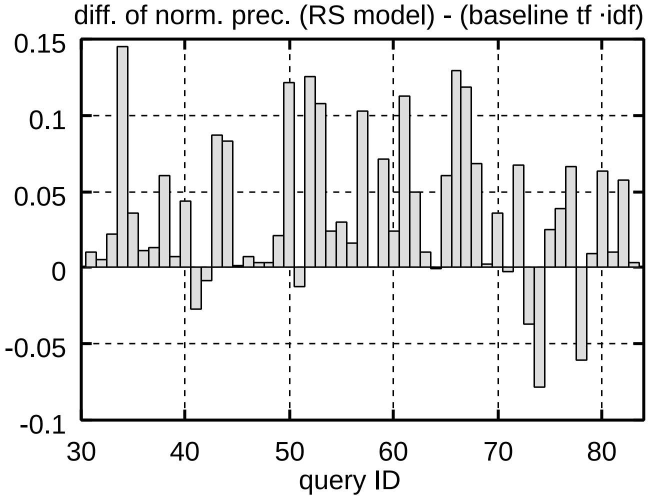


Figure 5: Difference of Precision for each query (rank A)

tribution depends on queries. On the other hand, the RS model improves overall performance by overcoming those drawbacks. Comparing the normalized precision of each query, the RS model improves by more than 0.05 for 18 queries, while it is below 0.05 for only 2 queries.

These results indicate that the RS model could expand document vectors more precisely, using the relevance of the documents.

To get more information from keywords, we consider applying the keyword classifying technique proposed by Aizawa *et al.*, which applies graph theory to the network of Japanese-English keyword pairs[7]. It is not possible to use general thesauri because authors of scientific papers tend to invent new terms.

Furthermore, it is necessary to consider general circumstances where databases are used for which keywords are not given. We plan to investigate two approaches: one is automatic keyword extraction, and the other is to give another source of relevance information. In both, the characteristics of document sets might be changed, and this might influence retrieval effectiveness.

In an actual retrieval process, users tend to repeat inputs of queries such as selecting and adding terms with relevance feedback of outputs from the system such as the display

Table 4: Examples of improvements by the RS model

Query Topic #34: Are there documents which describe improving methods of TCP for adapting to the wireless communication control?
Searching terms: tcp, wireless, communication, control, adapt, improve, method.
Rank and # terms are 'baseline → RS'.

ACCN	title	rank	# terms in the doc. vec.
329892	Novel harmonized retransmission scheme with TCP for wireless data communication systems.	41 → 10	3 → 7
312324	A TCP Packet Transmission Control Method over Wireless Channel	79 → 13	3 → 7
319233	TCP/IP suitable for Asymmetric Mobile Link	47 → 20	3 → 7

Table 5: Examples of inappropriate supplementary terms

Query Topic #74: Documents about factors and/or genes, which work on the cell cycle control in the G1-phase, and which are derived from yeast.
Searching terms: G1, phase, cell, cycle, control, work, factor, gene, yeast, derive.

ACCN	title	keyword	incorrect supplementary term
81832	Inhibition and uncoupling of the eukaryotic cell cycle caused by microbial metabolites	Leptomycin	G1-phase
308688	Checkpoint regulation of the cell cycle	Cell cycle	yeast

of candidate documents and terms. Thus, the interface is very important, and interim results should be available for users to arrange easily. We think that the RS model with document sets of keywords makes relevance feedback easier by outputting keywords as interim results and requiring the user to select appropriate keywords.

Acknowledgment

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

We participate in the NACSIS Collection Japanese Text Retrieval Workshop, and use the NACSIS Test Collection 1, constructed by the NACSIS R&D Dept. The collection contains summaries of papers presented at conferences hosted by academic societies described at `http://www.rd.nacsis.ac.jp/~ntcadm/acknowledge/thanks1-ja.html`.

References

[1] Sakurai, H., Hisamitsu, T., "Design and Evaluation of the Japanese Morphological Analyzer 'ANIMA'," *54th Annual Conventions of IPSJ*, March 1997. *(Japanese)*

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

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

[4] Earl, L.L., "Use of Word Government in Resolving Syntactic and Semantic Ambiguity," *Information Storage and Retrieval*, Vol.9, No.12, pp.639 – 664, 1973.

[5] Deerwester, S., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A., "Indexing by Latent Semantic Analysis," *J. American Society for Information Science*, Vol.41, No.6, pp.391–407, 1990.

[6] Miyahara, T., Kiyoki, Y., Kitagawa, T., "A Fast Algorithm and Its Implementation Method for Semantic Associative Search by a Mathematical Model of Meaning," *IPSJ Trans.*, Vol.38, No.7, pp.1399 – 1411, Jul. 1997. *(Japanese)*

[7] Aizawa, A. and Kageura, K, "An Approach to the Automatic Generation of Multilingual Keyword Clusters," *Proc. Compterm'98 (the First Workshop on Computational Terminology)*, pp.8 – 14, Aug. 1998, Montreal, Canada.