PERGAMON

Information Processing and Management 37 (2001) 1–14

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Re-ranking model based on document clusters

Kyung-Soon Lee[a,*], Young-Chan Park[b], Key-Sun Choi[a]

[a] *Division of Computer Science, Department of Electrical Engineering & Computer Science, Korea Advanced Institute of Science and Technology, KORTERM, 373-1 Kusung Yusong Taejon, 305-701 South Korea*
[b] *K4 M (Knowledge for the New Millennium), Taejon, South Korea*

Received 26 April 1999; accepted 20 February 2000

Abstract

In this paper, we describe a model of information retrieval system that is based on a document re-ranking method using document clusters. In the first step, we retrieve documents based on the inverted-file method. Next, we analyze the retrieved documents using document clusters, and re-rank them. In this step, we use static clusters and dynamic cluster view. Consequently, we can produce clusters that are tailored to characteristics of the query. We focus on the merits of the inverted-file method and cluster analysis. In other words, we retrieve documents based on the inverted-file method and analyze all terms in document based on the cluster analysis. By these two steps, we can get the retrieved results which are made by the consideration of the context of all terms in a document as well as query terms. We will show that our method achieves significant improvements over the method based on similarity search ranking alone. © 2000 Elsevier Science Ltd. All rights reserved.

Keywords: Document re-ranking; Inverted-file method; Cluster analysis; Dynamic cluster view; Combining evidence

1. Introduction

In the inverted file method, one of the traditional information retrieval models, we can often find that high-ranked documents may be far from the user's needs. The inverted file method has limitations on finding related documents since it simply checks the existence of query terms in documents without considering the context of documents.

* Corresponding author. Tel.: +82-42-869-5565; fax: +82-42-867-3565.
E-mail address: kslee@world.kaist.ac.kr (K.S. Lee).

PII: S0306-4573(00)00017-0

In the information retrieval field, cluster analysis is used to get retrieval results more efficiently or to classify documents into categories since this method is useful in estimating the similarities between documents. Document clustering assigns documents to automatically created clusters, based on the degree of association between documents and clusters. But, this is inappropriate to calculate the similarity of query and document since query consists of only a few terms for obtaining statistically meaningful frequency–vector (Mauldin & Carbonell, 1991). Recently, document clustering was applied to browsing and viewing of retrieval results. The Scatter/Gather system (Hearst & Pedersen, 1996) is a cluster-based document browsing method, as an alternative to ranked titles for the organization and viewing of retrieval results. Adaptive cluster-based browsing (Eguchi, 1999) method clusters search results, considering the incrementally expanded query as the user's viewpoint. The user can modify the query and then the system re-clusters the search results using the modified query.

Generally, when humans induce a relationship between query and document, he decides the relevance of document not based on existence of query terms, but based on semantics of query terms in documents. Thus, the retrieval method which intends to consider semantics of query terms have been studied such as query expansion, latent semantic indexing (LSI) and mutual information.

In the query expansion method, one retrieves more related documents by adding related words to the initial query. Other methods, which expand a query by analyzing behavioral relationship of terms in corpus or by using thesaurus, have been suggested. This query expansion can increase recall, however, operationally impractical because precision of top ranked documents can be lower (Fitzpatrick & Dent, 1997). Another complementary method in which a new query is made out of the user's relevance feedback has been studied. The result from this method entirely depends on the quality of the user's response. Therefore, recent query expansion methods analyze documents retrieved by an initial query (Buckley, Salton & Allan, 1994; Allan, 1995; Xu & Croft, 1996).

As an extension of vector space retrieval method, Latent Semantic Indexing is a concept-based retrieval method (Deerwester, Dumais & Harshman, 1990). Because dependencies between terms are explicitly taken into account in the representation and exploited in retrieval, a query can be very similar to a document even though they have no matching words.

Recent works have shown that retrieval effectiveness can be improved by using mutual information. Mutual information is a measure which represents the relation between words. Two-level document ranking using mutual information (Kang, 1997) re-evaluates the relationship between the terms of the retrieved documents and the terms of the query. This method depends on correctness of the mutual information construction.

Thus, many research efforts have been made on how to solve the keyword barrier which exists because there is no perfect correlation between matching words and intended meaning (Mauldin & Carbonell, 1991).

In this paper, we present a model of information retrieval system that is based on a document re-ranking method using document clusters. We re-evaluate the documents based on cluster analysis, with a varying number of relevant documents retrieved by an initial inverted-file method. We focus on the merits of inverted-file method and cluster analysis. In other words, we retrieve documents based on the inverted-file method and analyze all terms in documents based on the cluster analysis. In our method, the context of a document can be considered in the retrieved results by the combination of information search and cluster analysis.

2. Construction of document clusters

In many fields, clustering for multidimensional data is popular. In clustering methods, there are two kinds of method: hierarchical and nonhierarchical. Most of the researches on cluster analysis employed hierarchical methods.

The hierarchical clustering method begins with a set of single documents which is considered as a separate cluster. The two clusters that are the closest according to some similarity measure are agglomerated. This is repeated until all of the clusters belong to one hierarchically constructed cluster. The hierarchical cluster structure is called a *dendrogram* like that shown in Fig. 1.

We use *Ward's method* (Ward, 1963) which is a hierarchical clustering strategy that follows the general algorithm for the hierarchical agglomerative clustering methods. The document or cluster pair joined at each stage is the one whose merger minimizes the increase in the total within-group squared deviation about the variance. Each cluster has a cluster centroid in the form of a vector which is useful as a representative of a cluster when calculating query–document similarity. In Ward's method, the dissimilarity measure is the increase in variance. Clusters produced by Ward's method tend to be homogeneous and a symmetric (Frakes & Baeza-Yates, 1992).

To construct document clusters, we represent documents as vectors and calculate the similarities between them. We then cluster them based on the dissimilarities.

2.1. Document representation

Documents are represented by pairs of a term and its weight. The process of representation of documents is as follows:

1. do morphological analysis on each document using Korean morphological analyzer;
2. extract nouns through part-of-speech tagging using HMM tagger;
3. calculate document frequency and term frequency of each term in document;
4. assign weight to each term.

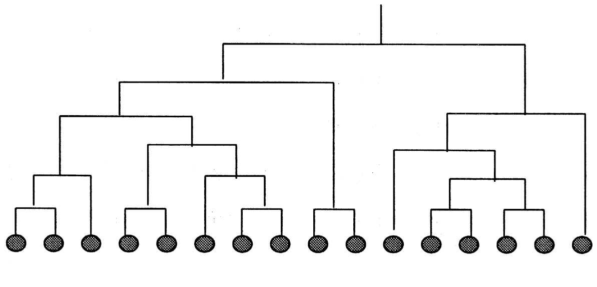


Fig. 1. Dendrogram of a hierarchical agglomerative cluster.

In constructing a term weighting scheme, three main components such as term frequency, document frequency, and normalization have been considered in the information retrieval literature. There are actual formulas for some well-known term weighting schemes (Lee & Ahn, 1996).

We use the atc weighting scheme among various weighting schemes for document weighting. The atc weighting scheme is calculated as follows:

$$\text{a: } 0.5 + 0.5 \cdot \frac{tf}{\max tf}, \quad \text{t: } \ln \frac{N}{n}, \quad \text{c:} \frac{1}{\sqrt{\Sigma_{\text{vector}} w_i^2}} \tag{1}$$

where tf is term frequency, maxtf is maximum tf in a document, N is the total number of documents in the collection, and n is the number of documents to which a term is assigned.

Since the document–document similarity depends on the weights of coinciding index terms in the two vectors, the term weighting scheme is an important factor affecting the effectiveness of document clustering.

2.2. Document clustering

We clustered documents using *Reciprocal Nearest Neighbors* algorithm (Murtagh, 1983) for Ward's method. Hierarchical agglomerative clustering produces 2N-1 clusters for N documents and their representatives. A centroid or cluster representative is a record that is used to represent the characteristics of the documents in a cluster.

In the re-ranking step, we determine dynamic clusters of retrieved documents using cluster hierarchy and calculate similarities between cluster centroids and query.

3. Document re-ranking model based on document clusters

The system architecture of document re-ranking model is shown in Fig. 2. This model is combining the inverted file method and the cluster analysis method. We construct the hierarchical document clusters depending on similarities between documents. At the first retrieval step, we retrieve documents based on the inverted-file method. At the second analysis step, we partition clusters according to the behaviors of retrieved documents from the previous step and calculate the query–cluster similarities. And we calculate new similarities to the documents from the similarities of the first step and those of the second. According to the combined similarities, we re-rank the documents in descending order.

3.1. The first retrieval based on inverted-file method

At the retrieval step, we retrieve documents based on the inverted-file method. We focus on each document at this retrieval step. The inverted-file method ranks the retrieved documents in decreasing order of query–document similarities. The query–document similarity depends on the weights of coinciding terms in the two vectors, and therefore the term weighting scheme is an important factor. We evaluated effectiveness of the re-ranking for various indexing methods of query and document weighting scheme.

In this step, we get N query–document similarities whose value is above 0.

3.2. Document re-ranking based on cluster analysis

As the degree of matching of evidences in documents is higher, the two documents are more similar. In document clustering, similar documents are classified as one cluster. Therefore, relevant documents are in the same cluster according to the *cluster hypothesis* (van Rijsbergen, 1979) which states that relevant documents tend to be more similar to each other than to non-relevant documents.

The documents in a cluster have effects on cluster centroid. The cluster centroid for a pair of clusters C_i and C_j is given by:

$$\frac{m_i C_i + m_j C_j}{m_i + m_j} \tag{2}$$

where m is the size of a cluster.

The same query–cluster similarity value is applied to all the documents in the cluster at the re-ranking stage. In this way, the documents in a cluster can affect one another through calculation of cluster centroids so that context retrieval is possible, due to the interaction of evidences contained in documents.

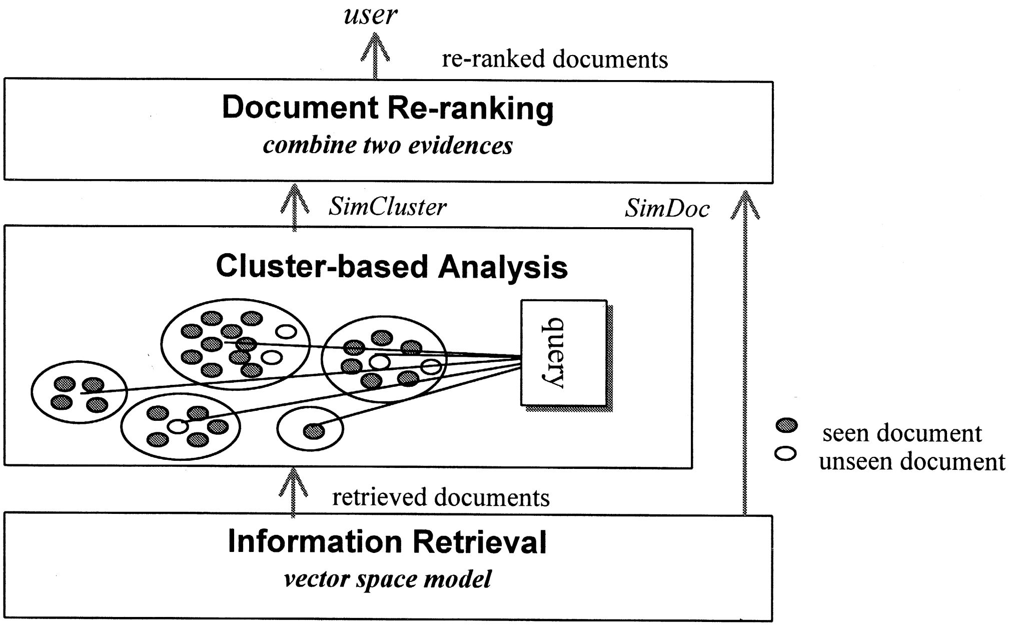


Fig. 2. System architecture of the document re-ranking model.

3.2.1. Cluster partitioning

Many research efforts have been made on how to apply clustering to get better retrieval results. Static clustering for all documents is followed by the calculation of matching degree of query and cluster centroid. In these methods, a query is compared in a top-down or bottom-up manner with each cluster of documents, which is produced hierarchically. On the contrary, the Scatter/Gather system (Hearst & Pedersen, 1996), one of the cluster-based document browsing methods, adapts dynamic clustering.

The advantages/disadvantages of static vs dynamic clustering are compared in Anick and Vaithyanathan (1997). Pre-clustering (or static clustering) has a disadvantage that cannot be matched with the user's query. However, run-time clustering (or dynamic clustering) is expensive, if dynamic clusters are based on pre-determined clusters, it can be adapted to the user's query.

According to the above observation, we use *static clustering* and *dynamic view*. That is, we apply static clustering to the set of whole documents and view clusters dynamically depending on retrieval results in the ranking. The results from the first retrieval step are documents containing terms in the query. We analyze the distribution of these documents in a cluster and partition the cluster dynamically from the viewpoint of the query.

Hierarchical agglomerative clustering produces $2N$-1 clusters and their representatives for N documents. Among these clusters, we determine how to select ones for the current query. As shown in Fig. 3, we determine whether higher clusters or lower clusters on the cluster hierarchy according to the rate of static members (S) from static clustering to dynamic members (D) from the first retrieval results. In other words, we select clusters having a minimum rate of static to dynamic cluster more than a threshold value. The clusters resulting from this step are different if the retrieved documents to a query are different. Therefore, we can produce clusters that are tailored to the characteristics of the query.

3.2.2. Calculating query–cluster similarities

Each cluster centroid is represented as a vector by pairs of a term and its weight. The

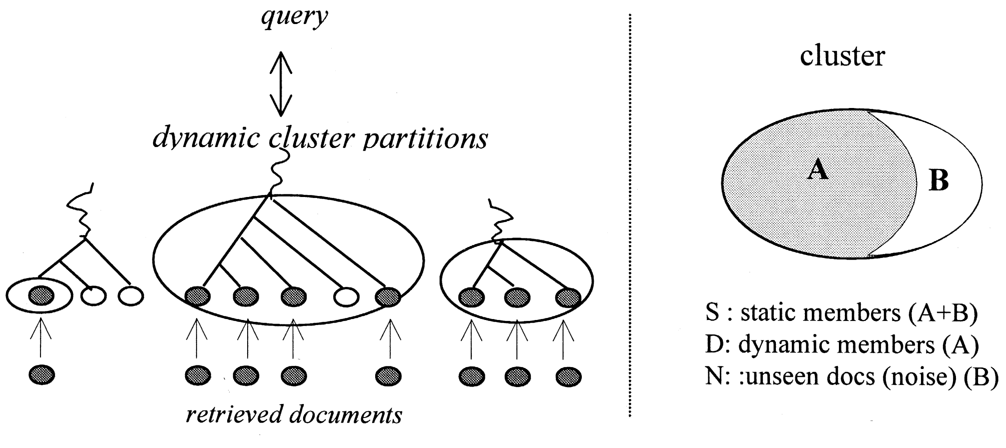


Fig. 3. Dynamic cluster view.

centroid $C(A_\alpha)$ of retrieval set A_α is defined as follows (Park, 1997):

$$C(A_\alpha) = \frac{\Sigma_{\alpha \in A_\alpha} \vec{a}}{\mid A_\alpha \mid}, \tag{3}$$

where $\mid A_\alpha \mid$ is a size of A_α and $\vec{a}$ is a vector of document a.

The results of cluster partitioning can contain the documents which are not the result of the first step. These have a negative effect on cluster centroid for a query. We adjust the value of cluster centroid to minimize the negative effect.

For each term matching query term i in cluster centroid (C_w),

$$w_i' = \frac{S \cdot w_i}{D}, \tag{4}$$

where S is the number of the static members in cluster, D is the number of the dynamic members in cluster, w_i is the weight of the current centroid, and w_i' is the new weight for centroid of the dynamic cluster.

We calculate the query–cluster similarity. This similarity of a cluster has the same effects all over the document collections in the same cluster. We focus on document collections at this step. A cluster close to the user's interest has a high similarity and a cluster far from the user's interest has a low similarity. Therefore we have a clue what assigns higher priorities to clusters close to the user's interest.

3.2.3. Combining two evidences

We calculate query–document similarity in the inverted-file retrieval and query–cluster similarity in the cluster analysis. That is, we focus on each document at the first step and on document collections at the second step.

We combine two similarities from the first retrieval and the second analysis step.

$$\mathrm{Sim}_{\mathrm{combined}} = \alpha \cdot \mathrm{Sim}_{\mathrm{inverted-file}} + \beta \cdot \mathrm{Sim}_{\mathrm{cluster-analysis}}, \tag{5}$$

where α and β are parameters to adjust the different values of the weighting schemes and give more importance to the similarities of the first or the second step. We re-rank documents according to combined similarities. And then, we present the results to the user. Even though a document having low query–document similarity can be given high query–cluster similarity due to the effects of other documents in the cluster. In the reverse case, this is the same.

At the re-ranking step, we get the view matching the query by applying dynamic cluster partitioning to documents of which similarity is calculated according to containment of query terms. And through the cluster analysis, the context of all terms in a document as well as query terms is considered.

4. Experiments and performance evaluation

4.1. Test collection

We evaluated re-ranking model with the ETRI–KEMONG test collection which is a Korean encyclopedia published by the Kemong company (Kemong, 1992). It is published in six volumes with 500 pages per volume. The text data contains 23113 entries, and its size is about 10 mega-bytes. The content of each entry describes the concept with other entries or more fundamental words. The test set contains 46 natural language queries and relevance information of entry lists related to each query.

Table 1 gives the structure of the test collection and Table 2 gives the statistics of the test collection.

4.2. Experiments and results

The goal of the experiments is to validate the proposed model. We took the 46 ETRI–KEMONG queries as originally written and retrieved the N top-ranked documents using SMART system in Korean version where N is the number of documents where similarity is above 0 in response to the query.

Fig. 4 shows an example of the document re-ranking system. Through the morphological analysis of the documents, we extract nouns and represent them as vectors, and construct the hierarchical document clusters depending on similarities between documents. For document clustering, weight in document vector was calculated by the atc weighting scheme though other weighting schemes such as nnn, ntn, ntc, ltn, ltc, and atn can be used. But, in vector space retrieval, we tested the performances of the various weighting schemes for query and document.

We use the SMART retrieval system using *n*-grams for Korean text retrieval (Lee & Ahn, 1996). This system is based on the vector space model and includes most of the well-tested weighting schemes, providing a rich environment for experimental testing.

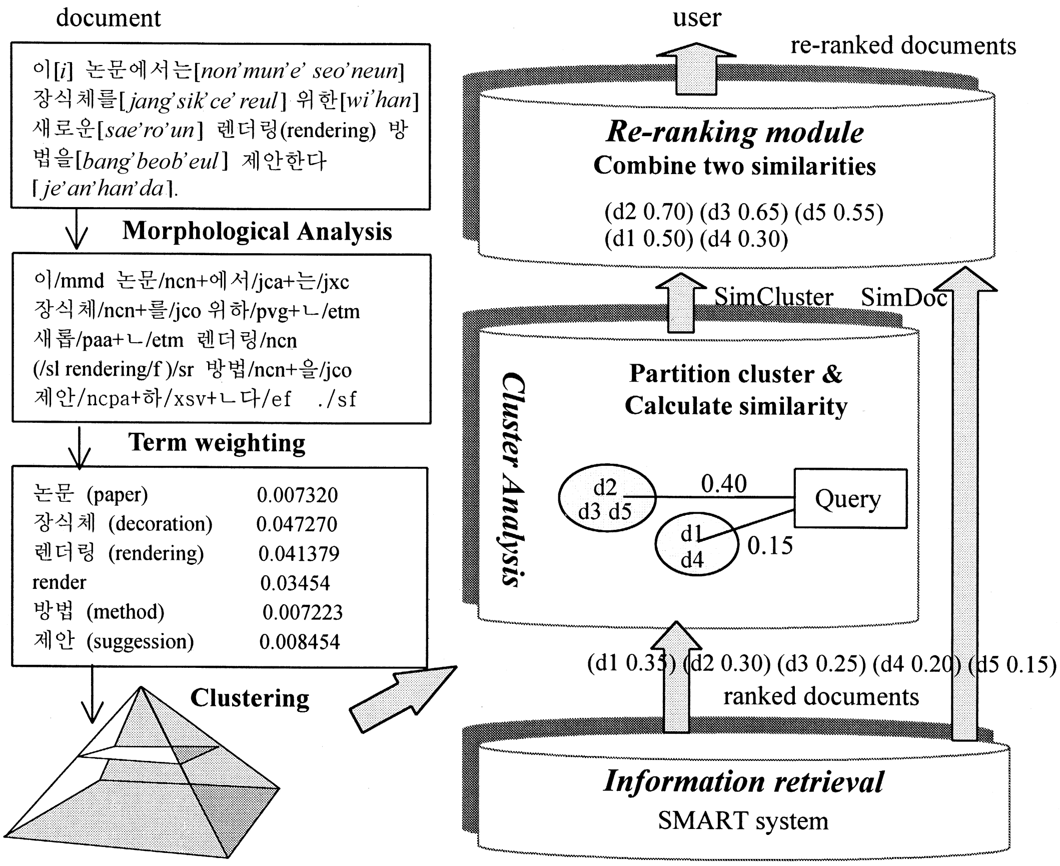
Table 1
The structure of the document

⟨*doc*⟩	a document;
⟨*id*⟩	identification of document; ⟨*id*⟩ 00001
⟨*title*⟩	entry title of encyclopedia;
⟨*see*⟩	standard entry used for the entry; ⟨*title*⟩ ga'geug ⟨*see*⟩ o'pe'ra (opera)
⟨*seealso*⟩	reference entry or related entry;
⟨*contents*⟩	contents of the entry;
⟨*subtitle*⟩	entry subtitle; ⟨*subtitle*⟩ *yeog'sa* (history) ⟨*contents*⟩ contents of the subtitle.

Table 2
Test set statistics

Documents	23113
Queries	45
Average document length	56 words
Average query length	3 words
Average relevant documents	9

At the time of running, we retrieved documents by an initial SMART search. After that, we analyze the distribution of these documents in a cluster and partition the cluster dynamically from the viewpoint of query. We calculate the similarity between cluster centroid and query. Even though a document has low similarity in the first retrieval step, it was given high priority in the second step. We combine two similarities from the first retrieval and the second analysis step, and re-rank documents according to the combined similarities.

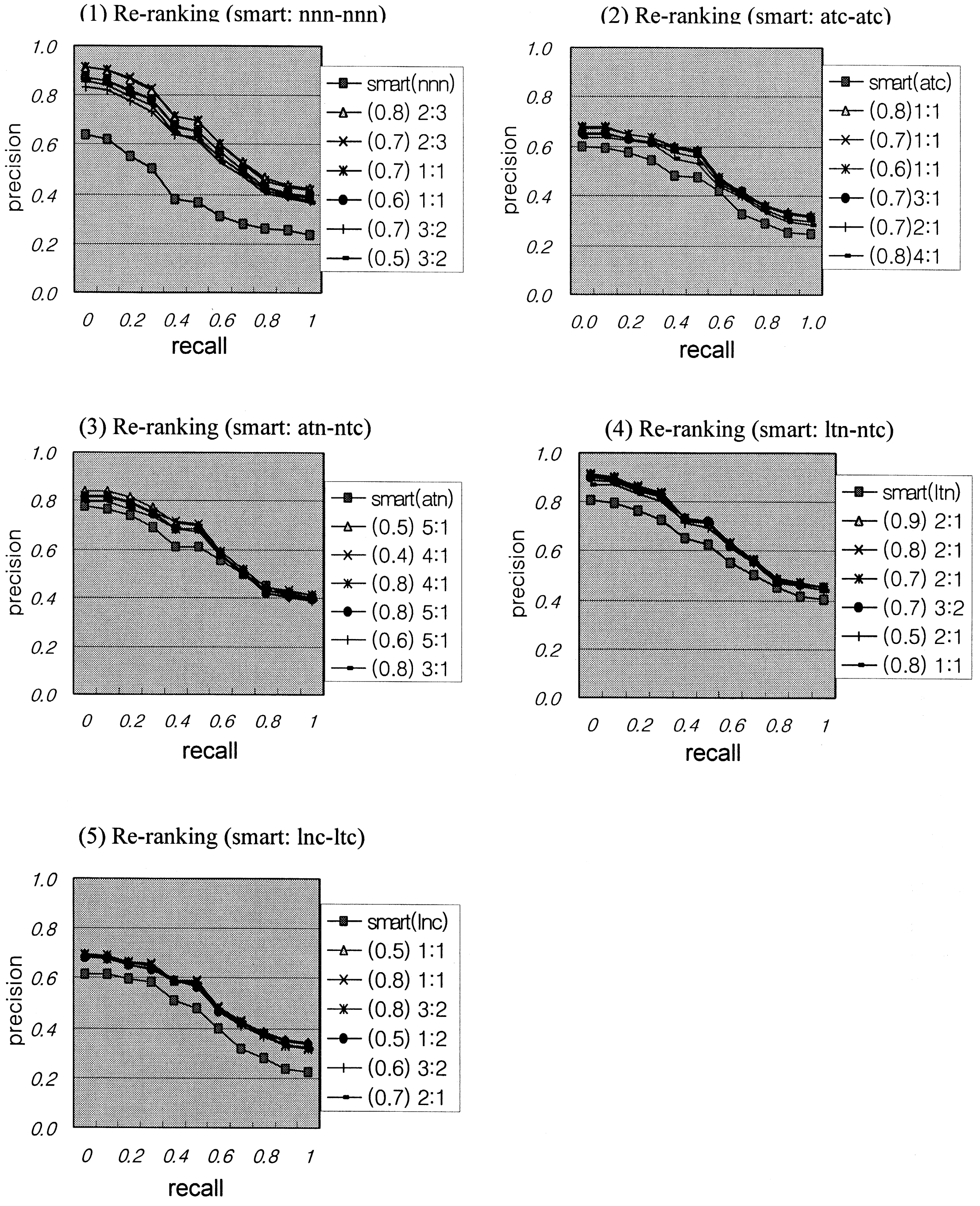


document
이[*i*] 논문에서는[*non'mun'e' seo'neun*] 장식체를[*jang'sik'ce' reul*] 위한[*wi'han*] 새로운[*sae'ro'un*] 렌더링(rendering) 방법을[*bang'beob'eul*] 제안한다 [*je'an'han'da*].
Morphological Analysis
이/mmd 논문/ncn+에서/jca+는/jxc
장식체/ncn+를/jco 위하/pvg+ㄴ/etm
새롭/paa+ㄴ/etm 렌더링/ncn
(/sl rendering/f)/sr 방법/ncn+을/jco
제안/ncpa+하/xsv+ㄴ다/ef ./sf
Term weighting
논문 (paper) 0.007320
장식체 (decoration) 0.047270
렌더링 (rendering) 0.041379
render 0.03454
방법 (method) 0.007223
제안 (suggession) 0.008454
Clustering
user
re-ranked documents
Re-ranking module
Combine two similarities
(d2 0.70) (d3 0.65) (d5 0.55)
(d1 0.50) (d4 0.30)
SimCluster
SimDoc
Cluster Analysis
Partition cluster &
Calculate similarity
d2
d3 d5
0.40
Query
d1
d4
0.15
(d1 0.35) (d2 0.30) (d3 0.25) (d4 0.20) (d5 0.15)
ranked documents
Information retrieval
SMART system

Fig. 4. An example of the document re-ranking.

Table 3
The retrieval effectiveness of the re-ranking model

Recall	Precision									
	smart nnn–nnn	re-rank (0.8)2:3	smart atc–atc	re-rank (0.8)1:1	smart lnc–ltc	re-rank (0.5)5:1	smart atn–ntc	re-rank (0.9)2:1	smart ltn–ntc	re-rank (0.5)1:1
0.0	0.6397	0.9110	0.6004	0.6775	0.6142	0.6962	0.7776	0.8422	0.8097	0.9107
0.1	0.6209	0.8999	0.5942	0.6775	0.6142	0.6907	0.7631	0.8390	0.7937	0.8996
0.2	0.5559	0.8663	0.5767	0.6498	0.5956	0.6647	0.7389	0.8174	0.7626	0.8591
0.3	0.5039	0.8225	0.5446	0.6387	0.5860	0.6470	0.6896	0.7694	0.7284	0.8334
0.4	0.3774	0.7152	0.4836	0.5970	0.5103	0.5905	0.6112	0.7125	0.6508	0.7350
0.5	0.3638	0.6983	0.4731	0.5885	0.4761	0.5700	0.6102	0.6999	0.6278	0.7136
0.6	0.3112	0.6017	0.4228	0.4753	0.3986	0.4759	0.5579	0.5955	0.5515	0.6331
0.7	0.2815	0.5305	0.3289	0.4160	0.3183	0.4229	0.4984	0.5215	0.5029	0.5683
0.8	0.2632	0.4634	0.2919	0.3648	0.2795	0.3844	0.4477	0.4450	0.4522	0.4879
0.9	0.2524	0.4351	0.2542	0.3320	0.2364	0.3535	0.4155	0.4261	0.4166	0.4715
1.0	0.2383	0.4199	0.2478	0.3195	0.2228	0.3389	0.4024	0.4114	0.4011	0.4535
11pt avg change %	0.4008	0.6694 + 67.16	0.4380	0.5215 + 19.06	0.4411	0.5304 + 20.24	0.5920	0.6436 + 8.72	0.6088	0.6878 + 12.98


(1) Re-ranking (smart: nnn-nnn)

(2) Re-ranking (smart: atc-atc)

(3) Re-ranking (smart: atn-ntc)

(4) Re-ranking (smart: ltn-ntc)

(5) Re-ranking (smart: lnc-ltc)

Fig. 5. The retrieval effectiveness results of the re-ranking.

The following are brief descriptions of the two methods tested and compared.

- SMART method: The SMART system for Korean text retrieval ranks the retrieved documents in decreasing order of query–document similarities.
- Proposed method: We re-evaluate the documents based on cluster analysis, with a varying number of relevant documents retrieved by an initial SMART search.

The above two methods are tested on the test data and their results are evaluated with respect to the result by manual retrieval that is a part of the test data.

We set up a test environment of the SMART system as bi-gram and evaluate performance in re-ranking method using four weighting systems such as lnc–ltc, atn–ntc, ltn–ntc, atc–atc, each of which produces a good retrieval result as a single weighting scheme, and nnn–nnn weighting scheme which considers simple term frequency. Performance differs depending on the threshold used in partitioning clusters in the re-ranking step. And the composition rate (α,β) of similarity of inverted method and that of cluster analysis affects performance. Fig. 5 compares the performances of the two methods in various weighting schemes, threshold and α,β values.

For example, in smart:atc–atc (0.8) 1:1, the number in the parenthesis means threshold in Fig. 5. (0.8) of (0.8) 1:1 means that we select a cluster having a minimum rate of static to a dynamic cluster more than 0.8 when partitioning a hierarchical cluster, and 1:1 represents the composition rate of similarity of cluster and similarity of SMART system. In Eq. (5), α and β were set to 1. The composition rate is to minimize effects on difference in similarity value as to the weighting scheme. Table 3 shows the most effective performance for Fig. 5 when using the proposed method for each weighting scheme in SMART system. The results were evaluated by applying the average precision of a set of queries at 11 representative recall points.

The performances of cluster-based retrieval for various threshold values are below those of inverted-file method. The best 11-point average in cluster-based retrieval was 0.310. But, a proposed method is interested not in ranks but similarities from cluster-based retrieval. Therefore, we evaluated the performance for the combined similarities.

In ETRI–KEMONG set, the specific example for ranking by SMART search (nnn–nnn weighting scheme) and re-ranking by cluster analysis is as follows (Table 4):

Query 18: 지구 [*ji-gu*] (earth) 자전 [*ja-jeon*] (rotation) 공전 [*gong-jeon*] (revolution)

The '공전' is polysemous word which has three meanings such as 'code of laws', 'public field'

Table 4
The results of the re-rank for the polysemous query

Document id	Title of the document	Rank	re-rank
1828	공전 (code of laws)	153	111
1829	공전 (public field)	110	110
1830	공전 (revolution of the earth)	109	22
1831	공전주기 (revolution cycle)	70	20
18215	지구 (the earth)	1	1

and 'revolution of the earth'. The documents 1830, 1831 and 18215 formed the nearest clusters because they share terms such as '공전 (revolution), 지구 (earth), 둘레 (circumstance), 위성 (satellite), 천체 (heavenly body), 태양 (the sun), 행성 (planet)'. But, the document 1828 which has irrelevant meanings for query, make a singleton cluster whose only member is the document itself. The document 1829 is the same. In ranking by inverted-file method, documents 1828, 1829 and 1830 have similar ranks, but document 1830 has high priority due to the effects of other documents (1831 and 18215) in the same cluster by cluster analysis.

The proposed method achieved about a 67.16% improvement for nnn–nnn, 19.06% for atc–atc, 20.24% for lnc–ltc, 8.72% for atn–ntc and 12.98% for ltn–ntc weighting scheme. These results are modestly encouraging. In our model, we could improve performance by reflecting the context of documents implicitly through cluster analyses for documents retrieved by the inverted-file method.

5. Conclusion and future work

In this paper, we have proposed a model for an information retrieval system that is based on a document re-ranking method using clusters. We use semi-dynamic clusters by performing cluster partitioning according to the behaviors of retrieved documents from the initial search. Therefore, our approach produces clusters that are tailored to characteristics of the query. Clustered contexts act as a set of logical foci for query refinement. Also, the context of a document is considered in the retrieved results by the combination of information search and cluster analysis as to the user's information needs.

We have presented strong evidence that the document re-ranking using clusters is one which can produce significant improvements over the method based on similarity search ranking alone.

In the future work, user-oriented retrieval can be made possible by adding a user profile management system as plug and play architecture to the re-ranking step in the proposed model. The relevance of the retrieved documents for the same query can be different depending on the user's interest. A user profile consists of information about the user that has bearing on the user's information needs. A simple user profile is much like a query. It consists of a set of key terms, often with given weights. User profile can consist of various records according to the user's individual peculiarities. The user profile system can select an appropriate user's profile for a given query, and calculate similarity between selected user profile instead of query and cluster centroid. Because the document cluster has appropriate structure to compare with the user's profile, the user's interest is considered in the re-ranking step by comparing user profile with cluster centroids. Our model may give higher priority to a collection of documents satisfying the user's interest so that the user can minimize cost and time in retrieving documents. Therefore, we expect that the use of user profile will improve the performance on the grounds that user profile can be viewed as the expanded query.

Acknowledgements

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the "Study on Multi-lingual Information Retrieval" project at the Advanced Information Technology Research Center (AITrc).

References

Allan, J. (1995). Relevance feedback with too much data. In *Proceedings of 18th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 337–343).

Anick, P. G., & Vaithyanathan, S. (1997). Exploiting clustering and phrases for context-based information retrieval. In *Proceedings of 20th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 314–323).

Buckley, C., Salton, G., & Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In *Proceedings of 17th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 292–298).

Deerwester, S., Dumais, S. T., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*(6), 391–407.

Eguchi, K. (1999). Adaptive cluster-based browsing using incrementally expanded queries and its effects. In *Proceedings of 22nd ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 265–266).

Fitzpatrick, L., & Dent, M. (1997). Automatic feedback using past queries: social searching? In *Proceedings of 20th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 306–313).

Frakes, W. B., & Baeza-Yates, R. (1992). In *Information retrieval: data structures & algorithms* (pp. 435–436). New Jersey: Prentice Hall.

Hearst, M. A., & Pedersen, J. O. (1996). Re-examining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of 19th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 76–84).

Kang, H. K. (1997). *Two-level document ranking methods using mutual information in natural language information retrieval*. Ph.D. thesis, Department of Computer Science, Korea Advanced Institute of Science and Technology.

Kemong (1992). *The Kemong Company new encyclopedia*. Seoul: Kemongsa Publishing Co.

Lee, J. H., & Ahn, J. S. (1996). Using n-grams for Korean text retrieval. In *Proceedings of 19th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 216–224).

Mauldin, M. L., & Carbonell, J. G. (1991). *Conceptual information retrieval: a case study in adaptive partial parsing*. Kluwer Academic Publishers: Boston, Dordrecht, London.

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *Computer Journal*, *26*, 354–359.

Park, Y. C. (1997). *Building word knowledge for information retrieval using statistical information*. Ph.D. thesis, Department of Computer Science, Korea Advanced Institute of Science and Technology.

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

Ward, J. H. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*(301), 235–244.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global. In *Proceedings of 19th ACM SIGIR International Conference on Research and Development in Information Retrieval* (pp. 4–11).