Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 41 (2005) 1065–1080
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

Document indexing: a concept-based approach to term weight estimation

Bo-Yeong Kang *, Sang-Jo Lee

Department of Computer Engineering, Kyungpook National University, 702-701 Sangyuk-dong, Pukgu, Daegu, Korea

Received 30 September 2003; accepted 11 August 2004
Available online 11 November 2004

Abstract

Traditional index weighting approaches for information retrieval from texts depend on the term frequency based analysis of the text contents. A shortcoming of these indexing schemes, which consider only the occurrences of the terms in a document, is that they have some limitations in extracting semantically exact indexes that represent the semantic content of a document. To address this issue, we developed a new indexing formalism that considers not only the terms in a document, but also the concepts. In this approach, concept clusters are defined and a concept vector space model is proposed to represent the semantic importance degrees of lexical items and concepts within a document. Through an experiment on the TREC collection of Wall Street Journal documents, we show that the proposed method outperforms an indexing method based on term frequency (TF), especially in regard to the few highest-ranked documents. Moreover, the index term dimension was 80% lower for the proposed method than for the TF-based method, which is expected to significantly reduce the document search time in a real environment.

Keywords: Weighting function; Index weight estimation; Automatic indexing; Information retrieval

1. Introduction

The growth of the Internet has seen an explosion in the amount of information available, leading to the need for increasingly efficient methods for information retrieval. To intelligently retrieve information, indexing should be based not only on the occurrences of terms in a document, but also on the content of the document. Despite this obvious need, most existing indexing and the weighting algorithms analyze term occurrences and do not attempt to resolve the meaning of the text. As a result, existing indexing

* Corresponding author. Tel.: +82 53 940 8692.
E-mail address: comeng99@hotmail.com (B.-Y. Kang).

0306-4573/$ - see front matter
doi:10.1016/j.ipm.2004.08.005

methods do not comprehend the topics referred to in a text, and therefore have dificulty in extracting semantically important indexes.

To address this shortcoming, in the present study we developed a novel indexing method that regards a document as a conglomeration of concepts. In the proposed methodology, indexes are extracted from a document based on these concepts and then weighted according to their degree of semantic importance in the document. For the extraction of concepts, we exploit concept clusters containing semantically related lexical items. Additionally, an *n*-dimensional concept vector space model is proposed to estimate the semantic importance degree of each concept and lexical item within a document.

This paper is organized as follows. Section 2 describes related work on indexing. In Section 3, we present our concept-based indexing methodology, and in Section 4 we present the results of, and compare, experiments using the proposed methodology and a traditional indexing scheme. Our conclusions are given in Section 5.

2. Related work

In general, index terms describe the content of a text to different extents. In indexing algorithms, this characteristic is captured by assigning each term a weight that acts as an indicator of the relative importance of the term. Many weighting functions have been proposed and tested (Fuhr & Buckley, 1991; Lee, 1995; Luhn, 1957; Salton, 1975a; Salton & Buckley, 1988; Salton & McGill, 1983; Sparck Jones, 1972, 1973). However, most such functions developed to date depend on statistical methods or on the document's term distribution tendency. Representative weighting functions include such factors as term frequency (TF), inverse document frequency (IDF), the product of TF and IDF, and length normalization (LN).

Most indexing and weighting functions based on statistical methods suffer from limitations that diminish the precision of the extracted indexes (Moens, 2000). TF is useful when indexing long documents, but not short ones. However, TF algorithms do not generally represent the exact TF because they do not take into account characteristics such as anaphoras, synonyms, and so on. In addition, IDF is inappropriate for indexing a reference collection that changes frequently because the weight of each index term needs be recomputed every time the documents change. LN was proposed to account for the fact that TF factors are numerous for long documents but negligible for short ones, obscuring the real importance of terms. However, as this approach uses the TF function, it suffers from the same shortcomings as TF does.

A further drawback of most TF-based methods is that they have diffculties in extracting semantically exact indexes that express the topics of a document. For example, in the sample text shown in the following, the important terms that could be topics of the text are *yoga*, *exercise*, *health* and *mind* etc.

> "*Yoga* combines the physical *exercises* that stretch and tone your *body* with the *nurture* and *development* of your emotional *health* and *well-being*. This simple *practice* which works your *back*, *hips*, *neck* and *shoulders*, is ideal for relaxing your *mind* and *body* when you feel tired and stressed. All you need is 20 to 25 *minutes* to help slow your *breathing*, gently exercise your *body* and move your *mind* toward a *state* of *stillness* and *clarity*".

However, the TF weight of the word *yoga* is 1, which is the same as that of semantically unimportant words such as *nurture* and *development*. Thus, the TF approach fails to capture the topics of the text and cannot discriminate the degree of semantic importance of each lexical item within the text. Various attempts have been made to enhance the indexing performance by exploiting linguistic phenomena (Kazman, Al-Halimi, Hunt, & Mantei, 1996; Kominek & Kazman, 1997). One such linguistic phenomenon is the *lexical chain*, which links related lexical items in a text (Morris & Hirst, 1991). If we look for lexical chains in the sample text shown in Fig. 1, we obtain the nine chains taking a term that has no relation with other

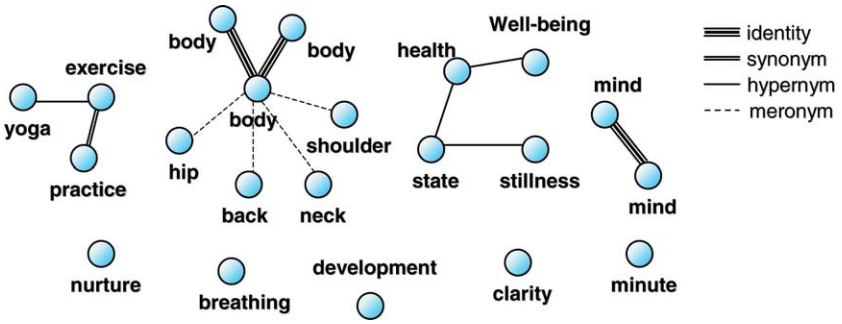

Fig. 1. Lexical chains of a sample text "yoga".

nouns as a chain. In this scheme, the words *yoga* and *exercise* are in the same chain because they are related by a hyponym/hypernym relation. This approach correctly indicates that the important words of the text are *yoga*, *exericse*, *health*, *mind* and *body* other than *nurture* and *development*.

It is generally agreed that lexical chains represent the discourse structure of a document and provide clues about the topicality of a document (Morris, 1988; Morris & Hirst, 1991). Morris and Hirst defined a lexical chain as a cohesive chain of words in which the criterion for inclusion of a word is that it bears a cohesive relation to a word that is already in the chain. Morris and Hirst suggested the use of a thesaurus, such as Roget's, for specifying whether a cohesive relation exists between words. Two words can be considered related if they are connected in the thesaurus in one of the following five ways:

1. Their index entries point to the same thesaurus category or to adjacent categories.
2. The index entry of one word contains the other.
3. The index entry of one word points to a thesaurus category that contains the other.
4. The index entry of one word points to a thesaurus category that in turn contains a pointer to a category pointed to by the index entry of the other.
5. The index entries of each word point to thesaurus categories that in turn contain a pointer to the same category.

Morris and Hirst constructed and evaluated the lexical chains by hand using five texts. However, they were never able to implement an automated version of their algorithm because on-line thesauri were not available to them.

In an attempt to transfer Morris and Hirst's lexical chain algorithm to the online lexical knowledge base, WordNet, Hirst and St-Onge defined three major types of relations between nouns in WordNet (Hirst & St-Onge, 1998). WordNet is comprised of four files: verbs, adverbs, adjectives, and nouns. Because the verb file has no relation with the three other files, and the adverb file has only unidirectional relations with the adjective file, Hirst and St-Onge limited the chaining process only to nouns in their research. Moreover, because the structure of WordNet is quite different from that of Roget's Thesaurus, they needed to replace the Roget's-based definition of semantic relatedness used by Morris and Hirst with one based on WordNet, while retaining the algorithm's essential properties. They defined three kinds of relation: extra-strong, strong and medium-strong. An extra-strong relation holds only between a word and its literal repetition; such relations have the highest weight. A strong relation has a lower weight than an extra-strong relation but a higher weight than a medium-strong relation; there are three kinds of strong relations. Finally, they postulated that two words are related in a medium-strong fashion if there exists an allowable path connecting a synset associated with each word, where an allowable path is one that contains no more than five links and conforms to one of the eight patterns described by Hirst and St-Onge.

Carthy had research into the use lexical chains to build effective topic tracking systems (Carthy, 2002). Topic tracking involves tracking a news event in a stream of news stories i.e. finding all subsequent stories to the news stream that discuss the given event. We could see that LexTrack outperforms the keyword-based system, KeyCos, that they implemented, in terms of recall performance. KeyCos was based on traditional IR techniques such as using cosine similarity to measure the similarity between a tracking story and incoming story.

Barzilay and Elhadad investigate the use of lexical chains as a model of the source text for the purpose of producing a summary (Barzilay & Elhadad, 1997). They presented the new algorithm to compute lexical chains in a text, merging several robust knowledge sources: WordNet, a part-of-speech tagger, shallow parser and a segmentation algorithm. They used the three kinds of relations that Morris and Hirst defined.

Al-Halimi and Kazman (Kazman et al., 1996; Kominek & Kazman, 1997) developed a method for indexing transcriptions of conference meetings by topic using lexical trees, the two-dimensional version of lexical chains. They conducted a preliminary study to verify the utility of lexical trees for automatic indexing of arbitrary text. However, although their method demonstrated the potential usefulness of lexical trees in text indexing and retrieval, in its present form their method is inappropriate for use in document retrieval. For an indexing method to be of use in information retrieval, each index term for the document should be a topic and have a weight that represents the degree of semantic importance of the term within the document. However, although the method of Al-Halimi and Kazman can extract topics as index terms from the transcript of a conference seminar, it does not contain a function to estimate the weight of each extracted topic.

Therefore, in the present study, we propose a new, conceptual approach based on lexical chains for extracting terms from a text and assigning them weighting that can capture the semantic content of a document and represent the importance of a word within a document considering concepts.

3. Concept-based indexing

To address the semantic issues of TF-based indexing methods, we propose an approach that considers not just the terms but also the concepts of a document. In this approach, the concepts of a document are extracted, and, from those concepts, the semantic indexes and their weights are derived.

3.1. System overview

A schematic overview of the proposed methodology is shown in Fig. 2. When applied to a input document, the proposed method first clusters semantically related terms that can represent the semantic content of the text and assigns scores to the extracted clusters and lexical items in the clusters based on term relations. Among scored concept clusters, representative concept clusters are selected by the defined criterion. Then, each scored representative concept is represented as a vector in concept vector space, and the importance of the terms within a document is then computed according to the overall text vector and the concept vector in which the terms are included. Finally, semantic indexes and their weights are extracted.

The proposed system has three main components:

- Clustering based on lexical chains.
- Weight estimation based on word relations.
- Weight re-estimation based on concept vector space.

The clustering component employs lexical chains, and the latter two components are related to the index term weighting based on the term relations and the concept vector space.

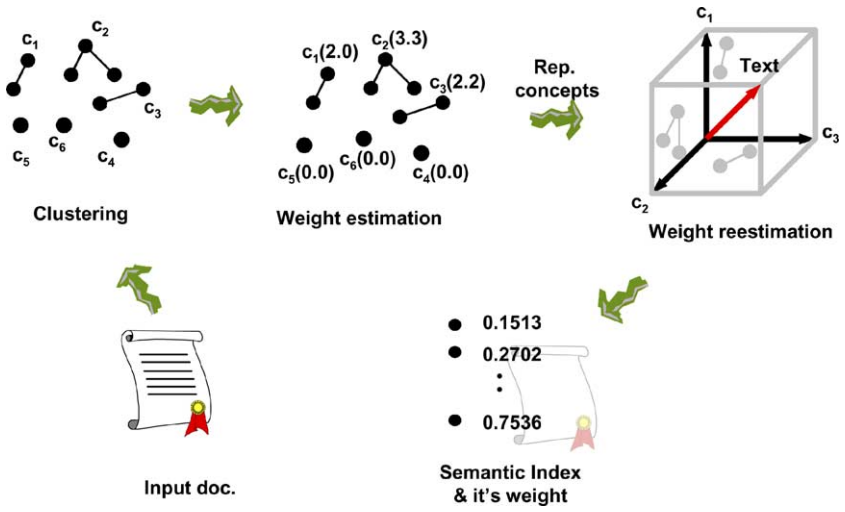

Fig. 2. An overview of the proposed system: the semantic processing flow.

3.2. Concept clusters

Documents generally contain various concepts, and we must determine those concepts if we are to comprehend the aboutness of a document. In accordance with the accepted view in the linguistics literature that lexical chains provide a good representation of discourse structures and topicality of segments (Morris, 1988; Morris & Hirst, 1991), here we take each lexical chain to represent a concept that expresses one aspect of the meaning of a document.

Morris and Hirst were able to use various kinds of syntactic categories (except pronouns, articles, and high-frequency words) when composing lexical chains, because they used Roget's Thesaurus as a knowledge base (Morris, 1988; Morris & Hirst, 1991). However, in the present work we use WordNet, which has far fewer cross-category connections compared to Roget's Thesaurus. Because of WordNet's limited cross-category connections, we followed the approach of previous investigators and limited our investigations to nouns (Budanitsky, 1999).

Hirst and St-Onge adapted the Roget's-based relations of Morris and Hirst to WordNet-based ones, extra-strong, strong and medium-strong relations (Hirst & St-Onge, 1998). Two words are related in a medium-strong fashion if there exists an allowable path between words, and a path is allowable if it contains no more than five links and conforms to one of the eight patterns described in Hirst and St-Onge. Consequently, numerous kinds of relations can be used in composing lexical chains. The large number of possible word relations means that, if the proposed method is used to index a massive number of documents, there would be a large number of parameters and hence the indexing and retrieval computations would take a long time. Therefore, in the present work on the clustering of lexical items, we considered only four kinds of relations—identity, synonymy, hypernymy (hyponymy), and meronymy. Hypernymy and hyponymy are regarded as one relation because they are inter-definable.

The use of traditional lexical chains for indexing is based on the notion that such sequences of lexical items contain clues about the discourse structure or topicality of a document. However, traditional lexical chains do not give any information on the semantic importance degrees of the lexical items or lexical chains within a document, and no explicit function has yet been developed to measure the relative semantic importance within the lexical chain method. The notion of a concept cluster proposed here not only groups related lexical items, but also assigns each lexical item and concept a weight that represents its semantic

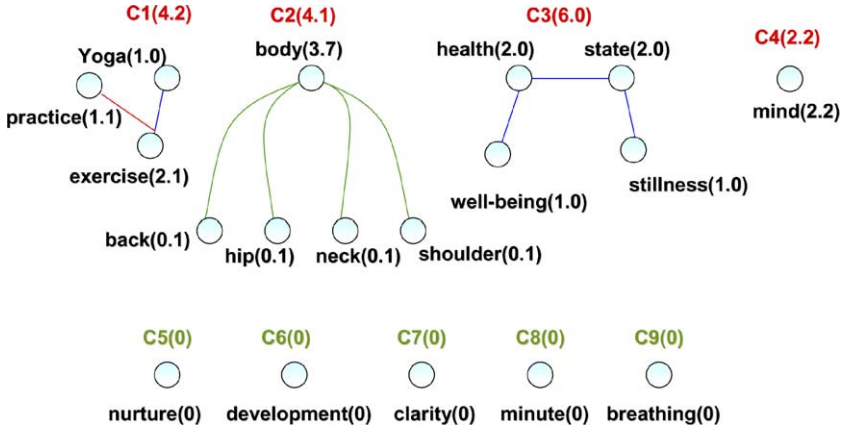

Fig. 3. Concept clusters of the sample text in Fig. 1.

importance degree within a document. Therefore, we define a concept cluster as a weighted lexical chain that represents one aspect of the meaning of a document and expresses the semantic importance degree within a document using the following definition.

Definition 1 (Concept cluster). Let $N = \{N_1, N_2, \ldots, N_l\}$ be the set of nouns in a document, and $R = \{\text{identity, synonym, hypernym(hyponym), meronym}\}$ be the set of lexical relations. Let $C = \{C_1, C_2, \ldots, C_m\}$ be the set of concept clusters in a document. Concept cluster C_j is composed of N_i and R_k, where $R_k \in R$, $N_i \in N$ and $C_j \in C$. Each N_i and C_j have a weight that represents their respective degrees of semantic importance within a document.

As an example of this approach, we apply our clustering approach to the sample text considered above Fig. 1. As shown in Fig. 3, the concepts in the sample text fall into nine categories (*exercise*, *body*, *health*, and *mind* etc.). Of the nine concept clusters considered, we see that the clusters 1, 2, 3 and 4 are more representative clusters than the other clusters because those clusters consist of a larger number of terms and lexical relations that are exploited to search for the semantically important terms in a document.

The way to construct a concept cluster from a given document is described in the following section in more detail.

3.3. Concept cluster identification

3.3.1. Clustering based on lexical chains

Clustering is achieved by four inter-noun relations: identity, synonymy, hypernym(hyponym), meronym(holonym). If we were to consider more relations (e.g., transitive, antonym, etc.), we would need to manage more parameters when calculating the word weights, which would degrade the information retrieval performance. Therefore, to reduce the range of the problem, we used only a restricted set of relations in the present work to show that the proposed approach enhances IR performance.

Clustering information is obtained from WordNet. Because the verb file has no relation with the three other files as mentioned in Section 2, Hirst and St-Onge limited the chaining process only to nouns in their research. By the same reason, in this paper, we take only nouns of clustering candidates. Furthermore, this result in stopwords elimination as side effect.

The procedural steps for clustering algorithms are simply described in Algorithm 1: Firstly, a document is tagged, and the chaining candidates are selected. Then, they are loaded into the clustering candidate array with processing identity relation. Identity relation is the repetition of a word, therefore it can be processed

Algorithm 1 (Clustering algorithm)
Input: Input document file
Output: A number of weighted clusters
Procedure:

1. Tag a document;
2. Select chaining candidates, $C_w = \{W_1, W_2, \ldots, W_n\}$, and load them into the clustering candidate array processing identity relation;
3. Initialize the *current candidate pointer i*
4. **while** $(i < n)$ **do**
5. **for** $(j = i + 1; j < n; j{+}{+})$ **do**
6. If HasRelation(W_i, W_j) defined in WordNet where $W_j \in C_w$ Link(W_i, W_j);
7. **end for**
8. Increase the *current candidate pointer i*
9. **end while**

directly when loading without the help of WordNet. In the third step, the current candidate pointer to indicate the current noun is initialized as index 1 to indicate the first noun among chaining candidates. If the current noun has a relation with other nouns after the current noun index, link it with other nouns, and if there is no relation with other nouns, increase the current noun index. If the current noun index is over the total number of nouns, the system stops the chaining process, otherwise it continues the chaining procedure.

3.3.2. Weight estimation based on term relations

The natural networks such as the World Wide Web have been found that they have a hub structure that the distributions of the number of connections follow power laws (Kleinberg & Lawrence, 2001). A hub on WWW is defined as a page that points to many nodes. Characteristic patterns of hubs and authorities can be used to identify communities of pages on the same topic (Kleinberg & Lawrence, 2001). These analysis of the Web's structure is leading to improved methods for accessing and understanding the available information. Steyvers and Tenenbaum presented the graph-theoretic analysis of three types of semantic networks, word association, WordNet and Reget's thesaurus. They showed that these semantic networks also have a small world structure similar to that found in WWW (Steyvers & Tenenbaum, submitted for publication). These researches on semantic networks and natural networks focuses on the analysis of the network characteristics, and use the link information as an important factor to find out the network structure.

Because the node that have many links with other nodes, such as a hub, play an important role in identifying the topical structure of a network, the node that have many links with other nodes in a network may be regarded as more important node than ones that have little links with other nodes. From this point of view, we deal the link information of a concept cluster as means to measure the importance degree of a word within a document.

To estimate the semantic importance of terms within a given document, we think that the words having more relations with other words are semantically more important in a document. Therefore, based on word relations, we define two scoring functions for each concept cluster and the terms in that cluster as Definitions 2 and 3.

Definition 2 (Score of noun). Let $NR^k_{W_i}$ denotes the relation number that noun W_i has with relation k. SR^k represents the weight of relation k. Then the score $S_{\mathrm{NOUN}}(W_i)$ of a noun W_i in a concept cluster is defined as:

$$S_{\mathrm{NOUN}}(W_i) = \sum_k (NR^k_{W_i} \times SR^k) \tag{1}$$

$S_{\mathrm{NOUN}}(W_i)$ is determined by the relations that W_i has with the other terms and their weights. A large value of $S_{\mathrm{NOUN}}(W_i)$ indicates that W_i is a semantically important term in a document. The relation weight SR^k is in the order listed: identity, synonym, hypernym(hyponym), meronym (i.e., identity highest and meronym lowest) (Fellbaum et al., 1998).

Based on Definition 2, we now define the scoring function of a concept cluster.

Definition 3 (Score of concept cluster). The score $S_{\mathrm{CONCEPT}}(C_x)$ of a concept cluster C_x is defined as:

$$S_{\mathrm{CONCEPT}}(C_x) = \sum_{i=1}^{n} S_{\mathrm{NOUN}}(W_i) \tag{2}$$

where $S_{\mathrm{NOUN}}(W_i)$ is the score of noun W_i, and $W_1, \ldots, W_n \in C_x$.

Thus, $S_{\mathrm{CONCEPT}}(C_x)$ is obtained by summing the scores of all the terms in C_x. A large value of $S_{\mathrm{CONCEPT}}(C_x)$ indicates that C_x is a semantically important concept in the document.

For example, consider the system in Fig. 4, in which the identity relation weight, SR^{iden}, is set to 0.7 and the synonym relation weight, SR^{syn}, is set to 0.5. Given that noun W_1 has one identity relation ($NR^{\mathrm{iden}}_{W_1} = 1$) and two synonym relations ($NR^{\mathrm{syn}}_{W_1} = 2$), the score of W_1 is found to be 1.7 by the following calculation:

$$S_{\mathrm{NOUN}}(W_1) = \sum_k (NR^k_{W_i} \times SR^k) = NR^{\mathrm{iden}}_{W_1} \times SR^{\mathrm{iden}} + NR^{\mathrm{syn}}_{W_1} \times SR^{\mathrm{syn}} = 1 \times 0.7 + 2 \times 0.5 = 1.7$$

$$S_{\mathrm{CONCEPT}}(C_1) = \sum_{i=1}^{4} S_{\mathrm{NOUN}}(W_i) = 3.4$$

From the weighted concept clusters, we discriminate representative concepts to represent the content of the document, since we cannot deal with all the concepts of a document. For example, in the system shown in Fig. 3, the clusters that best represent the content of the text are C3 and C4. If the number of concepts in a document is m, a concept C_j will be considered a representative concept if it satisfies the following criterion 4.

Definition 4 (Criterion for a representative concept). Concept clusters, C_j, that satisfy the following criterion are considered representative concepts:

$$S_{\mathrm{CONCEPT}}(C_j) \geqslant \alpha \cdot \frac{1}{m} \sum_{i=1}^{m} S_{\mathrm{CONCEPT}}(C_i) \tag{3}$$

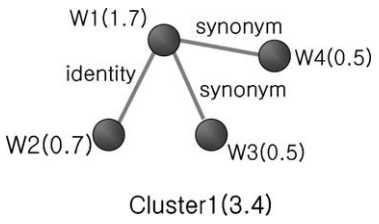

Fig. 4. Score of a sample cluster.

After extracting the representative concepts of a document, we re-estimate the semantic importance of the terms according to the importance of a concept in which the terms are included, which leads to the capture of the semantic index terms and their weights for the document.

3.3.3. Weight re-estimation based on concept vector space

In this section, we describe the method for estimating the semantic importance degree of each lexical item in a document based on the concept vector space. We can discern which concepts are important and which are not using Definitions 2–4 in Section 3.3.2. However, we need to re-estimate the fine-tuned weight of a term considering the concept in which the term is included.

For example, Fig. 5 shows the concept clusters of the sample text shown below. Four representative concept clusters are composed of 12 words and weighted using Definitions 2–4 in Section 3.3.2. The words *practice* and *director* both have weights of 0.3; however, the weight of cluster C4, which contains *practice*, is more than the weight of C3, which contains *director*.

This indicates that, within this document, *practice* belongs to a more important concept than does *director*. We see that *practice* is semantically more important than *director*, because *practice* is in a semantically more important cluster in the document.

> "This exercise routine, developed by Steve Winkler, P.T., director of rehabilitation at the Center for Spine in Savannah, and administrator of the Health Association, focuses on just the right spot: It strengthens your back muscles, abdominals, and obliques (the ab muscle that run from front to back along your lower ribs) and stretches your legs, hips, and chest. Combine this practice with minimum of three 30-min sessions of cardiovascular activity such as walking or jogging, and you should be on your way to a healthier back."

Therefore, we assume that the semantic importance degree of a lexical item is affected by the strength of the concept cluster in which it resides. When two lexical items are assigned the same score, the lexical item whose concept cluster has a higher score is deemed the semantically more important of the two items.

We require a measure that recomputes the weight of lexical items taking into consideration the concept cluster importance. To achieve this, a vector model is employed for the concept space model. In this model, the document is represented by a complex of concepts and the sematic importance degree of lexical items are discriminated by the vector space property.

Definition 5 (Concept vector space model). Concept space is an n-dimensional space composed of n independent concept axes. Each concept vector, $\vec{C}_i$, represents one concept, and has a magnitude of $|\vec{C}_i|$. In concept space, a document vector, $\vec{T}$ is represented by the sum of n-dimensional concept vectors, $\vec{C}_i$.

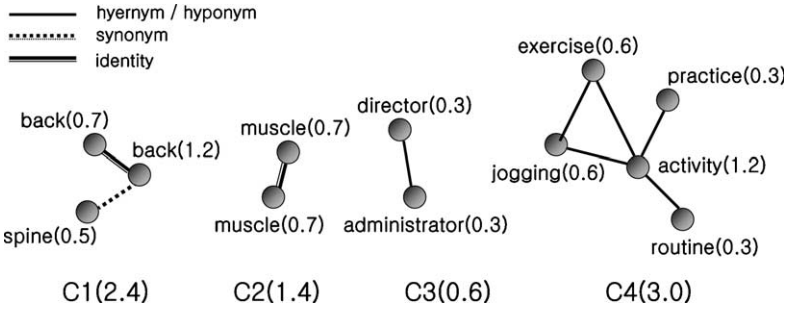

Fig. 5. Semantic importance of lexical items.

$$\vec{T} = \sum_{i=1}^{n} \vec{C}_i \tag{4}$$

Thus, the overall text concept vector is determined by the magnitude of each concept vector. For example, in the sample text of Fig. 6, the important terms which can be topics of a text are *anesthetic* and *machine*. From the composed six concept clusters, we discriminate cluster C3 and C4 as the representative concepts to delegate the content of a document in Fig. 6. Fig. 7 shows the concept space representation of the example in Fig. 6. The overall text vector of the sample text is composed of the representative concept vectors $\vec{C}_3$ and $\vec{C}_4$. Finally, from these concept vectors we must determine the semantic importance degrees of *machine*, *device*, and *anesthetic* within the overall text vector.

The semantic importance of each concept and lexical item can be derived from the properties of this concept vector space. Fig. 8 depicts the process by which the semantic importance of each concept is derived from the overall text vector. The text vector, $\vec{T}$, is derived from concept vectors $\vec{C}_1$ and $\vec{C}_2$. If the magnitudes of vectors $\vec{C}_1$ and $\vec{C}_2$ are $|\vec{C}_1|$ and $|\vec{C}_2|$, respectively, the overall text vector magnitude, $|\vec{T}|$, is $\sqrt{|\vec{C}_1|^2 + |\vec{C}_2|^2}$. In composing text vector magnitude, $|\vec{T}|$, the part that concept $\vec{C}_1$ contributes to $\vec{T}$ is x, and the part that concept $\vec{C}_2$ contributes is y. Each concept is composed of words and has a weight w_i. By generalizing this vector space property, the weights of lexical items and concepts can be estimated as follows.

Definition 6 (Weight quantity, Ω). Weight quantity is the quantity of a text, concept or word weight within the overall document weight. The text weight quantity (Ω_T), concept weight quantity (Ω_C), and word weight quantity (Ω_W) are defined as follows:

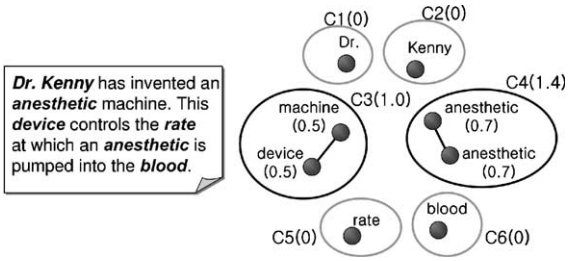

Fig. 6. Concept clusters of a sample text.

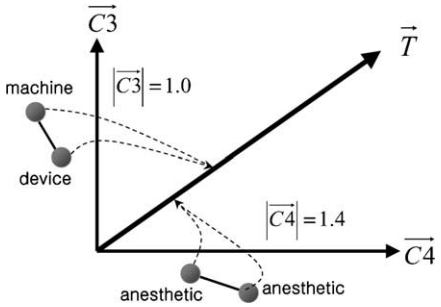

Fig. 7. The concept space version of the sample text in Fig. 6.

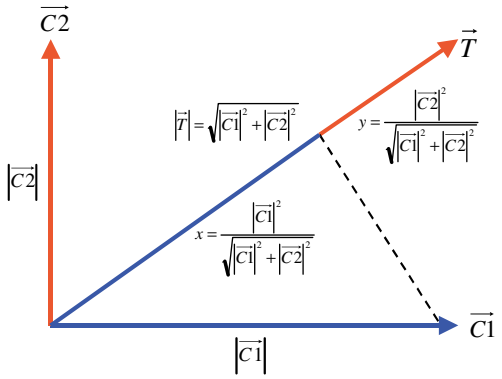

Fig. 8. Vector space property.

$$\Omega_{\mathrm{T}} = \sqrt{\sum_k |\vec{C}_k|^2} \tag{5}$$

$$\Omega_{\mathrm{C}_i} = \frac{|\vec{C}_i|^2}{\sqrt{\sum_k |\vec{C}_k|^2}} \tag{6}$$

$$\Omega_{\mathrm{W}_j} = \Omega_{\mathrm{T}} \times \Psi_{\mathrm{W}_j|T} = \frac{W_j \cdot |\vec{C}_i|}{\sqrt{\sum_k |\vec{C}_k|^2}} \tag{7}$$

The text weight quantity, Ω_{T}, is the combined magnitude of all concepts that form a text. The concept weight quantity, Ω_{C_i}, is the volume of a concept in an overall text vector; this quantity is derived by the same method as that used to derive x and y in Fig. 8. The word weight quantity, Ω_{W_j}, is derived from the product of the overall text weight and the ratio of word weight to text weight. $\Psi_{\mathrm{W}_j|T}$ is illustrated below.

Definition 7 (Weight ratio, Ψ). The weight ratio is the ratio of the weight quantity of a comparative target to the weight quantity of a text, concept or word. $\Psi_{\mathrm{C|T}}$, $\Psi_{\mathrm{W|C}}$ and $\Psi_{\mathrm{W|T}}$ are defined as follows:

$$\Psi_{\mathrm{W}_j|\mathrm{C}_i} = \frac{S_{\mathrm{NOUN}}(W_j)}{S_{\mathrm{CONCEPT}}(C_i)} = \frac{|W_j|}{|\vec{C}_i|} \tag{8}$$

$$\Psi_{\mathrm{C}_i|\mathrm{T}} = \frac{\Omega_{\mathrm{C}_i}}{\Omega_{\mathrm{T}}} = \frac{|\vec{C}_i|^2}{\sum_k |\vec{C}_k|^2} \tag{9}$$

$$\Psi_{\mathrm{W}_j|\mathrm{T}} = \Psi_{\mathrm{W}_j|\mathrm{C}_i} \times \Psi_{\mathrm{C}_i|\mathrm{T}} = \frac{W_j \cdot |\vec{C}_i|}{\sum_k |\vec{C}_k|^2} \tag{10}$$

The formulas for calculating the weights of words and concepts are given above in Definitions 2 and 3. $\Psi_{\mathrm{W}_j|\mathrm{C}_i}$ denotes the weight ratio of a word to the concept in which it is included. $\Psi_{\mathrm{C}_i|\mathrm{T}}$ is the weight ratio of a concept to the text. The weight ratio of a word to the overall text is denoted by $\Psi_{\mathrm{W}_i|\mathrm{T}}$.

Fig. 9 shows the process followed to derive the word weight quantity (Ω_{W}) within the text vector magnitude ($\Omega_{\mathrm{T}}, |\vec{T}|$). Suppose a document is composed of two concepts, one consisting of three words and the other of two words. When two concept clusters are constructed from a document, the weights that we can

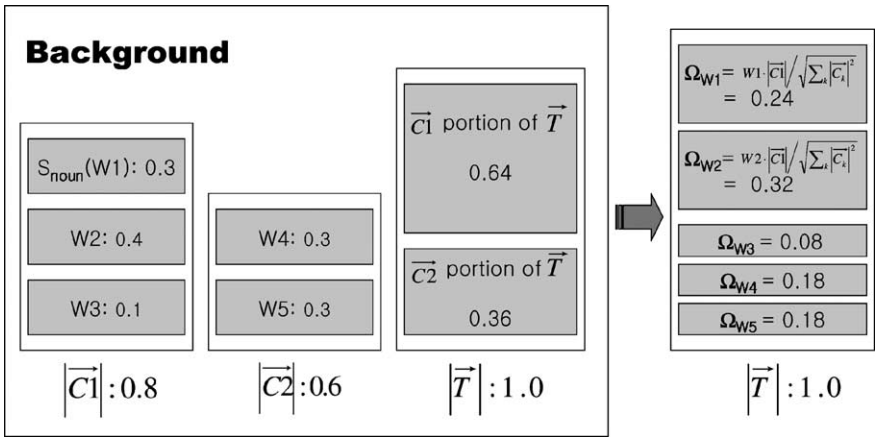

Fig. 9. Derivation of word and concept weights from a text vector.

derive directly from the constructed concept clusters are the weight of each concept ($S_{\text{CONCEPT}}(C_i), |\vec{C}_i|$) and the weight of each word ($S_{\text{NOUN}}(W_i)$). From the nature of the concept vector space, we can know the total text vector magnitude ($\Omega_{\text{T}}, |\vec{T}|$) and each concept weight portion (Ω_{C_i}). Hence, the above Definitions 6 and 7 of the weight of each word (Ω_{W}) and the word weight ratio ($\Psi_{\text{W}_j|\text{T}}$) within the text vector magnitude can be derived from the weights such as $S_{\text{NOUN}}(W_i)$, $S_{\text{CONCEPT}}(C_i)$ and $|\vec{T}|$.

After computing the numerical values of the weight quantity and weight ratio of each lexical item within an overall text, we extract nouns satisfying the following definition as semantic indexes.

Definition 8 (Semantic index). The semantic index that represents the content of a document is defined as follows:

$$\Omega_{\text{W}_j} \geqslant \beta \frac{1}{m} \sum_{i=1}^{m} (\Omega_{\text{W}_i}) \tag{11}$$

Although the weight quantity of a word is the same in documents, the relative importance of each word in a document differs according to the document weight quantity. Therefore, we view the weight ratio rather than the weight quantity as the semantic weight of indexes to a document.

Definition 9 (Weight of a semantic index). The weight of a semantic index, denoted by SW_{W_j}, is defined as follows:

$$SW_{\text{W}_j} = \Psi_{\text{W}_j|\text{T}} \tag{12}$$

4. Experimental results

In this section, we conducted document retrieval experiments in which the proposed method and TF · IDF method were applied to the TREC-2 collection of 1990 Wall Street Journal (WSJ) documents, which comprises 21,705 documents.

The TF · IDF has shown its superior performance for document indexing (Salton & Buckley, 1988). Therefore, this weighting scheme is used as a standard for comparison with the proposed method. We used IDF to both weighting scheme of TF and the proposed indexing for representing the importance degree of the term within a document collection, because the TF and semantic weight only represent the importance

degree of a term within a document, and cannot express the importance degree of the term within a document collection.

As the measurement of the retrieval effectiveness, we used precision and recall as following equations (13) and (14): *Retrieved* is the documents that has been retrieved by the system for a given query. *Relevant* is the documents that is relevant for a given query, in other words, answer documents for a given query.

$$\text{Precision} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Retrieved}|} \tag{13}$$

$$\text{Recall} = \frac{|\text{Relevant} \cap \text{Retrieved}|}{|\text{Relevant}|} \tag{14}$$

For the weights of basic relations (identity, hypernym, etc.), no general guidelines exist except for the research in WordNet (Fellbaum et al., 1998; Kazman et al., 1996), which simply states that the identity relation weight is the highest and the meronym relation weight is the lowest. In the present work, we followed the principle of WordNet when assigning the relation weights. The weights of the five relations used in the clustering of terms were set from 0.1 to 1.5 (identity highest and meronymy lowest).

These experiments were carried out on a Pentium IV 2.8 GHz computer with 1 GByte of RAM. For the relevance judgement of TREC-2 collection, 50 queries from query number 101–150 are supported. However, we only used 40 queries because there are not 1990 WSJ documents for the other 10 queries and we have no answer set to compare the relevance for that 10 queries. The relevance degree between the query and the document was calculated using the vector model. Comparison of the retrieval results obtained in the two sets of experiments showed that the proposed weighting scheme outperforms the traditional weighting scheme.

4.1. Precision and recall

The average precision for 40 queries ranging from Top 1 to Top 20 are shown in Fig. 10. The precision result shows that the proposed system outperforms the TF · IDF weighting method, especially in regard to

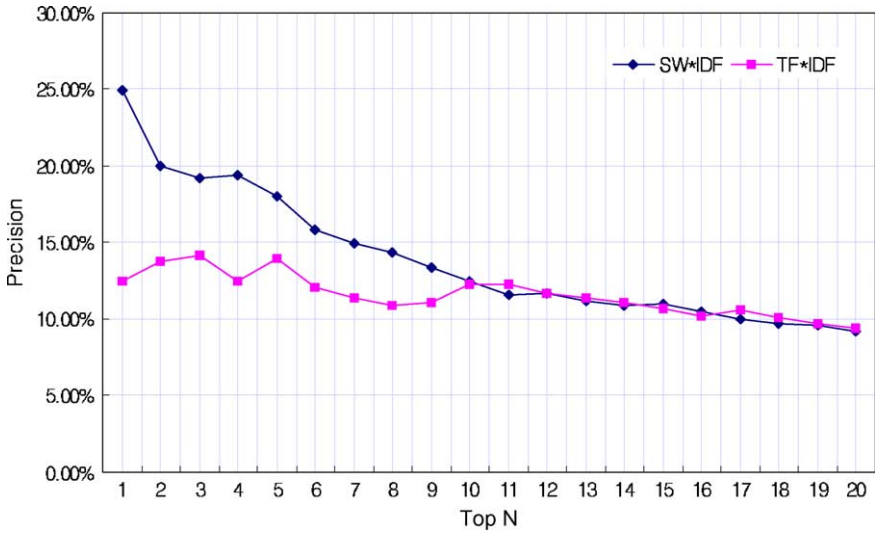

Fig. 10. Average precision results of top *N* documents for 40 queries.

Table 1
Average precision and recall for 40 queries

	Precision		Recall	
	TF · IDF	SW · IDF	TF · IDF	SW · IDF
Top 1	12.50	**25.00**	0.67	**1.53**
Top 5	14.00	**18.00**	8.37	**9.36**
Top 10	12.25	**12.50**	11.27	**12.06**
Top 20	**9.38**	9.25	**18.14**	16.23

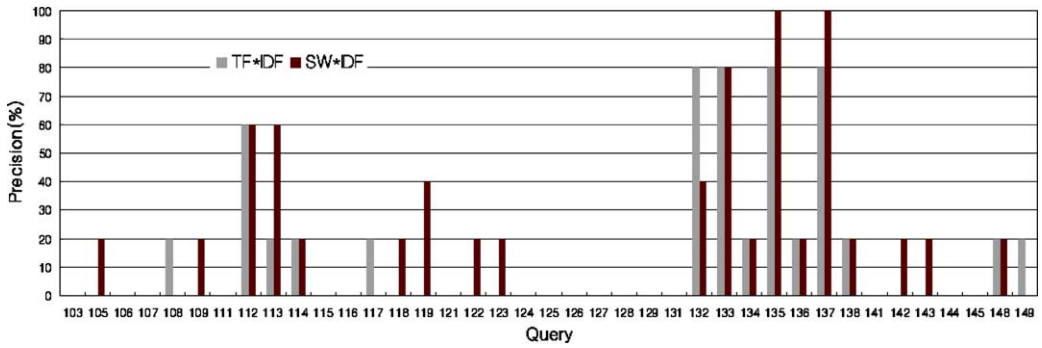

Fig. 11. Relevant documents of top 5 for each query.

the few highest-ranked documents. The average precision of TF · IDF for the 40 queries from Top 1 to Top 20 was 13.90% and that of the proposed method was 11.59%. Therefore, the proposed system (SW · IDF) increased the precision of the traditional TF · IDF method by as much as 2.31%.

Table 1 shows the overall search results, average precision and recall for the Top 1, Top 5, Top 10, and Top 20 documents. The search results show that the proposed system outperforms the TF · IDF weighting method in most of the categories, especially for documents ranked in the top 5 or less. As shown in Table 1, the precision of the proposed system for the top-ranked document is 12.5% higher than that of the TF · IDF system, and the precision of the proposed system for the top 5 documents is 4% higher. Moreover, shown in Fig. 11, the traditional TF · IDF method does not give any relevant results for 26 of the 40 queries (i.e., 65% of total queries) for Top 5, whereas the proposed method does not give any relevant results for 21 queries (i.e., 52.5%).

When users search the Web for information, they tend to focus on the document with the highest ranking. Thus, the relevance of the highest-ranked document plays an important role in user satisfaction. Moreover, many commercial search systems show only the top 10 or so relevant documents on the first result web page, and users tend not to look beyond this page. Therefore, the superior search performance of the proposed system compared to the traditional TF · IDF weighting scheme indicates that the proposed scheme should give enhanced user satisfaction in a commercial setting.

4.2. Dimension reduction of index terms

The index term dimension is the number of index terms that are used to represent a document. When a document is indexed based on the TF, all the terms in the document are used as indexes, and hence the index term dimension simply equals the number of words in the document. However, when we index a document using the proposed indexing scheme, we first extract representative concepts from the document and

Table 2
Comparison of the dimension and size of index terms

Type	TF · IDF	SW · IDF
Dimension (words)	89.55	17.8
Size (Mbytes)	61.9	12.9

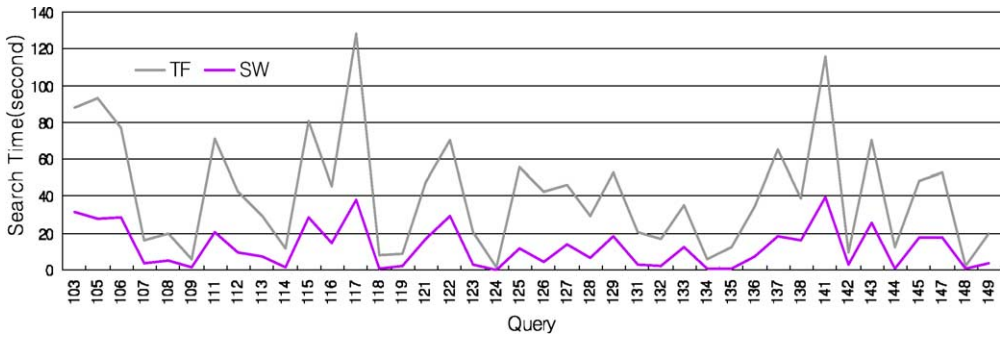

Fig. 12. Overall search time Top 100 document for each query.

then extract index terms from those concepts. Hence, the index term dimension of the document will be less than that obtained using the TF approach. This is clearly demonstrated in the present experiments on the 1990 WSJ documents, for which the average index term dimension was 89.55 using the TF method but 17.8 using the proposed method. Thus, on average the TF method represents the 1990 WSJ documents using about 90 words as index terms whereas the proposed method requires only about 18 words, indicating that the proposed scheme reduces the index term dimension by about 80% compared to the TF method.

When a document is searched, the retrieval system will already have loaded the index file to the memory. When a user inputs a query to the system, it constructs the inverted file appropriate to the query using the index file. As the size of the index file decreases, the time required for loading and construction of the inverted file becomes less, and the overall search time is reduced accordingly. The critical determinant of the index file size is the index term dimension; thus, a reduction in the number of index terms will decrease the size of the indexing file (Table 2).

For the 1990 WSJ documents, the index file size was 61.9 Mbytes using the TF method and 12.9 Mbytes using the proposed method. Loading the TF index file to the main memory required 11.719 s whereas loading that of the proposed method required only 1.75 s, which represents an 85.07% saving in loading time. When we searched the documents for 40 queries using the TF index files, the average search time was 41.3 s; in contrast, the proposed weighting scheme required on average only 12.4 s per search to carry out the same searches. Thus, the search time using the proposed scheme is on average 70.09% less than that using the traditional TF method. Fig. 12 shows the search time required for Top 100 documents of each query using the two methods. In all cases, the proposed method is faster than the TF method. Therefore, the proposed method is expected to enable high-speed searches in the real search environment.

5. Concluding remarks

In this paper, we have presented a novel approach to document searching that uses a concept vector space model to extract and weight indexes. Experiments comparing the proposed approach with results

obtained from real users indicated that the proposed scheme is capable of capturing the differing degrees of semantic importance of words within an overall document. Other experiments in which the proposed approach was compared with the traditional TF method highlighted the superior performance of the proposed scheme, especially in regard to top few documents ranked as most relevant. Importantly, the the index term dimension obtained using the proposed method was 80% less than that obtained using the traditional TF method, which should significantly reduce the search time in a real environment. In sum, the proposed method overcomes some of the limitations of existing stochastic indexing methods and should prove useful in a commercial setting.

References

Barzilay, R., & Elhadad, M. (1997). Using lexical chains for text summarization. In *Proceedings of the ACL'97* workshop on intelligent scalable text summarization.

Budanitsky, A. (1999). Lexical semantic relatedness and its application in natural language processing. Technical Report CSRG-390. University of Toronto.

Carthy, J. (2002). *Lexical chains for topic tracking*. Ph.D. Thesis. Ireland: University College Dublin.

Fellbaum, C. et al. (1998). *WordNet: An eletroic lexical database*. The MIT press.

Fuhr, N., & Buckley, C. (1991). A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems, 9*(3), 223–248.

Hirst, G., & St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Christiane Fellbaum (Ed.), *WordNet: An electronic lexical database*. Cambridge, MA: The MIT Press.

Kazman, R., Al-Halimi, R., Hunt, W., & Mantei, M. (1996). Four paradigms for indexing video conferences. *IEEE Multimedia, 3*(1), 63–73.

Kleinberg, J., & Lawrence, S. (2001). The structure of the Web. *Science, 294*, 1849–1850.

Kominek, J., & Kazman, R. (1997). Accessing multimedia through concept clustering. In *Proceedings of CHI'97* (pp. 19–26).

Lee, J. H. (1995). Combining multiple evidence from different properties of weighting schemes. In *Proceedings of the 18th SIGIR Conference* (pp. 180–188).

Luhn, H. P. (1957). Statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development, 1*(4), 309–317.

Moens, M.-F. (2000). *Automatic indexing and abstracting of document texts*. Kluwer Academic Publishers.

Morris, J. (1988). *Lexical cohesion, the thesaurus, and the structure of text*. Master's thesis. Department of Computer Science, University of Toronto.

Morris, J., & Hirst, G. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics, 17*(1), 21–43.

Salton, G. (1975). *A theory of indexing*. Bristol, UK.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing and Management, 24*(5), 513–523.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York.

Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*(1), 11–21.

Sparck Jones, K. (1973). Index term weighting. *Information Storage and Retrieval, 9*, 619–633.

Steyvers, M., & Tenenbaum, J. B. (submitted for publication). The large-scale structure of semantic networks: Statistical analyses and a model of semantic networks. *Cognitive Science*. Available from http://www-psych.stanford.edu/~msteyver.