

## RANKING DOCUMENTS IN THESAURUS-BASED BOOLEAN RETRIEVAL SYSTEMS

JOON HO LEE, MYOUNG HO KIM, and YOON JOON LEE

Department of Computer Science, Korea Advanced Institute of Science and Technology,  
373-1, Kusung-dong, Yusung-gu, Taejon, 305-701, Korea

(Received 18 February 1992; accepted in final form 30 November 1992)

**Abstract**—In this paper we investigate document ranking methods in thesaurus-based boolean retrieval systems, and propose a new thesaurus-based ranking algorithm called the Extended Relevance (E-Relevance) algorithm. The E-Relevance algorithm integrates the extended boolean model and the thesaurus-based relevance algorithm. Since the E-Relevance algorithm has all the desirable properties of the extended boolean model, it avoids the various problems of previous thesaurus-based ranking algorithms. The E-Relevance algorithm also ranks documents effectively by using term dependence information from the thesaurus. We have shown through performance comparison that the proposed algorithm achieves higher retrieval effectiveness than the others proposed earlier.

**Keywords:** Information retrieval, Boolean retrieval system, Ranking algorithm, Thesaurus.

### 1. INTRODUCTION

An Information Retrieval (IR) system provides users with relevant references that satisfy their information needs. The main objective of IR systems, however, is not just to present relevant references, but to aid in determining which documents are most likely to be relevant to the given requirements. The IR system should provide a sequence of documents ranked in decreasing order of query-document similarity.

Boolean retrieval systems have been most widely used among commercially available IR systems. This is because high performance is achievable and users are able to express their information needs conveniently by using a boolean query formulation. In conventional boolean retrieval systems, however, document ranking is not supported and similarity coefficients cannot be computed between queries and documents. There have been many works in the past to overcome this problem (Buell, 1981; Croft, 1986; Noreault *et al.*, 1977; Radecki, 1982; Salton *et al.*, 1983).

Thesaurus usage is an important component of many IR systems (Svenonius, 1986), and a particular methodology of document ranking has been applied to this case—as will be explained and extended in this paper. The thesaurus, which is a kind of knowledge base, consists of nodes and edges. Nodes are concepts rather than words and edges represent the binary relationships, such as broader-term, narrower-term, synonym, and related term. IR systems based on the thesaurus have several advantages. First, since index terms are selected from the thesaurus, they do not need to match terminology in documents. Documents on the same topic can be retrieved by the same thesaurus terms regardless of terminology in the documents. Second, retrieval effectiveness\* of IR systems can be improved by using term dependence information. The edges of the thesaurus represent term dependencies more exactly than the conventional statistical measures (Giger, 1988).

There have been a few thesaurus-based ranking algorithms that can be used to rank documents in thesaurus-based boolean retrieval systems (Kim & Kim, 1990; McMath *et al.*, 1989; Rada, Humphrey, & Coccia, 1985; Rada, Humphrey, Suh, Brown, & Coccia, 1985; Rada & Bicknell, 1989; Rada *et al.*, 1989). Though the previous thesaurus-based ranking

\*'Retrieval effectiveness' means the ability to rank documents (i.e., the ability to determine which documents are more relevant to users' information needs).

algorithms provide good retrieval effectiveness in many cases, they have several problems. First, using MIN or MAX functions to evaluate OR operators may produce inappropriate results in certain cases. Second, transforming input boolean queries into minimal disjunctive normal forms increases the complexity of computation. Third, most of them suffer from inefficiency in evaluating NOT operators.

In this paper we propose a new thesaurus-based ranking algorithm called the Extended Relevance (E-Relevance) algorithm. The extended boolean model (Salton *et al.*, 1983, 1985; Salton, 1989) and the thesaurus-based relevance algorithm (Rada, Humphrey, & Coccia, 1985; Rada, Humphrey, Suh, Brown, & Coccia, 1985) are integrated into the E-Relevance algorithm, which measures the similarity between a boolean query and a document. Since E-Relevance has all the desirable properties of the extended boolean model, it avoids the various problems of the previous thesaurus-based ranking algorithms. In addition, E-Relevance ranks documents effectively by using term dependence information from the thesaurus.

The remainder of this paper is organized as follows. Section 2 gives a brief introduction to thesaurus-based boolean retrieval systems. Section 3 describes the previous thesaurus-based ranking algorithms. In section 4 we indicate the various problems pertaining to the previous thesaurus-based ranking algorithms and then propose a new thesaurus-based ranking algorithm. The results of performance comparison are presented in section 5. The concluding remarks are described in section 6.

## 2. THESAURUS-BASED BOOLEAN RETRIEVAL SYSTEMS

### 2.1 The system architecture

The thesaurus-based boolean retrieval system possesses the advantages of both the use of the thesaurus and the boolean retrieval system. The thesaurus-based boolean retrieval system consists of the boolean retrieval subsystem and the ranking subsystem. In principle, it is best to present to users only a few documents ranked at the top after applying ranking algorithms to all documents stored. However, it may not be worthwhile to apply ranking algorithms to a large number of all the documents. Therefore, a small set of documents are identified efficiently by the boolean retrieval subsystem, and then those documents are ranked by the ranking subsystem.

Figure 1 shows the architecture of the thesaurus-based boolean retrieval system. The input documents are first analyzed and the appropriate thesaurus terms for the documents are extracted before they are stored into the document base. The extracted thesaurus terms

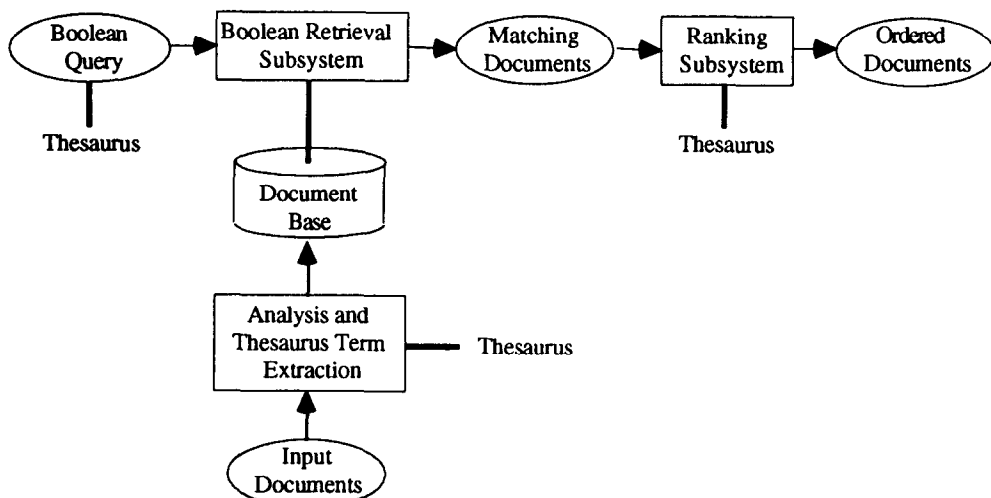


Fig. 1. The thesaurus-based boolean retrieval system.

for a document become the search indexes of that document. Afterwards, documents on the same topic can be retrieved reliably by the same thesaurus terms regardless of terminology used in the documents. The boolean retrieval subsystem receives boolean queries, which are logical expressions composed of thesaurus terms and logical operators AND, OR, and NOT. The ranking subsystem takes those documents retrieved by the boolean retrieval subsystem, and ranks them in decreasing order of query-document similarity. Term dependence information from the thesaurus can also play a crucial role in improving the quality of the ranking algorithm.

## 2.2 The thesaurus

The thesaurus is a means for describing the subject matter in a document-independent way. It is also a useful form of knowledge representation by linking concepts into a network. It consists of nodes that are concepts rather than words, and edges that represent the binary relationships, such as broader-term, narrower-term, synonym, and related term. That is, the knowledge is embodied in the nodes and edges of the thesaurus. Other terminologies often used to denote the thesaurus are "classification structure," "controlled vocabulary," and "ordering systems."

There are several hierarchical thesauruses used in conventional IR systems. For example, the Medical Subject Headings (MeSH) is used in the MEDLINE system (McCarn, 1980). MeSH contains approximately 15,000 indexing terms arranged in a hierarchical structure of depth nine. If the associated synonyms are considered, there are over 100,000 terms in total. The Computing Reviews Classification Structure (CRCS) is another hierarchical thesaurus maintained by the Association for Computing Machinery for indexing its publications (Sammet & Ralston, 1982). CRCS is strictly hierarchical, and no term has more than one parent. CRCS has about 1,000 terms with depth five. Both MeSH and CRCS represent 'is-a' or 'generalization' relationships between thesaurus terms. Figure 2 shows a part of CRCS. In the figure  $H.i_1 \dots i_n$  conveys the structural information as well as the depth of the given term. For example, 'H.3.2 Information Storage' represents that it is a subconcept of 'H.3 Information Storage and Retrieval', which is also a subconcept of 'H Information Systems'.

## 3. THESAURUS-BASED RANKING ALGORITHMS

Since a major role of IR systems is to generate a ranked output of documents rather than a set of documents, the ranking algorithm is an important component of IR systems.

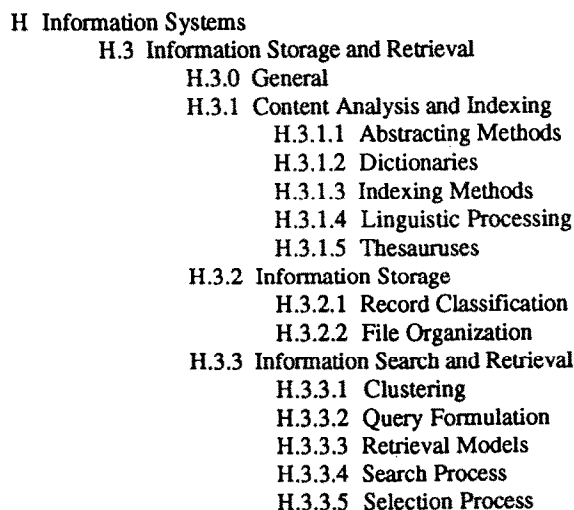


Fig. 2. A portion of the Computing Reviews Classification Structure.

Users are able to minimize their time spent to find useful information by reading the top-ranked documents first. In this section we review previous thesaurus-based ranking algorithms.

The following thesaurus-based ranking algorithms have been developed in the past. They use 'is-a' relationships from the thesaurus to calculate the conceptual closeness or the conceptual distance between a boolean query and a document.

- Relevance Algorithm (Relevance) (Rada, Humphrey, & Coccia, 1985; Rada, Humphrey, Suh, Brown, & Coccia, 1985)
- Distance Algorithm (R-Distance) (McMath *et al.*, 1989; Rada & Bicknell, 1989; Rada *et al.*, 1989)
- Distance Algorithm (K-Distance) (Kim & Kim, 1990)

In the Relevance, R-Distance, and K-Distance algorithms, a query is initially a logical expression consisting of thesaurus terms and logical operators AND, OR, and NOT. The expression is then converted into the minimal disjunctive normal form (DNF) by, for example, the Quine-McCluskey algorithm (McCluskey, 1956). Hence, the query can be viewed as a disjunction of conjunctive terms, where each conjunction may contain negated terms. A valid boolean query is expressed in DNF as follows:

$$Q = \text{Con}_1(Q) \text{ OR } \text{Con}_2(Q) \text{ OR } \dots \text{ OR } \text{Con}_p(Q) = \bigvee_{i=1}^p \text{Con}_i(Q)$$

$$\text{Con}_i(Q) = L_{i1} \text{ AND } L_{i2} \text{ AND } \dots \text{ AND } L_{im_i} = \bigwedge_{j=1}^{m_i} L_{ij}$$

where  $L_{ij}$ , which is the  $j$ th literal in the  $i$ th conjunction, is either a positive or negated term,  $m_i$  is the number of literals in the  $i$ th conjunction, and  $p$  is the number of conjunctions in the DNF query.

The previous thesaurus-based ranking algorithms are on the basis of the topological distance between two terms in the thesaurus, where edges represent the 'is-a' relationships such as broader-term and narrower-term. The topological distance is defined as the minimal number of 'is-a' relationships that must be traversed in the thesaurus from one term to the other. That is, the distance (often called the primitive distance function) between terms  $T_i$  and  $T_j$  is as follows:

$$\text{distance}(T_i, T_j) = \text{minimum number of 'is-a' relationships between } T_i \text{ and } T_j.$$

The related formulas for the Relevance algorithm are shown in Fig. 3, where a document  $D$  indexed with  $n$  terms is represented as  $D = T_1 \text{ AND } \dots \text{ AND } T_n$ . The Relevance algorithm computes how conceptually close a document is to a query. For each conjunction, it first computes the relevance of a document to that conjunction. The maximum value among those obtained from all the conjunctions becomes the final relevance of a document to a query. To calculate the relevance of a document to a conjunction, all pairwise combinations between literals in the conjunction and terms in the document are generated. For each pairwise combination, the Relevance algorithm calculates the relevance between a literal and a term by using the primitive distance function. Since the denominator in the calculation of the relevance between a conjunction and a document gives the maximum possible unnormalized value, the final relevance value will always fall between 0 and 1.

The R-Distance algorithm calculates how conceptually far a document is from a query. It first calculates the distance between each conjunction and a document. The final distance is the minimum value over those values obtained from all the conjunctions. To calculate the distance between a conjunction and a document, all pairwise combinations between literals in the conjunction and terms in the document are generated. For each pairwise combination, the R-Distance algorithm calculates the distance between a literal and a term by

$$\begin{aligned}
\text{RELEVANCE}(Q, D) &= \text{RELEVANCE}(\text{Con}_1 \text{ OR } \dots \text{ OR } \text{Con}_p, D) \\
&= \text{MAX}_{i=1, \dots, p} \text{Relevance}(\text{Con}_i, D)
\end{aligned}$$

$$\text{Relevance}(\text{Con}_i, D) = \text{Relevance}(L_{i1} \text{ AND } \dots \text{ AND } L_{im_i}, T_1 \text{ AND } \dots \text{ AND } T_n)$$

$$\begin{aligned}
&= \frac{\sum_{j=1}^{m_i} \sum_{k=1}^n \text{relevance}(L_{ij}, T_k)}{\text{MIN}(m_i, n) + \frac{1}{2}(m_i \cdot n - \text{MIN}(m_i, n))} \\
\text{relevance}(L_{ij}, T_k) &= \begin{cases} \frac{1}{1 + \text{distance}(T_{ij}, T_k)} & \text{if } L_{ij} \text{ is } T_{ij} \\ \frac{-1}{1 + \text{distance}(T_{ij}, T_k)} & \text{if } L_{ij} \text{ is NOT } T_{ij} \end{cases}
\end{aligned}$$

Fig. 3. Formulas related with the Relevance algorithm.

using the primitive distance function. Dividing the double sum by the product  $m_i \cdot n$  is for normalization purposes. For the computation of the conceptual distance between NOT  $T_{ij}$  and  $T_k$ , NOT  $T_{ij}$  is substituted by the set  $T_{ij}^{-1}$ , which is defined as the farthest nodes from  $T_{ij}$  within the thesaurus. Dividing the sum by the  $|T_{ij}^{-1}|$ , which denotes the cardinality of the set  $T_{ij}^{-1}$ , is for normalization. The related formulas for the R-Distance algorithm are shown in Fig. 4.

The K-Distance algorithm computes how conceptually far a document is from a query like the R-Distance algorithm. R-Distance satisfies the properties of a metric, which are

$$\begin{aligned}
\text{DISTANCE}(Q, D) &= \text{DISTANCE}(\text{Con}_1 \text{ OR } \dots \text{ OR } \text{Con}_p, D) \\
&= \text{MIN}_{i=1, \dots, p} \text{Distance}(\text{Con}_i, D)
\end{aligned}$$

$$\text{Distance}(\text{Con}_i, D) = \text{Distance}(L_{i1} \text{ AND } \dots \text{ AND } L_{im_i}, T_1 \text{ AND } \dots \text{ AND } T_n)$$

$$= \begin{cases} \frac{1}{m_i \cdot n} \sum_{j=1}^{m_i} \sum_{k=1}^n \text{distance}(L_{ij}, T_k) & \text{if } \text{Con}_i \neq D \\ 0 & \text{if } \text{Con}_i = D \end{cases}$$

$$\text{distance}(L_{ij}, T_k) = \begin{cases} \text{distance}(T_{ij}, T_k) & \text{if } L_{ij} \text{ is } T_{ij} \\ \frac{1}{|T_{ij}^{-1}|} \sum_{T \in T_{ij}^{-1}} \text{distance}(T, T_k) & \text{if } L_{ij} \text{ is NOT } T_{ij} \end{cases}$$

$$T_{ij}^{-1} = \left\{ X \in V \mid \text{distance}(T_{ij}, X) = \text{MAX}_{Y \in V} \text{distance}(T_{ij}, Y) \right\},$$

where  $V$  is the set of terms of a thesaurus

Fig. 4. Formulas related with the R-Distance algorithm.

zero, symmetric, positive, and triangular inequality properties (Rada *et al.*, 1989). The zero property, however, is only achieved by forcing  $\text{Distance}(\text{Con}_i, D)$  to zero when  $\text{Con}_i$  is equal to  $D$ . (Note the definition of  $\text{Distance}(\text{Con}_i, D)$  given in Fig. 4.) This results in the discontinuity problem discussed in Kim and Kim (1990). K-Distance overcomes this problem while sacrificing the triangular inequality property. In addition, K-Distance has extended the R-Distance algorithm by allowing edges of the thesaurus, query terms, and document terms to be weighted. It has also reduced the size of the substitution set  $|T_{ij}^{-1}|$ , which is one of the main problems in R-Distance.

#### 4. E-RELEVANCE: A NEW THESAURUS-BASED RANKING ALGORITHM

In this section we propose a new thesaurus-based ranking algorithm called the E-Relevance algorithm. The extended boolean model and the Relevance algorithm are integrated into the E-Relevance algorithm, which measures the similarity between a boolean query and a document. In section 4.1 we describe the problems pertaining to the Relevance, R-Distance, and K-distance algorithms. The extended boolean model, which becomes a part of the proposed algorithm, is briefly described in section 4.2. We present the E-Relevance algorithm in section 4.3.

##### 4.1 Motivation for a new approach

Though the previous thesaurus-based ranking algorithms work well in many cases, they have the following problems.

First, the use of MIN or MAX functions to evaluate OR operators may deteriorate retrieval effectiveness. The use of MIN or MAX functions, which seems to stem from the fuzzy set theory, has been criticized because the rank of a document depends only on the lowest or highest value (Bookstein, 1980). For example, suppose a query transformed into DNF consists of two conjunctions, and we have two documents  $D1$  and  $D2$  shown below.

$$D1: \text{Relevance}(\text{Con}_1, D1) = 0.50, \text{Relevance}(\text{Con}_2, D1) = 0.50.$$

$$D2: \text{Relevance}(\text{Con}_1, D2) = 0.10, \text{Relevance}(\text{Con}_2, D2) = 0.51.$$

Here,  $\text{Con}_1$  and  $\text{Con}_2$  are conjunctions of the query, and the associated number denotes the value of the function  $\text{Relevance}(\text{Con}_i, D_i)$ . Since in the Relevance algorithm the slight increase in the value of  $\text{Relevance}(\text{Con}_2, D_i)$  (i.e., from 0.50 to 0.51) dominates the huge reduction in the value of  $\text{Relevance}(\text{Con}_1, D_i)$  (i.e., from 0.50 to 0.10), the Relevance algorithm gives the document  $D2$  a higher rank than  $D1$ , which clearly is not a desirable output. In other words, the human will obviously decide that  $D1$  is more similar to the given query than  $D2$ .

Second, transforming input boolean queries into DNF increases the complexity of computation. In general, users of the IR system may not obtain all and only the relevant documents in one trial. They repeat by trial and error until a reasonable result is achieved. Users try various queries, which are frequently modifications of the previous queries. Figure 5 shows a typical example, which is the sequence of queries submitted in the boolean retrieval system based on the thesaurus CRCS. Suppose the R-Distance algorithm is applied to the results of boolean queries  $Q1$ ,  $Q3$ , and  $Q4$ . Figure 6 illustrates the procedure to calculate the document value of a document  $D$ . Although  $Q4$  is defined with  $Q1$  and  $Q3$ , the values of  $\text{R-DISTANCE}(Q1, D)$  and  $\text{R-DISTANCE}(Q3, D)$  cannot be used directly in calculating  $\text{R-DISTANCE}(Q4, D)$ .  $\text{R-DISTANCE}(Q4, D)$  repeats many redundant calculations done in calculating  $\text{R-DISTANCE}(Q1, D)$  and  $\text{R-DISTANCE}(Q3, D)$ . For example, in Fig. 6d the eight primitive distance functions are redundantly repeated for each document retrieved by  $Q4$  to compute  $\text{R-DISTANCE}(Q4, D)$ . This problem cannot be avoided in the previous thesaurus-based ranking algorithms because it is due to transforming input boolean queries into DNF.

Third, NOT operations of the R-Distance and K-Distance algorithms are very inefficient. For the computation of the conceptual distance between NOT  $T_i$  and  $T_j$ , those dis-

- Q1: Retrieval Models
- Q2: Query Formulation AND Thesauruses
- Q3: Search Process
- Q4: (1: OR 3:) AND Thesauruses
- Q5: 4: AND NOT Clustering

Qi denotes the query number. Q1 is modified in Q4 and Q4 is modified in Q5. Users sometimes write a query Q2 which is not used as a component of later searches.

Fig. 5. An example sequence of boolean queries.

tance algorithms substitute NOT  $T_i$  by the set  $T_i^{-1}$ , the size of which is greatly increased according to the number of terms contained in the thesaurus. Table 1 shows the size of the substitution set for each negated term to calculate the conceptual distances for queries in Fig. 9 (Kim & Kim, 1990). The queries are part of the test data to evaluate the retrieval effectiveness of our ranking algorithm. The queries are represented with the terms from CRCS consisting of only 1,000 terms. Thus, for the thesaurus consisting of a large number of terms, the performance of those distance algorithms degrades significantly when many queries having negated terms are submitted.

4.2 The extended boolean model

The extended boolean model represents a unifying retrieval model in which the conventional boolean model, the fuzzy set model, and the vector space model are special cases

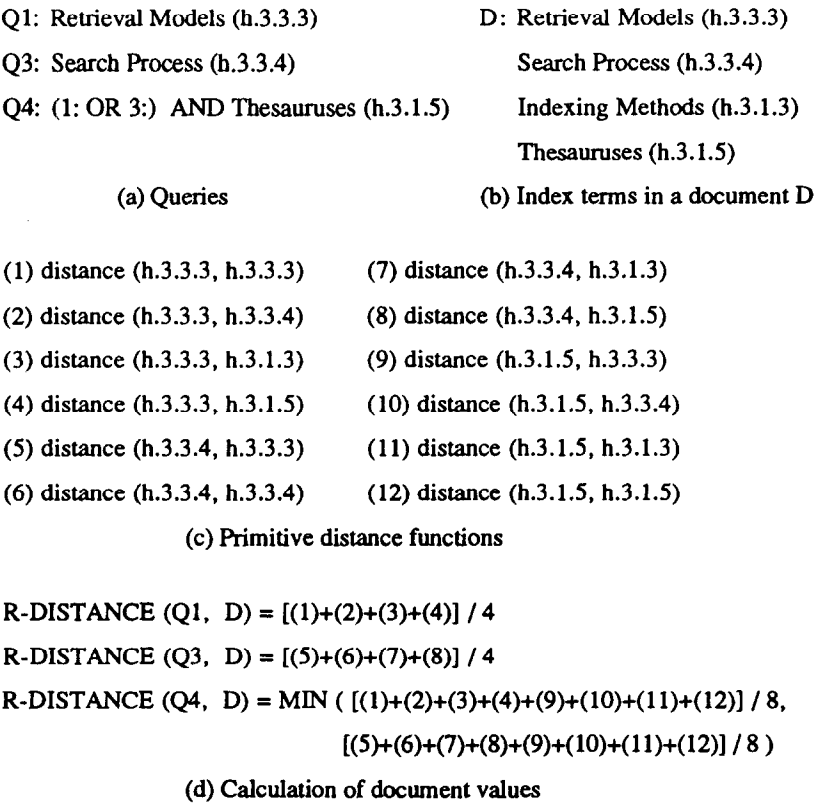


Fig. 6. Calculation of document values in the R-Distance algorithm.

Table 1. The size of the substitution set for a negated term

|            | Query 5 | Query 6 | Query 7 | Query 8 | Query 9 |
|------------|---------|---------|---------|---------|---------|
| R-Distance | 495     | 495     | 495     | 516     | 495     |
| K-Distance | 47      | 46      | 46      | 53      | 46      |

(Salton *et al.*, 1983). The query-document similarity defined in the extended boolean model is based on  $L_p$  vector norm computations, and is controlled by a parameter  $p$ ,  $1 \leq p \leq \infty$ , providing a special interpretation for the boolean operators. As the value of  $p$  decreases, the interpretation of the boolean operators is relaxed more and more. The distinction between the boolean OR and AND operators is lost completely when the  $p$  value reaches its lower limit.

In order to improve retrieval effectiveness, the various types of weights, such as document term weights, query term weights, and query clause weights have been proposed in the literature. In general, all weights are limited to the range from 0 to 1. The document term weight reflects the importance of the individual term attached to the document. A document  $D$  is represented by  $((T_1, d_1), \dots, (T_n, d_n))$  where  $d_i$  designated the weight of term  $T_i$  in  $D$ . The query term weight and query clause weight reflect the importance of the individual term and clause in the query. A BNF grammar that specifies weighted boolean queries is shown in Fig. 7 (since the NOT operator is only used to exclude some documents of those retrieved, only AND NOT is meaningful). A query  $Q$  is represented by  $((T_1, q_1) \text{ OR } (T_2, q_2)), q_{1\text{OR}2}) \text{ AND } (T_3, q_3)$  where  $q_i$  is the weight of term  $T_i$  in  $Q$  and  $q_{1\text{OR}2}$  is the weight of clause  $(T_1 \text{ OR } T_2)$ .

The extended boolean model provides a way of evaluating those weights such as the document term, query term, and query clause weights. The similarity between a weighted boolean query and a weighted document can be calculated by applying the expressions (1)–(4) given below, recursively. For example, to obtain the document value with respect to a non-weighted boolean query\* such as  $((T_1 \text{ OR } T_2) \text{ AND } T_3)$ , we first find the individual weights of terms  $T_1$ ,  $T_2$ , and  $T_3$  in the document (i.e.,  $d_1$ ,  $d_2$ , and  $d_3$ ). We then proceed to find the document value with respect to the clause  $(T_1 \text{ OR } T_2)$ . Finally, the document value for the complete query  $((T_1 \text{ OR } T_2) \text{ AND } T_3)$  is found.

$$\begin{aligned} & \text{Sim}((LQ, q_{LQ}) \text{ AND } (RQ, q_{RQ}), D) \\ &= 1 - \left[ \frac{q_{LQ}^p (1 - \text{Sim}(LQ, D))^p + q_{RQ}^p (1 - \text{Sim}(RQ, D))^p}{q_{LQ}^p + q_{RQ}^p} \right]^{1/p}. \end{aligned} \quad (1)$$

$$\text{Sim}((LQ, q_{LQ}) \text{ OR } (RQ, q_{RQ}), D) = \left[ \frac{q_{LQ}^p \text{Sim}(LQ, D)^p + q_{RQ}^p \text{Sim}(RQ, D)^p}{q_{LQ}^p + q_{RQ}^p} \right]^{1/p}. \quad (2)$$

\*The non-weighted boolean query is equivalent to the weighted query in which all the term and clause weights are 1.

```

Query = (LQ, qLQ) AND (RQ, qRQ) |
        (LQ, qLQ) OR (RQ, qRQ) |
        (LQ, qLQ) AND NOT (RQ, qRQ) |
Term
LQ =   Query
RQ =   Query

```

Fig. 7. BNF grammar for weighted boolean queries.

$$\text{Sim}((LQ, q_{LQ}) \text{ AND NOT}(RQ, q_{RQ}), D) = q_{LQ} \text{Sim}(LQ, D) \cdot (1 - q_{RQ} \text{Sim}(RQ, D)). \quad (3)$$

$$\text{Sim}(T, D) = \text{the weight of the term } T \text{ in the document } D. \quad (4)$$

In the extended boolean model all the problems of the previous thesaurus-based ranking algorithms do not occur. We describe the reasons in the rest of this section.

The problem in evaluating the OR operator can be avoided. As mentioned before, the extended boolean model includes the fuzzy set model as a special case. As the  $p$  value moves from 1 to  $\infty$ , the boolean operators AND and OR are interpreted more and more strictly. When  $p$  goes to infinity, the extended boolean model becomes identical to the fuzzy set model. Note that in the fuzzy set model MIN and MAX functions are used to evaluate AND and OR operators, which is shown to be undesirable. Since the  $p$  value smaller than 10 is mostly used in the extended boolean model, the problem incurred by the use of MIN or MAX functions can be avoided.

Redundant calculations, which arise from transforming input boolean queries into DNF, are eliminated. Given the queries in Fig. 5, the similarity between  $Q4$  and a document  $D$  is defined with  $\text{Sim}(Q1, D)$  and  $\text{Sim}(Q3, D)$  as follows:

$$\begin{aligned} \text{Sim}(Q4, D) = 1 - [(1 - (\text{Sim}(Q1, D)^p + \text{Sim}(Q3, D)^p)^{1/p})^p \\ + (1 - \text{Sim}(\text{Thesauruses}, D))^p]^{1/p}. \end{aligned}$$

Note that when computing  $\text{Sim}(Q4, D)$ , the terms included in  $Q1$  and  $Q3$  need not be considered. In other words, the values of  $\text{Sim}(Q1, D)$  and  $\text{Sim}(Q3, D)$ , which are computed beforehand, can be used in computing  $\text{Sim}(Q4, D)$ .

The problem of inefficient evaluation of NOT operators in the distance algorithms, which is caused by a large substitution set for a negated term, is avoided. In the extended boolean model, the similarity between each term in a query and a document is calculated independently of the clause to which that term belongs. That is, since the similarity between each term in a query and a document is calculated first and then the operators are applied, the problem arising from the large substitution set does not occur.

#### 4.3 The E-Relevance algorithm

We have described how the extended boolean model avoids those problems occurring in the previous thesaurus-based ranking algorithms. However, the definition of the similarity between a single term  $T$  and a document  $D$  (i.e.,  $\text{Sim}(T, D)$  shown in the expression (4)) has some problems. First, term dependence information is not used in calculating  $\text{Sim}(T, D)$ . Though document  $D$  may contain some other terms, the extended boolean model does not take advantage of term dependence information between  $T$  and those terms. Second, when document term weights are restricted to 0 and 1, it is difficult to rank the documents, because only a few document values are generated.

We propose the E-Relevance algorithm, which eliminates the defects of the extended boolean model. The E-Relevance algorithm uses term dependence information in computing  $\text{Sim}(T, D)$  by exploiting the Relevance algorithm, which can only compute similarity between a non-weighted boolean query and a non-weighted document. When a document  $D$  is indexed with only thesaurus terms  $(T_1, \dots, T_n)$ , the Relevance algorithm computes the similarity between a single term  $T$  and the document  $D$  (i.e.,  $\text{Relevance}(T, D)$ ) as follows:

$$\begin{aligned} \text{Relevance}(T, D) &= \frac{\sum_{i=1}^n \text{relevance}(T, T_i)}{1 + \frac{1}{2}(n-1)} \\ \text{relevance}(T, T_i) &= \frac{1}{1 + \text{distance}(T, T_i)} \end{aligned}$$

$\text{distance}(T, T_i) = \text{minimum number of 'is-a' relationships between } T \text{ and } T_i.$

Note that these expressions can be obtained from the formulations in Fig. 3, when  $Q = (T)$ .

In the above expressions, the function  $\text{Relevance}(T, D)$  computes how similar the term  $T$  is to the document  $D$ . Hence, when documents are not weighted, we can observe that  $\text{Relevance}(T, D)$  shown above has a better semantics of similarity in computing  $\text{Sim}(T, D)$  than the extended boolean model in which  $\text{Sim}(T, D)$  is either 0 or 1. We extend this idea into the weighted document environment as follows. Suppose that a document  $D$  is represented by pairs of a term  $T_i$  and a weight  $d_i$  (i.e.,  $((T_1, d_1), \dots, (T_n, d_n))$  (where  $d_i$  is the value between 0 and 1)). Since non-weighted documents imply that the weight for each document term is 1, we insist that the similarity between terms  $T$  and  $T_i$  should be defined as the multiplication of the function  $\text{relevance}(T, T_i)$  and the weight for the document term  $T_i$  (i.e.,  $d_i$ ). Consequently, our proposed definition of  $\text{Sim}(T, D)$  is as follows:

$$\text{Sim}(T, D) = \frac{\sum_{i=1}^n (\text{relevance}(T, T_i) \cdot d_i)}{1 + \frac{1}{2}(n - 1)}. \quad (4')$$

Our definition of  $\text{Sim}(T, D)$  improves the extended boolean model in that term dependence information is effectively used. In addition, even when document term weights are restricted to 0 and 1, document values are not limited to a few values, as in the extended boolean model.

The overall procedures of the E-Relevance algorithm are as follows, where documents are represented by pairs of a thesaurus term and a weight (i.e.,  $((T_1, d_1), \dots, (T_n, d_n))$ ). When a weighted boolean query according to the rule in Fig. 7 is given, the E-Relevance algorithm first computes  $\text{Sim}(T, D)$  between each term  $T$  in a query and a document  $D$  by using the expression (4'). The final document value is then calculated by applying the expression (1)–(3) given in section 4.2, recursively. The E-Relevance algorithm has all the desirable properties of the extended boolean model, while avoiding the various problems of the extended boolean model as well as the previous thesaurus-based ranking algorithms. Documents are also ranked effectively by using term dependence information from the thesaurus.

## 5. PERFORMANCE COMPARISON

In this section experimental results are presented to compare the performance of the E-Relevance algorithm with those of the previous thesaurus-based ranking algorithms, such as Relevance, R-Distance, and K-Distance. The ranking algorithm attempts to simulate a human expert's rankings. Therefore, for the performance evaluation of the ranking algorithm, it is most desirable to compare the performance of the ranking algorithm with that of humans (Kim & Kim, 1990; McMath *et al.*, 1989; Rada, Humphrey, & Coccia, 1985; Rada *et al.*, 1989). Rank correlation methods are used to see how the two rankings are cor-

|   |
|---|
| Query 1: Retrieval Models (h.3.3.3)                   |
| Query 2: Retrieval Models (h.3.3.3) AND               |
| Search Process (h.3.3.4)                              |
| Query 3: Information Search and Retrieval (h.3.3)     |
| Query 4: Information Search and Retrieval (h.3.3) AND |
| Retrieval Models (h.3.3.3)                            |

Fig. 8. Four boolean queries from test data used in McMath *et al.* (1989).

|   |
|---|
| Query 5: Artificial Intelligence (i.2) AND<br>NOT Knowledge Representation Formalisms and Methods (i.2.4)                                     |
| Query 6: Artificial Intelligence (i.2) AND<br>Speech Recognition and Understanding (i.2.7.5) AND<br>NOT Deduction and Theorem Proving (i.2.3) |
| Query 7: Artificial Intelligence (i.2) AND<br>NOT Deduction and Theorem Proving (i.2.3)   |
| Query 8: Artificial Intelligence (i.2) AND<br>Frames and Scripts (i.2.4.1) AND<br>NOT Programming Languages (d.3)                             |
| Query 9: Artificial Intelligence (i.2) AND<br>Deduction and Theorem Proving (i.2.3) AND<br>NOT Applications and Expert Systems (i.2.1)        |

Fig. 9. Five boolean queries from test data used in Kim and Kim (1990).

related. The Spearman correlation coefficient  $\rho$  (Kendall, 1975) used in the evaluation of the previous algorithms is defined as follows:

Given  $k$  entities  $e_1, \dots, e_k$ , the Spearman correlation coefficient  $\rho$  between two rankings  $r_1, \dots, r_k$  and  $r'_1, \dots, r'_k$  is given by

$$\rho = 1 - 6 \times \left( \frac{\sum_{i=1}^k (r'_i - r_i)^2}{k(k^2 - 1)} \right).$$

The coefficient is 1 for identical rankings, 0 for unrelated rankings, and  $-1$  for inversely related rankings.

In order to evaluate the performance of the ranking algorithm, we need boolean queries, a set of documents indexed with thesaurus terms, and human experts' ranking for each pair of a query and a document. The test data used in this paper is from Kim & Kim (1990) and McMath *et al.* (1989). The test data from McMath *et al.* (1989) consists of four boolean queries having only positive terms, nine documents indexed with CRCS terms describing a part of *Information Storage and Retrieval*, and 15 students' ranking judgments. In the test data from Kim & Kim (1990), five boolean queries having negated terms and six articles from *Communications of the ACM* were used. Twenty students' ranking judgments were given and queries and articles were represented with CRCS terms describing a part of *Artificial Intelligence*. The intra-observer reliability and the inter-observer reliability have been tested to be high enough in both experiments. Figures 8 and 9 show all the queries used in the experiments.

Table 2. Spearman correlation coefficients for queries without negated terms

|             | Query 1 | Query 2 | Query 3 | Query 4 |
|-------------|---------|---------|---------|---------|
| Relevance   | 0.867   | 0.783   | 0.933   | 0.800   |
| R-Distance  | 0.879   | 0.742   | 0.842   | 0.783   |
| K-Distance  | 0.867   | 0.833   | 0.904   | 0.817   |
| E-Relevance | 0.867   | 0.817   | 0.933   | 0.867   |

Table 3. Spearman correlation coefficients for queries with negated terms

|             | Query 5 | Query 6 | Query 7 | Query 8 | Query 9 |
|-------------|---------|---------|---------|---------|---------|
| Relevance   | 0.943   | 0.829   | 0.943   | 0.314   | 0.257   |
| R-Distance  | 0.486   | 0.886   | 0.829   | -0.086  | -0.771  |
| K-Distance  | 0.986   | 0.943   | 0.943   | 0.600   | 0.829   |
| E-Relevance | 0.943   | 1.000   | 0.943   | 0.657   | 0.714   |

The E-Relevance algorithm proposed in this paper and the Relevance, R-Distance, and K-Distance algorithms proposed earlier are applied to the test data, and the ranks of the documents are calculated for each query. The Spearman correlation coefficients are computed between the ranks of the ranking algorithms and that of the people. Tables 2 and 3 are the results of performance comparison with the test data from McMath *et al.* (1989) and Kim & Kim (1990), respectively. Table 4 is the summarized statistical data. Since the document values change according to the *p* value in E-Relevance, we have evaluated the performance of the E-Relevance algorithm with increasing the *p* value from 1 to 9. As a result, the best performance is achieved when the *p* value is about six, which approximately coincides with the results of the extended boolean model given in Salton (1989).

Tables 2 and 3 show that the E-Relevance algorithm simulates the human performance very closely, regardless of the existence of negated terms in the query. Table 4 shows that the E-Relevance algorithm achieves the highest retrieval effectiveness of any algorithms proposed earlier in an average case. It also shows that the proposed algorithm has the smallest variance, which indicates that the performance of E-Relevance is most stabilized among all the algorithms proposed so far. The table also shows that the E-Relevance algorithm gives the highest retrieval effectiveness in the worst as well as best cases.

6. CONCLUDING REMARKS

IR systems must be designed to aid users in determining which documents of those retrieved are most likely to be relevant to the given queries. They should also provide efficient mechanisms of computing query-document similarity to cope with a large document base and complex queries arbitrarily posed by IR system users. Although conventional boolean retrieval systems accomplish efficient document retrievals, they suffer from an inability to rank the documents retrieved.

A thesaurus is a kind of knowledge base, which provides controlled vocabularies for describing the subject matter. Since the index terms are selected from the thesaurus, documents can be indexed in a document-independent way. Though there are many IR systems using the thesaurus, term dependence information from the thesaurus has not been effectively used to rank documents.

In this paper we have proposed a new ranking algorithm, called E-Relevance, for thesaurus-based boolean retrieval systems. The E-Relevance algorithm measures the similarity between a boolean query and a document. The E-Relevance algorithm has all the desirable properties of the extended boolean model, while avoiding the various problems of the extended boolean model and the previous thesaurus-based ranking algorithms.

Table 4. Statistical summary of the performance comparison results

|             | Average | Variance | Worst  | Best  |
|-------------|---------|----------|--------|-------|
| Relevance   | 0.741   | 0.070    | 0.257  | 0.943 |
| R-Distance  | 0.510   | 0.327    | -0.771 | 0.886 |
| K-Distance  | 0.858   | 0.013    | 0.600  | 0.986 |
| E-Relevance | 0.860   | 0.013    | 0.657  | 1.000 |

Note that the extended boolean model has problems of not utilizing term dependence information, as well as generating a limited number of document values for non-weighted documents. We have indicated three major problems of the previous thesaurus-based ranking algorithms and shown that the E-Relevance algorithm achieves higher retrieval effectiveness.

*Acknowledgements*—This work was supported in part by Korea Science and Engineering Foundation Grant No. 921-1100-005-1. We would like to thank the referees for many helpful comments.

## REFERENCES

- Bookstein, A. (1980). Fuzzy requests: An approach to weighted boolean searches. *Journal of the American Society for Information Science*, 31(4), 240–247.
- Buell, D.A. (1981). A general model of query processing in information retrieval systems. *Information Processing & Management*, 17(5), 249–262.
- Croft, W.B. (1986). Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science*, 37(2), 71–77.
- Giger, H.P. (1988). *Concept based retrieval in classical IR systems*. Paper presented at the 11th ACM SIGIR Conference, Grenoble, France.
- Kendall, M. (1975). *Rank correlation methods*, 4th Edition 2nd impression. London, High Wycombe: Charles Griffin & Company Ltd.
- Kim, Y.W., & Kim, J.H. (1990). A model of knowledge based information retrieval with hierarchical concept graph. *Journal of Documentation*, 46(2), 113–136.
- McCarn, D.B. (1980). MEDLINE: An introduction to on-line searching. *Journal of the American Society for Information Science*, 31(3), 181–192.
- McCluskey, E.J. (1956). Minimization of boolean functions. *Bell System Technical Journal*, 35(6), 1417–1444.
- McMath, C.F., Tamaru, R.S., & Roda, R. (1989). A graphical thesaurus-based information retrieval system. *International Journal of Man-Machine Studies*, 31(2), 121–147.
- Noreault, T., Koll, M., & McGill, M.J. (1977). Automatic ranked output from boolean searches in SIRE. *Journal of the American Society for Information Science*, 28(6), 333–339.
- Rada, R., Humphrey, S., & Coccia, C. (1985). *A knowledge-base for retrieval evaluation*. Paper presented at the ACM Annual Conference, Denver, CO.
- Rada, R., Humphrey, S., Suh, M., Brown, E., & Coccia, C. (1985). *Relevance on a biomedical classification structure*. Paper presented at the Expert Systems in Government Symposium, Washington, DC.
- Rada, R., & Bicknell, E. (1989). Ranking documents with a thesaurus. *Journal of the American Society for Information Science*, 40(5), 304–310.
- Rada, R., Mill, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1), 17–30.
- Radecki, T. (1982). A probabilistic approach to information retrieval in systems with boolean search request formulations. *Journal of the American Society for Information Science*, 33(6), 365–370.
- Salton, G., Fox, E.A., & Wu, H. (1983). Extended boolean information retrieval. *Communications of the ACM*, 26(11), 1022–1036.
- Salton, G., Fox, E.A., & Voorhees, E. (1985). Advanced feedback methods in information retrieval. *Journal of the American Society for Information Science*, 36(3), 200–210.
- Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Reading, MA: Addison-Wesley.
- Sammet, J.E., & Ralston, A. (1982). The new (1982) computing reviews classification system—final version. *Communications of the ACM*, 25(1), 13–25.
- Svenonius, E. (1986). Unanswered questions in the design of controlled vocabularies. *Journal of the American Society for Information Science*, 37(5), 331–340.