Information Retrieval, 1, 329–344 (2000)
© 2000 Kluwer Academic Publishers. Manufactured in The Netherlands.

The Co-Effects of Query Structure and Expansion on Retrieval Performance in Probabilistic Text Retrieval

JAANA KEKÄLÄINEN*
KALERVO JÄRVELIN†
Department of Information Studies, University of Tampere, Finland

Received April 21, 1999; Revised September 3, 1999; Accepted

Abstract. The effects of query structures and query expansion (QE) on retrieval performance were tested with a best match retrieval system (InQuery[1]). Query structure means the use of operators to express the relations between search keys. Six different structures were tested, representing strong structures (e.g., queries with facets or concepts identified) and weak structures (no concepts identified, a query is 'a bag of search keys'). QE was based on concepts, which were first selected from a searching thesaurus, and then expanded by semantic relationships given in the thesaurus. The expansion levels were (a) no expansion, (b) a synonym expansion, (c) a narrower concept expansion, (d) an associative concept expansion, and (e) a cumulative expansion of all other expansions. With weak structures and Boolean structured queries, QE was not very effective. The best performance was achieved with a combination of a facet structure, where search keys within a facet were treated as instances of one search key (the SYN operator), and the largest expansion.

Keywords: query expansion, query structures, concept-based query formulation, semantic conceptual relationships, text retrieval

1. Introduction

In text retrieval an information need is typically expressed as a set of search keys. In exact match—or Boolean—retrieval relations between search keys in a query are marked with the AND operator, the OR operator, or proximity operators which, in fact, are stricter forms of the AND operator. Thus, the query has a structure based on conjunctions and disjunctions of search keys. These relations vaguely mimic syntagmatic and paradigmatic relations of the syntax of natural language. (Keen 1991, Green 1995.) A query constructed with the Boolean block search strategy (a query in the conjunctive normal form) is an example of a facet structure. Within a facet, search keys representing one aspect of a request are connected with the OR operator, and facets are connected with the AND operator. A facet may consist of one or several concepts.

In best match retrieval, matching means ranking documents according to scores calculated from the weights of search keys occurring in documents. These weights are typically based

*www.uta.fi/∼lijakr/
†www.uta.fi/∼likaja/

on the frequency of a key in a document and on the inverse collection frequency of the documents containing the key (tf.idf weighting). (Ingwersen and Willett 1995.) In best match retrieval, queries may either have a structure similar to Boolean queries, or queries may be 'natural language queries' without differentiated relations between search keys. Query structure refers to the syntactic structure of a query expression, marked with query operators and parentheses. Further, we divide the structures of best match queries into strong and weak. In the former, concepts—and possibly facets—are marked through operators; in the latter, concepts cannot be recognised through syntax.

Query formulation may be based on search keys given by the user, written request, or interaction between the user and intermediary. The original query does not always give satisfactory results, thus it may be reformulated by adding search keys with or without reweighting, which process is known as query expansion (QE briefly). QE has been studied intensively because the selection of search keys is crucial for retrieval performance. (Fidel and Efthimiadis 1995, Efthimiadis 1996.) Query structures have been studied as well, and some evidence of better performance for strongly structured queries in best match retrieval exists (e.g., Belkin et al. 1995, Hull 1997, Turtle 1990), though not in QE studies. Queries with different types of operators, e.g., 'soft' interpretations of the Boolean operators and types of 'probabilistic' operators, have been combined (Rajashekar and Croft 1995, Shaw and Fox 1995). This method leads to rather complicated structures, but improves performance. Because neither the number nor the overlap of search keys in different queries is reported, it is hard to judge the possible interaction of the structure and the number of search keys or QE. Thus the interaction of query structure and QE is an open problem tackled in this paper.

Recently, it has been suggested that retrieval performance could be enhanced by ensuring that *all* search concepts are represented in top ranking documents (e.g., Buckley et al. 1998). Hawking et al. (1997), and Hawking et al. (1997) have tested the forming of concept groups and concept-based scoring. The researchers report on positive effects of these methods on retrieval performance. In the present study we will develop further the idea of concept-based query formulation. We will compare strongly and weakly structured queries. We will analyse the retrieval effectiveness of these structure types in combination with different query expansion types in a probabilistic retrieval system (InQuery).

The interaction and effects of the following variables are tested:

- query structures based on identification of single search keys, phrases, concepts or facets
- QE with different semantic relationships
- the combination of weights in scoring by operators.

In initial query formulation, search concepts are identified from requests and corresponding search keys are elicited from a test thesaurus. In QE, search keys representing concepts that are semantically related to the original search concepts in the test thesaurus are added to queries. All queries, both unexpanded and expanded, are formulated into different structures. This study is not about thesaurus-based QE, rather, it is about the co-effects of query structures and query expansion. The thesaurus is used to provide various kinds of expansions in a controlled way. Our attempt is not to compare this approach to unaugmented human performance.

2. Method

2.1. *Test environment*

The test environment was a text database containing Finnish newspaper articles operated under the InQuery retrieval system (version 3.1). The database contained 54,000 articles published in three Finnish newspapers. The average article length was 233 words, and typical paragraphs were two or three sentences in length. The database index contained all keys in their morphological basic forms. In the basic word form analysis all compound words were split into their component words in their morphological basic forms. For the database there is a collection of 35 requests, which are 1–2 sentences long, in the form of written information need statements. A recall base for these requests consists of 16,540 articles. It is collected by pooling the result sets of hundreds of different queries formulated from the requests in different studies, using both exact and partial match retrieval. For the present study 30 requests were selected on the basis of their expandability, i.e., they provided possibilities for studying the interaction of the test parameters. The number of known relevant documents for the 30 queries is 1,066.

The InQuery system was chosen for the test, because it has a wide range of operators, including probabilistic interpretations of the Boolean operators, and it allows search key weighting. Moreover, InQuery has shown good performance in several tests (e.g., Allan et al. 1997, Harman 1995, Xu and Croft 1996). InQuery is based on Bayesian inference networks (see, Allan et al. 1997, Turtle 1990). All keys are attached with a *belief value*, which is approximated by the following tf.idf modification:

$$0.4 + 0.6 * \left(\frac{tf_{ij}}{tf_{ij} + 0.5 + 1.5 * (dl_j/adl)} \right) * \left(\frac{\log((N + 0.5)/df_i)}{\log(N + 1.0)} \right)$$

where

tf_{ij} = the frequency of the key i in the document j
dl_j = the length of document j (as a number of keys)
adl = average document length in the collection
N = collection size (as a number of documents)
df_i = number of documents containing key i.

The InQuery query language provides a set of operators to specify relations between search keys. As with Boolean operators it is possible to construct facets, and mark relationships between concepts. The probabilistic interpretations for the operators used in this study are given below:

$$P_{and}(Q_1, Q_2, \ldots, Q_n) = p_1 * p_2 * \cdots * p_n$$
$$P_{or}(Q_1, Q_2, \ldots, Q_n) = 1 - (1 - p_1) * (1 - p_2) * \cdots * (1 - p_n)$$
$$P_{sum}(Q_1, Q_2, \ldots, Q_n) = (p_1 + p_2 + \cdots + p_n)/n$$
$$P_{wsum}(w_s, w_1 Q_1, w_2 Q_2, \ldots, w_n Q_n) = \frac{w_s(w_1 p_1 + w_2 p_2 + \cdots + w_n p_n)}{(w_1 + w_2 + \cdots + w_n)}$$

where P denotes probability, Q_i is either a key or an InQuery expression, $p_i, i = 1 \ldots n$, is the belief value of Q_i, w_i, $i = 1 \ldots n$, is the weight of Q_i, and w_s is a weight given for a clause (Rajashekar and Croft 1995, Turtle 1990).

The probability for operands connected by SYN operator is calculated by modifying the tf.idf function as follows:

$$0.4 + 0.6 * \left(\frac{\sum_{i \in S} tf_{ij}}{\sum_{i \in S} tf_{ij} + 0.5 + 1.5 * (dl_j / adl)} \right) * \left(\frac{\log((N + 0.5)/df_s)}{\log(N + 1.0)} \right)$$

where

tf_{ij} = the frequency of the key i in the document j
S = a set of search keys within the SYN operator
dl_j = the length of document j (as a number of keys)
adl = average document length in the collection
N = collection size (as a number of documents)
df_S = number of documents containing at least on key of the set S.

2.2. *Concept based query formulation and expansion*

Three levels—a conceptual, linguistic and string level—can be differentiated in query formulation (UMLS 1994). Differentiating between conceptual and linguistic levels has several advantages: it becomes obvious that a concept—and a facet—may be represented by several expressions; these expressions may be varied according to the search environment; the expressions may be selected from any language; the searchers may just select concepts without having to concern themselves with search keys, which may be attached to each concept through a conceptual model and further automatically formed to a query. Järvelin and others (1996) developed this idea to a thesaurus data model for QE. In the model, concepts and their relations are represented at the conceptual level. Each concept is represented by one or more natural language expressions at the linguistic level. The expressions for each concept are synonyms. At the string level each expression is represented by one or more search strings, which also model, when needed, truncation and phrase component proximity. Matching in IR is done at the string, not at the linguistic level.

We adopted the thesaurus data model for query formulation and expansion. Since the test database includes newspaper articles, we needed a conceptual model for this domain to test QE based on semantic relationships. No such model was available, thus, we chose to construct a test thesaurus. The collection of concepts was started by identifying all concepts from the test requests. Then, for each of these concepts all plausible hierarchically narrower and associatively related concepts were collected. Consideration was given to the completeness of hierarchies. In the organisation of concept relations, concepts were treated independently of the context of the requests. However, the newspaper domain guided the selection of concepts and relations. For each concept, all plausible expressions were gathered, and these expressions were turned into search strings. The thesaurus was constructed by three persons using dictionaries, handbooks, primary literature and their own knowledge. The relations between concepts and between expressions are valid for the whole domain, i.e., they are standard thesaurus or semantic relations. The test thesaurus was aimed at QE in a database of Finnish newspaper articles, thus, its language was Finnish. (Sormunen 1994, Kekäläinen 1999.)

The thesaurus includes 832 concepts, 1,345 expressions for the concepts, and 1,558 search strings for the expressions. Concepts have hierarchic (generic, partitive and instance) and association relationships. Expressions representing concepts are each other's synonyms or quasi-synonyms, i.e., equivalence exists between expressions at the linguistic level. The most typical or obvious of the synonyms, in a linguistic sense, was chosen as a principal expression, *term*, of the concept. Several strings, which are spelling variants, may represent each expression. If these strings are phrases, they are formed with different proximity operators. There is no variation caused by word truncation because the words are in their basic forms in the database index. The thesaurus is a database managed with the ExpansionTool, which is a tool for concept based query construction and expansion (see Järvelin et al. 1996).

In query formulation the researchers identified facets representing different aspects of the request and selected concepts for the facets from the thesaurus in accordance with the request. Concepts were tentatively organised in facets according to a Boolean block search strategy, i.e., concepts representing the same aspect form a disjunctive block or facet, denoted within parentheses by "|".

The facets are combined conjunctively by "&".[2] Real searchers were not involved in the query formulation because this study seeks to test the effect of structural and representation parameters on the retrieval performance, not to find out how real searchers would have used the thesaurus, nor to evaluate the performance of their queries.

In this study the number of facets in a query is called *complexity*, and the total number of concepts in facets is *coverage*. The number of facets was fixed in each request, i.e., the queries included all aspects of the request. The coverage of an unexpanded query was also determined by the request. Coverage was varied in QE by adding semantically related concepts of the original concepts to the query, first narrower and then associative concepts (*conceptual expansion*). At the linguistic level concepts were represented by expressions. In the unexpanded query each concept was represented by its term. In *synonym expansion* all synonymous expressions of the term were added to the query. Then, search strings representing expressions replaced them. All strings corresponding to an expression were always added to the query, thus, there is no expansion at the string level. The total number of search strings in a query is called *broadness*. Although the relation between expressions and strings was not one to one but one to many (1.2 strings per expression, on average), we do not consider the number of expressions separately. Broadness was counted at the string level because matching is based on search strings. Below, we refer to search strings as (search) keys.

The different levels of query formulation and expansion are demonstrated with an example.[3] Let us assume the following request: *The processing and storage of radioactive waste*. At the conceptual level two facets with altogether three concepts are recognised from the request. However, conceptual level is an abstraction, because concepts cannot be discussed without names, thus, when the query is represented as a Boolean block search, it is at the linguistic level, and concepts are represented by terms. A sample conceptual query plan follows:

radioactive waste & (process | storage)

At the string level, search keys replace terms. The query is expressed using the syntax of the query language. Our sample query is first formulated into the Boolean query structure using the query language of InQuery. An unexpanded query, ($Q0$), contains the search concepts selected on the basis of the request. The concepts are represented by terms. The sample unexpanded query is following:[4]

Q0: #**and**(#3(radioactive waste) #**or**(process storage))

With the synonym expansion (Qs) synonyms of the terms are added to the query. The sample query with the synonym expansion follows:

Qs: #**and**(#**or**(#3(radioactive waste) #3(nuclear waste))
#**or**(storage store stock process))

In the narrower concept expansion (Qn) the expressions—terms and their synonyms—of the narrower concepts of the original search concept are added to the query. Thus, synonyms are also included in this expansion. The sample query follows:

Qn: #**and**(#**or**(#3(radioactive waste) #3(nuclear waste)
#3(high active waste) #3(low active waste))
#**or**(storage store stock repository process))

In the next expansion (Qa) the expressions of the associative concepts of the original search concepts are added to the original unexpanded query, that is expressions representing the narrower concepts are not included in this query. The sample query with the associative concept expansion is as follows:

Qa: #**and**(#**or**(#3(radioactive waste) #3(nuclear waste) #3(spent fuel)
#3(fission product)) #**or**(storage store stock process refine))

All the previous expansions are accumulated in the largest expansion (*Qf*). Practically, Qf expansion is a bag of Qn and Qa expansions, because these include all search keys. The sample query with the largest expansion follows:

Qf: #**and**(#**or**(#3(radioactive waste) #3(nuclear waste) #3(high active waste)
#3(low active waste) #3(spent fuel) #3(fission product))
#**or**(storage store stock repository process refine))

In highly agglutinative languages, like Finnish and German, compound words are often spelled together (e.g., *ydinvoimalaitos*, *atomkraftwerk*, which mean *nuclear power plant*). Compound word splitting makes all parts of the word searchable. In the test setting the database index contains both the whole compound words and their parts. Compound word splitting has also a query expansion effect. Many hierarchical relations in Finnish are based on compound words (e.g., *tehdas* → *paperitehdas* → *hienopaperitehdas* which means *factory* → *paper mill* → *paper mill producing fine paper*). When compound words are split, the use of any component word as a search key will match all compound words that include the component. Typically searchers might use heads (like *tehdas/mill*) to get all the narrower concepts. Usually there is no way to control whether the search key matches only heads, or any other compound part, thus, false matches occur. However, as the compound words contain the search key, many of them have the right kind of association to the search key. To some extent compound word splitting causes automatically narrower concept expansion and associative concept expansion to queries. This reduces the effects of these expansion types, but on the other hand the test setting resembles other, less agglutinative languages.

2.3. Query structures

Altogether, 13 query structures were combined with 5 expansion levels resulting in 62 different queries[5] for each of the 30 requests. We have earlier reported of the performance of some structure types (Kekäläinen and Järvelin 1998, Kekäläinen 1999). In this paper we will present two weak structure types: one with single keys and another with phrases; and four strong structure types: one with concepts identified and three with facets identified.

SUM (average of the weights of keys) and *WSUM* (weighted average of the weights of keys) queries represented weak structures. An unexpanded SUM query was constructed of the original concepts of a request, and each concept was represented by a single key or a set of keys corresponding to the term but without phrases. In the expansions, all expressions were added as single words, i.e., no phrases were included. In WSUM queries phrases were identified and expressed with proximity operators. Weighting original keys higher than expansion keys is a usual method in QE (e.g., Wang et al. 1985, Voorhees 1994). In WSUM queries expansion keys and the keys of equivalent expressions (i.e., synonyms for the terms of the original concepts) were given smaller weights (the weight of 1) than the keys of the original concepts (the weight of 2). Examples of these structures with narrower concept expansion are given below:

> *SUM/Qn*
> #**sum**(radioactive waste nuclear waste high active waste low active waste
> storage store stock repository process)
>
> *WSUM/Qn*
> #**wsum**(1 2 #3(radioactive waste) 1 #3(nuclear waste) 1 #3(high active waste)
> 1 #3(low active waste) 2 storage 1 store 1 stock 1 repository 2 process))

In the concept-based SUM-of-synonym-groups-query (*SSYN-C*) each *search concept* formed a clause with the SYN operator. SYN clauses were combined with the SUM operator. All keys within the SYN operator are treated as instances of one key. *SSYN-F* queries were similar to SSYN-C queries, but SYN groups were facets, i.e., all concepts representing one aspect of a request were collected into a group. In *ASYN-F* queries the operator combining facets was the probabilistic AND operator, otherwise queries were similar to SSYN-F queries.

SSYN-C/*Qn*
#**sum**(#**syn**(#3(radioactive waste) #3(nuclear waste) #3(high active waste)
#3(low active waste)) #**syn**(storage store stock repository) #**syn**(process))

SSYN-F/*Qn*
#**sum**(#**syn**(#3(radioactive waste) #3(nuclear waste) #3(high active waste)
#3(low active waste), #**syn**(storage store stock repository process))

ASYN-F/*Qn*
#**and**(#**syn**(#3(radioactive waste) #3(nuclear waste) #3(high active waste)
#3(low active waste), #**syn**(storage store stock repository process))

A query with Boolean operators (*BOOL*) was constructed using the block search strategy. It is a typical facet-based query structure. The interpretations of Boolean operators are given above, i.e., queries were not strict Boolean queries. Earlier studies by Turtle (1990) and Hull (1997) provide evidence that queries with Boolean operators are more effective than weakly structured queries in best match retrieval systems. Examples of Boolean query structure are given above (see Section 2.2).

The structuredness of queries may be analysed by structural features. In this study, phrase identification, search key weighting, concept and facet identification characterised structuredness. Thus, SUM is the weakest structure with single search keys, followed by WSUM with the surplus of phrase identification and search key weighting. SSYN-C forms a middle group since phrases and concepts are identified but no facets. BOOL, SSYN-F and ASYN-F are equal in structuredness because they all include phrases and facets. They differ in operators combining keys and facets.

3. Results

The complexity of the queries ranged from 3 to 5, the average being 3.7. The average coverage of the unexpanded queries was 4.9. The broadness of unexpanded queries when no phrases were marked (i.e., SUM structure) was 6.1 on average, and for expanded queries without phrases, on average, as follows: Qs 18.5, Qn 30.6, Qa 49.5, Qf 62.3. The broadness of queries with phrases was Q0 5.4, Qs 14.1, Qn 24.4, Qa 42.1, Qf 52.4, on average. The number of documents in the result set, or document cut-off value (DCV), was fixed to 50. Average precision was then calculated over 11 cut-off points (1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50). In addition, average precision at 10 recall points (10–100) was calculated, and histograms of the performance of structure types on different queries were drawn.

3.1. DCV precision

With three query structures QE proved beneficial at all expansion levels (SSYN-C, SSYN-F, ASYN-F), and the largest expansion (Qf) was the best (Table 1). The precision scores of

Table 1. Average precision over DCVs 1–50 for the query structures at varying QE levels. ($N = 30$) (NB. The best precision score of each column is in italics, the best precision score of each row is in bold face.)

	Query structures					
QE-levels	SUM	WSUM1	SSYN-C	SSYN-F	ASYN-F	BOOL
Q0	***.477***	.443	.450	.451	.456	*.425*
Qs	.440	*.463*	.505	**.510**	.509	.320
Qn	.407	.440	.503	.519	**.527**	.204
Qa	.393	.461	**.549**	.541	**.544**	.293
Qf	.408	.452	*.555*	***.563***	*.561*	.251

SSYN-C, SSYN-F and ASYN-F queries were very similar, and with QE notably better than precision scores of the other structures. The effect of QE on WSUM queries was mixed: the Qn expansion decreased the precision of WSUM queries, otherwise QE was beneficial. Weighting the original search keys (terms) higher than expansion keys gave middle range results. QE at any level was detrimental for SUM and BOOL queries.

Without expansion the SUM queries were the most effective. This is somewhat unexpected: queries without phrase information gave better precision than the queries with phrases. An explanation might be that a phrase as the *only* expression for a concept was too strict a condition, single search keys were more effective. The splitting of compound words for the database index might also have an effect because single words match to the components of compound words. Another explanation might be that the repetitive search keys, which resulted from relaxing the phrases, were not removed from queries. In InQuery this kind of repetition within SUM or WSUM structures gives extra weight to the repeated keys. However, the advantage was lost with expansions, because expansions without phrases increase the possibility of arbitrary search key combinations in matched documents. All expansions were unbalanced since SUM structure does not take into account the relative importance of the search keys, nor does it recognise concepts or facets.

In the WSUM queries the balance in expansions was sought by weighting original search keys higher than expansion keys, as suggested in literature. The ranking is based on weighted averages of the search key weights. The weighting somewhat supported QE, but the overall performance was not markedly better than the performance of the SUM queries. Thus, the problem of arbitrary search key combinations in matched documents remained.

For the Boolean queries the result can be explained with the interpretation given to the AND and OR operators. If a search key is not present in a document it gets a default weight of 0.4, if the key is present it is given the tf.idf weight. The weight for all keys within the OR operator is calculated as follows: $1 - (1 - p_1) * (1 - p_2) * \cdots * (1 - p_n)$, where p_i is the weight of the search key i. The more search keys the OR clause includes, the higher weight it gets, whether the search keys are present or not. The weights of OR clauses are combined as a product. This leads to ranking where the OR clause or clauses with fewest keys are decisive, i.e., in the shortest OR clauses the presence or absence of search keys becomes decisive irrespective of the importance of the keys for the query.

The three strong SYN structures were related to the Boolean structure, but formulated with different operators. The SYN operator was used instead of the OR operator, and, in two

Table 2. Results of the Friedman test's pairwise comparisons (the expression $x > y$ refers to $p < 0.005$).

SUM	WSUM	BOOL	SSYN-C	SSYN-F	ASYN-F
Q0 > Qn	no sign. diff.	Q0, Qs > Qn	Qa, Qf > Q0	Qa, Qf > Q0	Qa, Qf > Q0

cases (the SSYN structures), the SUM operator instead of the AND operator. It proved that especially replacing the OR by the SYN was useful. The operator combining the concepts or facets was not so decisive. Furthermore, queries with concept-based structure performed almost equally to queries with facet-based structure. The result is not surprising because the number of concepts per facet was not high. Nevertheless, the identification of concepts is easier than the identification of facets, thus the result suggests that facets could be dispersed.

Statistical significance. To decide whether the differences in performance between structure and expansion level combinations are statistically significant, we ran the Friedman test, which is based on ranks, i.e., a non-parametric alternative for comparing more than two related samples (Conover 1980, Hull 1993). After the Friedman test showed that there was a significant difference ($p < 0.005$), pairwise comparisons were conducted to reveal the significant differences between combinations. Table 2 shows the differences when the best expansion was compared against others within each query structure. In all other structures, except WSUM, the difference between the best and the worst combination was significant.

The best expansions for each structure were then compared over the structures. Significant differences were found between the *best* SYN expansions and the following other groups: SSYN-C, SSYN-F, ASYN-F > SUM, WSUM, BOOL.

3.2. Precision at 10 recall levels

Figure 1 visualises the results. It is based on average precision at 10 recall points and shows the worst query structure and expansion combination, and the best expansion of each query structure type. The worst case is the query with Boolean structure with the narrower concept expansion (BOOL/Qn). In the middle there is a group of very similarly behaving structure and expansion combinations, the best Boolean (Q0), SUM (Q0), WSUM (Qs). The best SSYN and ASYN queries are high above others all the way, both at low and high recall.

3.3. Performance of structure and expansion combinations by requests

Reading the results one should bear in mind that none of the structures or expansions was optimal for all requests. To demonstrate the differences in performance between requests, the combinations for each request were ranked according to the DCV precision. The median precision score (over all 6 combinations) for each request was taken as a baseline to which the best combinations were compared. In figures 2–7 the average DCV precision

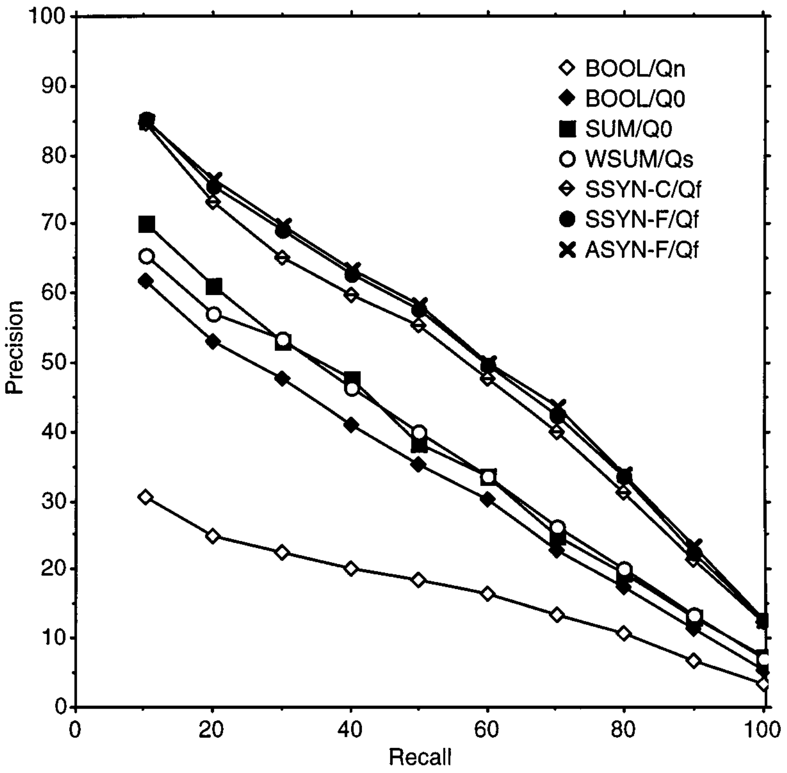

Figure 1. Precision-recall graph of a sample of query structure and QE combinations.

histograms of the best combinations of each structure type are given. These histograms measure the average precision[6] of a combination against the median average precision of all corresponding combinations on that request. These graphs should give an overview of the performance of the query structure types by requests. Although there is variation between requests, the strong SYN structures seldom fall below the median.

4. Discussion and conclusions

We have tested concept-based QE with different query structures in a best match retrieval system. The results show that the effects of QE on retrieval performance depend on the structure of the query. For the performance of weakly structured SUM and facet-structured Boolean queries QE was detrimental. The expanded Boolean queries gave the worst precision scores overall. The effect of QE on WSUM queries varied at different expansion levels, but differences in performance were small ($\leq$2.3 percentage units). With the strong SYN structures QE improved performance and the largest expansion was the best. These combinations achieved the highest average precision scores overall. These

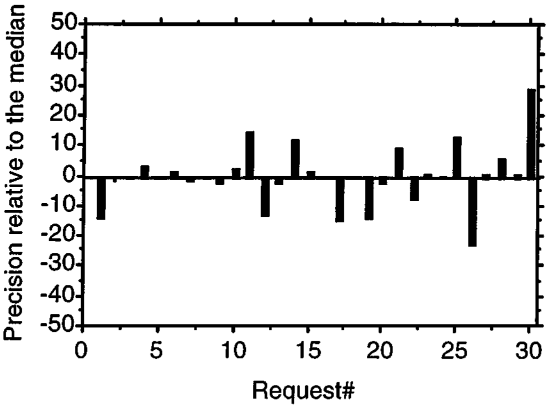

Figure 2. SUM/Q0.

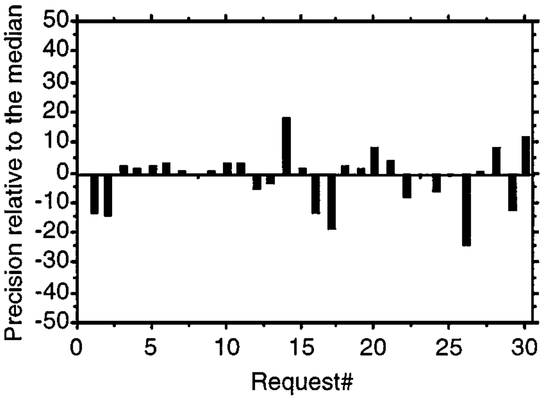

Figure 3. WSUM/Qs.

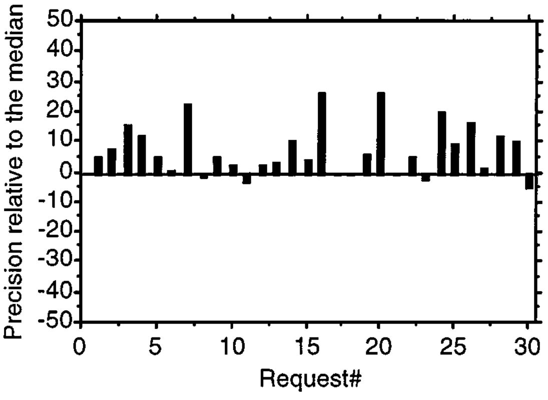

Figure 4. SSYN-C/Qf.

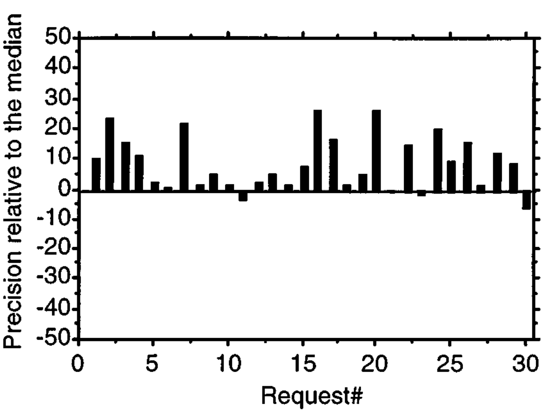

Figure 5. SSYN-F/Qf.

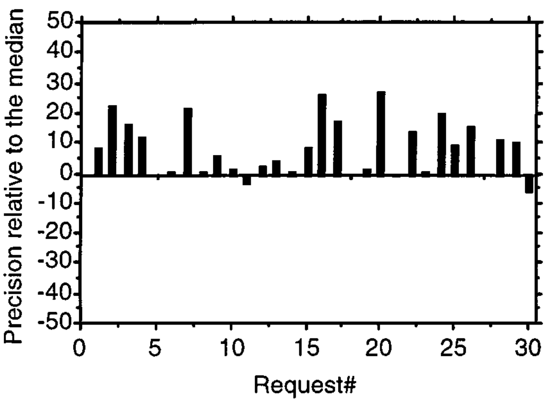

Figure 6. ASYN-F/Qf.

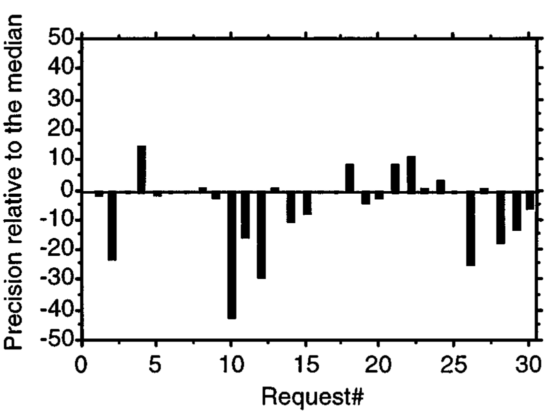

Figure 7. BOOL/Q0.

were also significantly better than the performance of other query structure and expansion combinations. In general, QE interacts with query structure: with a large expansion strong query structures seem necessary, but with a slight or no expansion structure does not matter.

The test thesaurus of the study was rather small-sized. However, the expressions representing concepts were not specially selected for the request topics, neither the concept relations. The growth of the thesaurus would be a crucial matter for the generality of the results because it might rise the number of associative and—to some extent—hierarchical relations. We believe that the growth of the thesaurus would not change the relative QE effectiveness of different query structures, although the number of expansion keys might then rise and the absolute performance change. Pirkola (1998) studied the effects of query structures on retrieval performance in cross-lingual IR environment. In his study, requests were translated using bilingual dictionaries which brought about a QE effect. This may be compared to thesaurus-based expansion. Pirkola showed that the performance of strongly structured translated queries was notably better than the performance of weakly structured counterparts. This result was achieved with translation dictionaries not specially designed for the study, thus it supports our results.

We do not know of any other studies that have systematically examined the effects of query structure and expansion. The forming of concept groups and concept-based scoring tested by Hawking, Thistlewaite and Bailey (1997), and Hawking et al. (1997) is similar to our approach, though different structure types were not compared. The results of these studies confirm our findings.

It seems that the semantic division of relationships—typical for thesauri—was not particularly useful in QE. In most cases the best performance was obtained by the largest expansion including all semantic relationships. When the largest expansion did not work the explanation might be either that the thesaurus did not provide accurate relations and expressions for the concepts, or that the request was very precise, or the vocabulary of the topic was not very variable. However, this problem needs further investigation. Since the largest expansion performed generally best, it seems that any keys semantically associated with the original keys, disregarding their relationship type, should be considered for QE. In statistical thesaurus construction statistically associated keys are identified but the type of the relationship cannot be ascertained. The problem with statistically selected expansion keys might be that the concept or facet boundaries are broken, and thus, the query structure is lost. The relative merits of semantic vs. statistical key associations for QE remain a research issue. In a further study we will compare the expansion keys provided by a statistical and an intellectual thesaurus, as well as the effectiveness of QE based on both types of thesauri.

Acknowledgments

This study was funded by The Finnish Multimedia Programme, KaMu-TILA, The Emil Aaltonen Foundation, The Aamulehti Centenary Fund under the hospices of The Finnish Cultural Foundation, The Pirkanmaa Regional Fund, and The Academy of Finland.

Notes

1. The InQuery software was provided by the Center for Intelligent Information Retrieval, University of Massachusetts, Amherst, MA, USA.
2. We avoid using the Boolean operators here since we are using several different probabilistic operators to form the facets and to combine them.
3. Test queries were in Finnish. The translation of the example may not be exact, but should clarify the idea of the test. The example is shortened, i.e., the expanded sample queries are not complete with all search keys.
4. NB. #3 is a proximity operator in the InQuery query language. The keys within an #N operator must be found within N words of each other in the text in order to contribute to the document's belief value. We used #3 to match all phrases. #and and #or are the probabilistic InQuery operators corresponding to the Boolean conjunction and disjunction.
5. The number of queries per request was 62, not 65, because one weighted structure was only combined with non- and full-expansion.
6. The average precision was calculated over the 11 DCVs.

References

Allan J, Callan J, Croft B, Ballesteros L, Broglio J, Xu J and Shu H (1997) INQUERY at TREC 5. In: Voorhees EM and Harman DK, Eds., Information Technology: The Fifth Text Retrieval Conference (TREC-5). National Institute of Standards and Technology, Gaithersburg, MD, pp. 119–132.

Belkin N, Kantor P, Fox EA and Shaw JA (1995) Combining evidence of multiple query representations for information retrieval. Information Processing and Management, 31(3):431–448.

Buckley C, Mitra M, Walz J and Cardie C (1998) Using clustering and superconcepts within SMART: TREC 6. Online, available from: <URL: http://trec.nist.gov /pubs/trec6/papers/cornell.ps>, cited 7.7.1998 (to appear in Proceedings of TREC-6).

Conover WJ (1980) Practical Nonparametric Statistics, 2nd ed. John Wiley & Sons, New York.

Efthimiadis EN (1996) Query expansion. In: Williams ME, Ed., Annual Review of Information Science and Technology, Vol. 31. Information Today, Medford, NJ, pp. 121–187.

Fidel R and Efthimiadis EN (1995) Terminological knowledge structure for intermediary expert systems. Information Processing & Management, 31(1):15–27.

Green R (1995) The expression of conceptual syntagmatic relationships: A comparative survey. Journal of Documentation, 51(4):315–338.

Harman DK (1995) Overview of the Fourth Text Retrieval Conference (TREC-4). Online, available from: <URL: http://trec.nist.gov/pubs/trec4 /papers/overview.ps>, cited 5.2.1998.

Hawking D, Thistlewaite P and Bailey P (1997) ANU/ACSys TREC-5 experiments. In: Voorhees EM and Harman DK, Eds., Information technology: The Fifth Text Retrieval Conference (TREC-5). National Institute of Standards and Technology, Gaithersburg, MD, pp. 359–375.

Hawking D, Thistlewaite P and Craswell P (1997) ANU/ACSys TREC-6 Experiments. Online, available from: <URL: http://trec.nist.gov/pubs/trec6/papers/anu.ps>, cited 26.2.1998 (to appear in Proceedings of TREC-6).

Hull D (1993) Using statistical testing in the evaluation of retrieval experiments. In: Korfhage R, Rasmussen EM and Willett P, Eds., Proceedings of the 16th International Conference on Research and Development in Information Retrieval. New York, NY, ACM, pp. 349–338.

Hull DA (1997) Using structured queries for disambiguation in cross-language information retrieval. In: AAAI Spring Symposium on Cross-Language Text and Speech Retrieval Electronic Working Notes. Online, available from: <URL: http://www.clis.umd.edu/dlrg/filter/sss/papers/hull3.ps>, cited 13.8.1997.

Ingwersen P and Willett P (1995) An introduction to algorithmic and cognitive approaches for information retrieval. Libri, 45:160–177.

Järvelin K, Kristensen J, Niemi T, Sormunen E, and Keskustalo H (1996) A deductive data model for query expansion. In: Frei H-P, Harman D, Schäuble P and Wilkinson R, Eds., Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, pp. 235–249.

Keen EM (1991) The use of term position devices in ranked output experiments. Journal of Documentation, 47(1):1–22.

Kekäläinen J and Järvelin K (1998) The impact of query structure and query expansion on retrieval performance. In: Croft WB, Moffat A, van Rijsbergen CJ, Wilkinson R and Zobel J, Eds., Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp. 130–137.

Kekäläinen J (1999) The Effects of Query Complexity, Expansion and Structure on Retrieval Performance in Probabilistic Text Retrieval. Ph.D. Thesis, University of Tampere. Acta Universitatis Tamperensis, Vol. 678.

Pirkola A (1998) The Effects of Query Structure and Dictionary setups in Dictionary-Based Cross-Language Information Retrieval. In: Croft WB, Moffat A, van Rijsbergen CJ, Wilkinson R and Zobel J, Eds., Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp. 55–63.

Rajashekar TB and Croft WB (1995) Combining automatic and manual index representations in probabilistic retrieval. Journal of the American Society for Information Science, 46(4):272–283.

Shaw JA and Fox EA (1995) Combination of multiple searches. In: Harman DK, Ed., The Third Text REtrieval Conference (TREC-3). National Institute of Standards and Technology, Gaithersburg, MD, pp. 105–108.

Sormunen E (1994) Vapaatekstihaun tehokkuus ja siihen vaikuttavat tekijät sanomalehtiaineistoa sisältävässä tekstikannassa [Free-text searching efficiency and factors affecting it in a newspaper article database]. VTT Julkaisuja 790. Espoo: Valtion Teknillinen Tutkimuskeskus. [In Finnish.]

Turtle HR (1990) Inference Networks for Document Retrieval. Ph.D. Dissertation, Computer and Information Science Department, University of Massachusetts. COINS Technical Report, pp. 90–92.

UMLS (1994) UMLS Knowledge Sources, 5th experimental edn. National Library of Medicine, Bethesda, MD.

Wang Y-C, Vandendorpe J and Evens M (1985) Relational thesauri in information retrieval. Journal of the American Society for Information Science, 36(1):15–27.

Voorhees E (1994) Query expansion using lexical-semantic relations. In: Bruce Croft W and van Rijsbergen CJ, Eds., Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, pp. 61–69.

Xu J and Croft WB (1996) Query expansion using local and global document analysis. In: Frei H-P, Harman D, Schäuble P and Wilkinson R, Eds., Proceedings of the 19th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, pp. 4–11.