

A Retrospective Evaluation Method for Exact-Match and Best-Match Queries Applying an Interactive Query Performance Analyser

Eero Sormunen

Department of Information Studies

FIN 33014 University of Tampere, Finland

Mail eero.sormunen@uta.fi

Abstract. A retrospective method for the performance comparison of queries based on different IR models is introduced. The method is based on the interactive optimisation of queries by a group of test searchers using a query performance analyser. The case experiment focused on comparing the maximum effectiveness of Boolean exact-match queries, and structured and unstructured best-match queries. The experiment verified the problems in maintaining precision of Boolean queries at high recall levels. Interesting similarities were also observed between structured and unstructured best-match queries challenging the results of earlier studies.

Keywords: Evaluation, Experimental Design and Metrics

1 Introduction

Different types of queries are a challenge for experimental evaluation methods. Depending on the IR model upon which the IR system is based, the query might be a Boolean expression, a vector with weights for each term, a natural language sentence or a bag of words [24]. The comparisons of best-match and exact-match Boolean systems are rare. This is not a surprise since experimenters are facing the problem of diversely developed evaluation methods. Traditional test designs are often IR model specific.

Apart from the differences in evaluation environments and practices, finding an appropriate measure for comparisons remains a key problem. Boolean IR systems retrieve unordered document sets matching exactly the query, and performance is typically measured by two overall figures, precision and recall averaged across the test topics. Best-match systems rank documents in order of decreasing probability of relevance, and performance is measured by averaging precision at some standard points of operation, e.g. fixed recall levels $R_{0.1} \dots R_{1.0}$. The comparison of results is difficult between IR models [9].

1.1 Boolean and best-match queries: query-centred experiments

The study by Salton et al. [16] is a classical experiment comparing queries of different IR models. The effectiveness of Boolean queries, vector-based queries, and extended Boolean queries (based on the p-norm model) was compared. All queries were composed from the same set of query terms (a set of single terms). In the extended Boolean queries, connectives (AND, OR) were softened, and in the vector-based queries the effect of operators was completely abolished. All query results were ranked using the tf-idf formula – also within the result sets of Boolean

queries. The same retrieval software was used for all queries; only the p parameter for Boolean operators was tuned. Precision at standard recall levels could be used as a performance measure. The main finding was that extended Boolean and vector-based queries outperformed traditional Boolean queries.

Paris & Tibbo [12] made a similar experiment by formulating candidate Boolean queries for each of the 100 search topics of the CF test collection using nine protocols. One or a few candidate queries were generated from each protocol. Among other things, protocols guided to vary the number of conjunctions in search for “optimal” Boolean formulations. “Optimal” queries were required to achieve full or nearly full recall at maximum level of precision. Best-match queries were derived from the Boolean queries found optimal by excluding AND operators. Phrases and disjunctive structures were retained for best-match queries. Performance was measured by the overall average precision and by the E measure calculated from high recall queries ($R_{ave}=0.98-0.99$). The main finding was that Boolean queries slightly outperformed best-match queries.

The problem in the above experiments is that they neglected some essential differences between IR systems. Queries compared were forced to contain the same set of terms, and an equivalent structure (if possible). As can be seen from the results, comparisons are sensitive to the strategies adopted in query formulation. Salton’s approach is not fair to Boolean queries. The number of conjunctions in Boolean queries derived from the topic descriptions in natural language tends to be high (i.e., query exhaustivity was high). In queries without expansion, precision collapses already at low recall levels because of the exact match requirement [2, 20]. Similar drop is not likely to happen in best match queries. In the latter study [12], Boolean queries contained few conjunctive structures (i.e., query exhaustivity was low) to guarantee high recall. Best-match queries were punished since higher query exhaustivity could have improved their performance [22].

The conclusion from the query-centred experiments is that queries should be designed separately for different matching models to guarantee fair comparison. Empirical support for this view was published in [18] questioning the results of an experiment by Tenopir and Shu [26].

1.2 Boolean and best-match comparisons: user-centred experiments

Boolean and best-match systems have also been compared by using real searchers to formulate Boolean queries and by using automatic procedures to generate best-match queries [11], or by using real searchers for both Boolean and best-match systems [7]. Ranking algorithms were not applied in these experiments for Boolean queries. Precision and recall were calculated for each query at the document cut-off value equalling the size of the Boolean query result, and averaged over all search topics. In a similar study by Turtle [27], the novelty of documents was used as the ranking criterion in Boolean queries. Average precision at the standard recall levels could be used as the performance measure. However, the idea of using document age as a topical relevance criterion has not received general acceptance [11].

Queries can be formulated by real users for all matching methods compared but the effects of user-related variables on system related variables are difficult to control. This is not a major problem if the goal is to measure the overall performance of a given user group in a given retrieval environment, see [7]. However, the control of user related variables becomes an issue when the goal is to study the core

characteristics of IR systems based upon different matching models. Another problem is that the systematic differences in queries designed for models compared are hard to analyse. Users can select any query terms or query structures they like inducing unnecessary variation in resulting queries.

1.3 The goals and motivation of this study

Both query-centred and user-centred experiments described above have shared similar research goals. They were attempts to reveal the overall superiority of one IR model over the other (see examples in [4]). On the other hand, justified views have been presented that overall performance differences between different IR methods are small [11, 15]. The variation of performance differences from one search topic to another has been suggested as a relevant starting point for future comparisons [12].

Professional search experts have also raised interesting questions: When do best-match queries perform best? When Boolean queries work better? What are appropriate query formulation strategies for an IR system based on particular matching model? See [3] and [25]. These questions are timely since modern IR systems support both Boolean and best-match queries as an integrated functionality. The trend towards IR systems where the user may select between a bag of words queries (simple mode) and structured queries (advanced mode) is clear in the World Wide Web but the present implementations vary a lot.

This paper presents a new method for controlled comparisons of queries based on different matching models and the results of a case study applying the method. The basic ideas behind the proposed method are:

1. Queries are formulated and optimised separately for each IR method. This gives a fair basis for comparisons.
2. The queries used in the comparison are derived from an *inclusive query plan* designed for each search topic. The total set of queries that can be derived from the inclusive query plan is called *query tuning space*. Since queries for all IR models are derived from joint query plans, uncontrolled variation in resulting queries can be reduced.
3. Optimisation requires that relevance data is available and used in the search formulation process. This means that the proposed method is *retrospective*, see [14]. Harter [6] was the first to propose a method for searching optimal query formulations on basis of relevance data.
4. An interactive Query Performance Analyser (QPA) is used as an interactive tool by which the users are able to efficiently and conveniently search for optimal formulations in query types investigated. QPA gives instant visual feedback of the effectiveness of queries formulated [21].

The paper is organised in the following way. Section 2 will explain the basic concepts of the method. Section 3 describes the case study conducted to demonstrate the proposed method. The findings of the case study and the experiences got from the proposed method are discussed in Section 4.

2 A new method for query comparisons

Traditional query-centred experiments employed single queries having an equivalent structure for all systems compared. To overcome the restrictions of fixed queries we introduce a facet-based approach to represent query plans, all queries in the query tuning space and optimal queries.

2.1 A facet-based framework for query structures

The notion of *facet* has been adopted in the Boolean IR literature to represent the relationship between query structures and search topics as expressed information needs. A facet is a concept (or a family of concepts) identified from, and defining one exclusive aspect of a search topic. The notion of facet helps to identify query terms that play a similar semantic role, and are interchangeable in a query or a text. Terms within a facet are naturally combined by Boolean disjunctions. Facets themselves present the exclusive aspects of desired documents. Thus, a natural interpretation for facets is Boolean conjunction or negation [5, p. 76-81].

We need two additional concepts to characterize the structure of Boolean queries. *Query exhaustivity* (*Exh*) is simply the number of facets that are exploited in a query. *Query extent* (*QE*) measures the broadness of a query, e.g. the average number of query terms used per facet. The structural properties, exhaustivity and extent, are illustrated in Figure 1. In query q_i ,

$$Exh(i) = n, \text{ and}$$

$$QE(i) = \left(\sum_{j=a}^n |facet\text{-}terms(F_j)| \right) / n,$$

where $|facet\text{-}terms(F_j)|$ gives the number of terms selected for facet F_j . In query q_i , these numbers for facets $[A]$, $[B]$, and $[N]$ are, respectively, k , l , and m .

The changes made in query exhaustivity and in query extent to achieve appropriate retrieval goals are called query tuning. The range within which query exhaustivity and query extent can change sets the boundaries for query tuning. In query q_i , exhaustivity may be tuned from 1 to n , and extent from 1 to $(k+l+...+m)/n$. Figure 1 illustrates actually a model of an inclusive query plan. The inclusive query plan is always one interpretation of a search topic, and there is no way to guarantee that it is a complete one (if there is such). However, it is based on a controlled work of a search expert, and gives a common platform to compare the behaviour of different query types, see [18].

Query tuning space consists of all queries that can be derived from the inclusive query plan. The size of the query tuning space can be computed using the formula

$$N_{QTS} = (2^k - 1) + (2^k - 1) \cdot (2^l - 1) + \dots + (2^k - 1) \cdot (2^l - 1) \cdot \dots \cdot (2^m - 1)$$

where k , l , and m are the number of query terms for different facets of the inclusive query plan (see Figure 1). For instance, if we assume a query plan consists of 5 facets and 5 query terms per each facet, the number of optional queries is 31 for the first facet, i.e. at $Exh=1$. Further, $31 \times 31 = 961$ optional queries are available for $Exh=2$, $31 \times 31 \times 31 = 29,791$ for $Exh=3$, $31 \times 31 \times 31 \times 31 = 923,521$ for $Exh=4$, and

28,629,151 for $Exh=5$. In the given example, the size of the query tuning space is 29,583,455. The risk of combinatorial explosion is obvious as soon as the exhaustivity and extent of inclusive query plans grow.

2.2 Query optimisation with QPA

Sormunen [18, 19] introduced a heuristic algorithm for optimising Boolean queries at standard recall levels. In the same study, it was shown that the optimisation of queries could be made interactively by real searchers exploiting the Query Performance Analyser. Of course, the interactive optimisation is realistic only in small and medium sized query tuning spaces. The advantage of interactive optimisation is that the constraints of optimisation can be conveniently changed from one experiment to another by giving the searchers new guidelines. The real power of the searcher-based optimisation is that heuristics requiring common sense and searching expertise can be applied.

The Query Performance Analyser is a web-based tool developed at the University of Tampere for the performance analysis and visualisation of individual queries. On top of a laboratory test collection, the tool offers immediate performance feedback in the form of recall-precision curves, and a visualisation of actual query results. The searcher is able to study, in a convenient and effortless way, the effects of any query modifications. The performance data for all queries are stored automatically, and the precision of optimal queries at a particular recall level can be checked easily [21].

The use of QPA¹ is quite straightforward and intuitive. After selecting the search topic, the database, and the IR system to be used, the user enters the query formulation page. (S)he types in the query using the query language of the target IR system (here the Inquiry search language). After the query has been processed in the IR system and query results downloaded, the recall-precision figures are computed, and the resulting P/R graph is automatically presented to the user.

QPA displays the resulting P/R graph highlighted together with the best results achieved in earlier queries. Thus, the user sees immediately after executing a query whether or not any progress has been made. If the average precision over all recall levels exceeds that of earlier queries, the query is automatically assigned to the "Hall of Fame". Actually, any precision curve can be presented in the background as a reference. Basic data of all queries are stored in to a log file.

2.3 Performance measures

Performance of queries based upon different IR models can be measured and represented in several comparable ways but only two of these support the idea of separately optimised queries. Salton et al. [16] assumed that all query results are ranked by using the tf-idf formula. This is a simple and convenient method when the IR system supports the ranking also in Boolean queries, e.g. [1, 16]. In WWW, ranking of Boolean query results is common, see e.g. AltaVista (URL: www.altavista.com). Ranked Boolean queries are not genuine Boolean queries but they are realistic since they are based on the exact match principle.

¹ See the demo of QPA: <http://www.info.uta.fi/~lomise/pinball3.5/pinball3.5.html>.

Another option is to optimise Boolean queries at standard points of operation, e.g. at fixed recall levels $R_{0.1}$ - $R_{1.0}$ [18]. In this case, a P/R curve is interpolated from several Boolean queries found optimal at a particular recall level. The combined P/R curve is compared to the P/R curve of a single optimised best-match query. This approach is more complex than the approach used by Salton and the validity of comparisons based on a group of Boolean queries and a single best-match query may be questioned.

3 A case study

The goal of the case study was to compare the effectiveness and characteristics of Boolean queries, structured probabilistic queries and unstructured probabilistic queries all optimised separately on basis of full relevance data.

3.1 Research questions

The research questions of the study were:

1. *General performance characteristics.* Are there differences in the average performance capability between Boolean queries, structured best-match queries and unstructured best-match queries? The tentative hypothesis was that the effectiveness of structured best-match queries should be highest since it combines weighting from the best-match IR models with the query structures of the Boolean IR model but avoids the pitfalls of exact matching.
2. *Query exhaustivity and extent.* How exhaustivity and extent vary in queries optimised separately for different query categories? An earlier experiment showed that high recall could be achieved in Boolean queries only by reducing dramatically the exhaustivity of queries [19, 20]. It has also been suggested that, in best-match queries, higher exhaustivity could be used to maintain precision in high recall searching [22]. Unstructured queries have usually suffered from query extension [10]. Thus, the extent of optimal unstructured queries should be lower than in other queries.
3. *Performance characteristics at different exhaustivity levels.* How exhaustivity is related to performance in different query categories? Different search topics contain a varying number of searchable facets, and real users identify and apply all or some of them in the query formulation process. An interesting question is, which of the query categories most likely leads to the highest effectiveness at a given exhaustivity level.

3.2 Methods and data

Test Collection

The test environment is a text database containing Finnish newspaper articles operated under the InQuery retrieval system (version 3.1). The database contains 53,893 articles published in three Finnish newspapers. For the database there is a collection of 35 topics, which are 1-2 sentences long, in the form of written information need statements. For the topics of the collection there is a recall base of 17,337 articles, which fall into four relevance categories. The base was collected by pooling the result sets of thousands of different queries formulated from the topics in different studies, using both exact-match, and best-match retrieval [10, 18].

In addition, the test collection provides inclusive query plans that were designed by an experienced search analyst. One of the goals in inclusive query planning was to identify all searchable facets for each search topic. The mutual recall capability of facets was estimated to find one fixed facet order to be used in experiments [18].

A subset of 18 search topics was used in this experiment. For this subset, the results of a comprehensive text analysis of all relevant documents are available, i.e. how query plan facets have been expressed [18]. Thus, the subset of the test collection provides several extraordinary features: reliable relevance data, inclusive query plans including an ordered set of facets, and occurrence data how query plan facets have been expressed in relevant documents.

Inquery

The easiest way to design a test setting is to optimise queries in a retrieval system that supports three query categories:

1. Boolean queries: Exact-match Boolean queries creating a distinct result set are supported.
2. Structured best-match queries: Facet-based query structures are supported in ranking documents but exact match is not required.
3. Unstructured best-match queries: Facet-based query structures may be dismissed.

All this was available in InQuery.

InQuery is a best-match retrieval system but it also allows retrieval of strict Boolean result sets. All result sets, whether agreeing Boolean conditions or best match queries, are ranked. InQuery is based on Bayesian inference networks and it supports a wide range of operators, including strict Boolean AND, OR, NOT and proximity operators as well as various best match operators. For details, see [1, 8, 27].

Query plans

Inclusive query plans designed in the earlier research project for the test collection [18] were used as a starting point of query plans in the present study. The ordered set of facets was taken as a frame for query plans but query terms used in the earlier experiment were rejected. Expressions identified in the facet analysis of relevant documents were used instead. Collecting query terms this way guaranteed that all terms occur at least some training set documents, and on an equal basis for all search topics. In addition, the idea was to create a reference for a study of real users trying to capture best query terms without any external help (an idea for a future experiment).

A critical issue in retrospective evaluation is the risk of over-fitting [14]. The problem is that the optimum may be found on basis of unpredictable document features like spelling errors or rarely used expressions. A secure way to avoid over-fitting is to use different documents for optimising queries (*a training set*) and testing their performance (*a test set*).

The 661 relevant documents for the 18 search topics were divided into two groups by taking a systematic sample. The sample used as the training set consisted of 335

articles. The rest of documents (326 articles) were used as the test set for performance measuring.

Query terms for each query plan facet were selected through the following process:

1. a list of all expressions used to represent a facet in the training set documents was composed,
2. all complex phrases not having an established status like "chemical, biological and nuclear weapons" or "arms factory and armoury" were excluded since they were not regarded as likely query term candidates,
3. all expressions occurring only in one relevant document were excluded.

The aim of pruning the original list was to make query plans more manageable for test searchers. Expressions appearing rarely in texts are neither likely to appear as query terms in real searching situations.

The original query plans contained from 2 to 5 facets (average 3.9). After the pruning process, the total number of query terms accepted was 452, which corresponds to 25 query terms per query. The average number of query terms was 6.4 per facet ranging from 1 to 20. Facets related to named persons and organizations provided quite few query terms. Since the names of persons and organizations are usually quite good query terms both in terms of recall and precision, their facets are typically ranked first in query plans. Thus, the average number of query terms was as low as 4.8 for the first facets (i.e. Exh=1) while ranging between 5.9 and 7.6 terms in other facets.

The three versions of query plans were generated and stored as a text file to make the work of test searchers as convenient as possible.² Operators *#band* and *#and* were used to connect facets in Boolean and structured queries, respectively. Operator *#syn* was used to combine query terms within facets. In unstructured queries, all query terms were combined by the default operator *#sum*, except for phrases. Proximity operator *#5* was used for phrases in all query types. The use of proximity operator is not common in experiments using unstructured queries, but can be justified when queries are formulated manually.

Optimisation

Three test searchers, all competent users of the InQuery system and the Query Performance Analyser, were selected as query optimisers. They were given written guidelines, and the procedure of optimisation was also explained in an introductory session. After this all test searchers made some optimisations to train themselves, and a new meeting was held to clarify the details of the procedure.

The optimisation was conducted in two stages. First, each test searcher got a set of six search topics, and an overall time limit of 6 hours per search topic in optimisation. After all searchers had completed their work, each searcher was given three search topics optimised by two other searchers. The idea of the second round

² *#band* is a strict Boolean AND-operator; *#and* is a 'soft' Boolean operator giving a product of the weights of all keys or InQuery expressions within its scope. All operands within the *#syn* are treated as instances of one search key. *#sum* gives an average of the weights of the operands. *#n* is a proximity operator specifying its operands within n words in given order.

was to check syntactic and technical errors in optimisation results as well as find more optimal queries. A time limit for performing the second round was 2 hours per search topic.

The test searchers were advised to seek optimal queries separately at each exhaustivity level, and test at least 10 query versions for each exhaustivity level and query type. Boolean queries were optimised first, next structured queries, and finally unstructured queries. The order of query types was not rotated but the searchers were encouraged to return to optimise earlier query types if any doubts raised in course of the work.

A separate copy of QPA was used for the optimisation of different query types, and the searcher could make direct comparison only within a query category but not between them. The measure of effectiveness used in comparing queries was precision averaged across recall levels $R_{0.1}$ - $R_{1.0}$. All queries with time stamps, user ids, and measured precision averages were automatically stored into a log file. Best queries overall were available on the "Hall of Fame" but this file typically contained only optimal queries for one exhaustivity level. Best queries for other exhaustivity levels had to be checked from the log file.

The use of two stage optimisation turned out to be useful since two major syntactic errors affecting substantially the optimisation results were observed, and could be corrected. The other aim of redundant work was to reveal "blind spots" in optimisation procedures adopted by individual searchers. The second searcher could improve 16 (23%) of the Boolean, 23 (32%) of the structured, and 22 (31%) of the unstructured queries. In one search topic, the effectiveness of queries improved substantially but most improvements did not have practical importance.

The total number of queries attempted per search topic was about 520 for Boolean, 280 for structured and 350 for unstructured queries. The number of attempts per a search topic ranged from 77 to 3050. These figures correlated with the number of potential combinations originating from a particular query plan ranging. For instance, the query plan for the former topic (77 attempts) contained 2 facets and 13 query terms while the latter (3050 attempts) contained 5 facets and 52 query terms.

Test runs and data analysis

The three series of queries achieving the highest average precision over all recall levels at each exhaustivity level were collected from the log files. The relevant documents of the training set were not removed from test database but they were excluded by using the operator *#bandnot* of InQuery. Actual test queries were of form *#bandnot(Q_{opt} #syn(n₁, n₂ ...n_m))*, where *Q_{opt}* is the optimised query, and *n₁, n₂ ...n_m* are id-numbers for relevant documents belonging to the training set.

Query exhaustivity and extent data was gathered from the lists of optimal queries. Standard tools available for InQuery were used to collect and analyze performance data. We compared the performance as average precisions at standard recall levels and grand precision averages over recall levels. Statistical significance was tested with Friedman two-way analysis by ranks using both types of precision averages.

3.3 Results

Performance and structure of optimal queries

Structured best-match queries performed somewhat better than the other queries (Figure 2). The average precision in structured queries was 0.07 above Boolean and 0.06 above unstructured best-match queries but the differences observed were not statistically significant. At the lowest recall levels, the precision of Boolean queries achieved that of structured queries while, at the highest recall levels, Boolean queries were not as effective as the best-match queries. It turned out that the observed lower precision of Boolean queries at highest recall levels was statistically significant (see Table 1).

The results gave partial support to our tentative hypothesis that structured queries should perform better than other query categories since they combine weighting and query structures but avoid pitfalls of exact matching (distinct result sets). The success of Boolean queries at the lowest recall level may sound surprising but this phenomenon has a test environment based explanation. Even in the strict Boolean mode, InQuery ranks the documents within the result set. Boolean queries enjoyed similar weighting benefit as structured queries. At the highest recall levels, the precision of Boolean queries fell below that of other queries because the exact match requirement rejects completely some of the relevant documents.

The stalemate between structured and unstructured queries at the highest recall levels was not in line with the tentative hypothesis. A potential explanation for equal performance may relate to the characteristics of documents that are retrieved only at the highest recall levels, the least retrievable documents [18]. Typical of the least retrievable documents is that they either do not contain searchable expressions for one or more query facets, or the expressions used in the text do not match terms used in the query. In addition, the number of expression occurrences is lower in least retrievable documents. The least retrievable documents do not provide much evidence for weighting based on term occurrences or on co-occurrence of facets. Two other potential explanations for the stalemate between structured and unstructured queries are discussed in Section 4.

Structural characteristics of optimised queries

The average exhaustivity and extent of optimised queries is presented in Figure 3. It turned out that the exhaustivity of Boolean queries was only about 2.8 while rose to 3.6 in structured queries and up to 3.7 in unstructured queries. The measured exhaustivity difference was statistically significant between the Boolean and best-match queries but not between structured and unstructured queries. The average extent of queries was highest in unstructured queries (3.5), lowest in the structured queries (3.0), and quite high in Boolean queries (3.3). Extent differences were not statistically significant in this data set. Thus the discussion to follow is speculative in nature because it is based on illustrative examples without firm statistical evidence.

The low exhaustivity of the Boolean queries was not a surprise since the requirement of exact match limits the use of facets. If full recall is required, the exhaustivity of queries may drop below 2 (see [20]) implicating that many Boolean queries optimal for high recall searching employed only one facet. In this study, Boolean queries were not required to retrieve all relevant documents, and the

optimum was found at a higher level of exhaustivity leading to higher average precision across fixed recall levels than single facet queries. In structured and unstructured queries, the average exhaustivities were very close to the maximum (3.9).

In the Boolean queries, 20 out of the 71 facets were not employed in optimal queries. In 11 search topics, at least one facet was neglected but in seven search topics all facets were exploited. In 6 search topics, the exhaustivity of Boolean queries was equal to or more than 4. This is just to emphasize the contradiction of average results and individual queries. Sometimes expressions for several facets of a query co-occur in most relevant documents but it is also common that only one or two facets could be employed.

In the structured queries, the number of neglected facets was only 6/71 in 4 search topics, and in the unstructured queries the number was 4/71 facets in 4 search topics, respectively. The results suggest that in structured or unstructured best-match searching all searchable facets should be employed. All six facets that were rejected in the optimisation process were quite general and difficult to express by query terms of any discriminating power.

The extent of optimal queries in unstructured queries deserves further analysis. Earlier studies, e.g. [10], have shown that query expansion does not improve the effectiveness of unstructured queries but just the opposite. Our results suggest that there is neither difference in the extent nor in the effectiveness of structured and unstructured queries. This contradiction is discussed in Section 4 as well.

Optimal queries on different exhaustivity levels

The aim of comparing optimal queries at different exhaustivity levels is to comprehend better the behaviour of different query types when the number of available or employed facets varies. Figure 4 presents a comparison of queries optimised at low exhaustivity levels 1 and 2.

If only one facet is employed, no clear performance differences between the query types were observed. The difference in averages was less than 0.02 and overall differences or differences at individual recall levels were not statistically significant (Table 1). The average precision of queries was well below the optimum presented in Figure 2 except at the highest recall levels $R_{0,9} - R_{1,0}$. If the goal of searching is to retrieve all relevant documents, single facet queries may be as competitive as any more focused best-match queries of higher exhaustivity (precision varies from 0.054 to 0.056). The top of the ranked list may be richer of relevant documents in multi-facet queries but the last relevant document has always a very low rank. Overall, precision was quite low for all query types and for all exhaustivity levels at the highest recall levels.

In case of two facets, the average precision of Boolean and structured queries was slightly above (0.05 - 0.06) the precision of unstructured queries. The role of query structures is advantageous both in strict and soft sense. At the highest recall level $R_{0,9} - R_{1,0}$, the effect of strict AND-operators drops the precision of Boolean queries and they are no more competitive with best-match queries. The above results turned out to be statistically significant (see Table 1). The precision curve for the Boolean queries reached its maximum position at exhaustivity level 2. In other words, one could see Exh=2 as the default value for exhaustivity in optimal Boolean searching.

Of course, other factors may lead to increase or decrease exhaustivity in individual cases but, anyway, the over-exhaustivity of queries is a major performance risk in Boolean searching, see [18].

Figure 5 presents the P/R graphs for the exhaustivity levels 3 and 4. When three or four facets were employed, the precision of Boolean queries fell clearly below the precision of other query types. This was especially clear at $R_{0.5}$ and above. At the lowest recall level, Boolean queries were still competitive. The average precision was again higher in structured queries than in unstructured queries (difference 0.03-0.05), but at the lowest and at the highest recall levels the difference was negligible. However, the difference observed between best-match queries was not statistically significant (see Table 1).

4 Discussion and conclusions

4.1 Findings of the case experiment

The results of the case experiment corroborated the findings in [18, 19, 20] suggesting that the requirement of exact-match in traditional Boolean queries leads to the fall of precision in high recall searching. In this study, we could compare Boolean queries and best-match queries, and verify that the decline of effectiveness is associated with the exhaustivity of queries. Over-exhaustivity is an effectiveness risk in the formulation of Boolean queries. The fall of precision was steadier in best-match queries.

Although Boolean queries were less effective than the others, all query types suffered from low effectiveness at the highest recall levels $R_{0.9}$ - $R_{1.0}$. Even under the idealized conditions (a homogeneous and relatively small collection of documents, query terms from relevant documents, optimisation with full relevance data) precision fell close to or below 10 percent. This result gives a chance for a pessimistic prognosis in searching of large databases like web indexes.

Earlier experiments based on predictive evaluation methods have shown that structured queries benefit of query expansion but unstructured queries suffer from the increase of query extent [10]. Similar difference between structured and unstructured queries has been observed in CLIR experiments [13, 23]. Our results suggest that there is difference neither in precision nor in extent of optimal queries. Two potential explanations were easy to name:

1. *Phrases and proximity operators.* Even in unstructured queries phrases were expressed by using #5-operator. In the studies mentioned, phrases have been split into component words and treated as independent query terms. An additional test was conducted the effect of phrases and proximity operators. The average precision of unstructured queries fell only insignificantly. Thus proximity operators were not explaining the contradiction in results.
2. *Only highly focused query terms used.* All terms used were justified expressions for the query plan facets, and were representative for the relevant documents of the training set. In addition, all query terms that did not improve the effectiveness of queries were very likely to be rejected in the optimisation process. This is not the case in predictive evaluation where queries tend to contain also very poor or even harmful query terms. The role of the query structure is to minimize the effect of noise generated by these terms [8]. The sensitivity of different query types to the effects of noise query terms was tested

by adding up to 5 broad terms to each facet of optimal queries. It turned out that precision was falling fastest in Boolean and structured queries. So, the second explanation failed.

After these failures to falsify the original findings one could conclude that different evaluation methods may emphasise different aspects of the same phenomenon. The results suggest that if an optimal set of query terms covering all relevant facets (high exhaustivity) and alternative expressions (high extent) is found, the structure of queries brought by operators do not have any role in improving performance. The situation may be different in real life where the set of optimal query terms is extremely difficult to discover (as predictive evaluations suggest).

4.2 Retrospective evaluation and the use of QPA

The image of the retrospective evaluation method as applied by Shaw [17] and criticized by Robertson [14] has been very poor. The work by Sormunen [18, 19] partially based on the ideas of Harter [6] justified that the retrospective approach can be used reasonably in analyzing the performance capability and structures of Boolean queries. This study expanded the use of the retrospective approach to the comparison of Boolean and best-match queries. Another difference was that now query optimisation was based totally on the interactive use of the Query Performance Analyser.

Over-fitting in optimising queries with the help of full relevance data is a justified fear but this fear should not be overemphasized. The use of separate training and test sets is a simple solution to solve the problem although it increases the amount of work. Even a more important question is how experiments are designed and the number of uncontrollable variables is reduced. The role of query plans as a solid framework for the query tuning space, and the control of the optimisation process are key issues in this respect. The more there are degrees of freedom in the query optimisation process, the more difficult it is to make valid inference on empirical results.

The use of the Query Performance Analyser was not comprehensively evaluated as an optimisation tool but first impressions were encouraging. The number of query modifications compared by the searchers was substantial. In predictive evaluation and in traditional experimental designs, an equal versatility of queries is difficult to achieve. The advantage of QPA is that it supports also the detailed analysis of query results, see [18]. An obvious danger in the optimisation is that test searchers easily get excited of "game playing". In hunt for higher scores, test persons may forget the actual goals of their work. For instance, we noticed that those exhaustivity levels at which the highest performance was achieved, more candidate queries were composed than at other exhaustivity levels. If the risk is not realized in user guidance and control, data may suffer from biases.

5 Acknowledgements

The author is grateful to Jaana Kekäläinen, Jussi Koivisto, Erkka Leppänen and Katja Nirkkonen for their contribution in the experiment. The members of the FIRE group have also helped to improve the manuscript.

InQuery (TM) SOFTWARE Modifications Copyright (c) 1998-2000 by the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts at Amherst. All rights reserved. InQuery (TM) Copyright (c) 1996-2000 by Dataware

Technologies, Inc., Hadley, Massachusetts, U.S.A. (413-587-2222; <http://www.dataware.com>). All rights reserved. The InQuery (TM) software was developed in part at the Center for Intelligent Information Retrieval (CIIR) at the University of Massachusetts at Amherst (For more information, contact 413-545-0463 or <http://ciir.cs.umass.edu>). InQuery (TM) is registered trademark of Dataware Technologies, Inc.

6 References

- [1] Allan, J., Callan, J., Croft, W.B., Ballestros, L., Broglio, J., Xu, J. & Shu, H. (1997). INQUERY at TREC 5. In: Harman, D.K. & Voorhees, E.M. (Eds.) *Information technology: The Fifth Text REtrieval Conference (TREC-5)*. Gaithersburg, National Institute of Standards and Technology, 119–132.
- [2] Blair, D.C. & Maron, M.E. (1985). An evaluation of retrieval effectiveness for a full-text document retrieval system. *Communications of the ACM* (28)3, 289-299.
- [3] Feldman, S. (1996). Testing Natural Language: Comparing DIALOG, TARGET, and DR-LINK. *Online* 20(6), 71-79.
- [4] Frants, V.I., Shapiro, J., Taksa, I. & Voiskunskii, V.G. (1999). Boolean Search: Current State and Perspectives. *Journal of the American Society of Information Science* 50(1), 86-95.
- [5] Harter, S.P. (1986). *Online Information retrieval*. Orlando: Academic Press.
- [6] Harter, S.P. (1990). Search Term Combinations and Retrieval Overlap: A Proposed Methodology and Case Study. *Journal of the American Society for Information Science* 41(2), 132-146.
- [7] Hersh, W.R. & Hickam, D.H. (1995). An Evaluation of Interactive Boolean and Natural Language Searching with Online Medical Textbook. *Journal of the American Society for Information Science* 48(7), 478-489.
- [8] Hull, D. (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]. [Cited 13.8.1997.] Available from: <URL: <http://www.clis.umd.edu/dlrg/filter/sss/papers/hull3.ps>>
- [9] Keen, E.M. (1992). Presenting results of experimental retrieval comparisons. *Information Processing & Management* 28(4), 491-502.
- [10] Kekäläinen, J. (1999). *The Effects of Query Complexity, Expansion and Structure on Retrieval Performance in Probabilistic Text Retrieval*. Doctoral Thesis. Tampere: University of Tampere. (Acta Universitatis Tamperensis 678).
- [11] Lu, X.A., Holt, J.D. & Miller, D.J. (1996). Boolean System Revisited: Its Performance and its Behaviour. In: Harman, D.K. (Ed.) *The Fourth Text REtrieval Conference (TREC-4)*. Gaithersburg, National Institute of Standards and Technology, 459–473.

- [12] Paris, L.A.H. & Tibbo, H.R. (1998). Freestyle vs. Boolean: A comparison of partial and exact match retrieval systems. *Information Processing & Management* 34(2/3), 175-190.
- [13] Pirkola, A. (1999). *Studies on linguistic problems and methods in text retrieval: the effects of anaphor and ellipsis resolution in proximity searching, and translation and query structuring methods in cross-language retrieval*. Doctoral Thesis. Tampere: University of Tampere. (Acta Universitatis Tamperensis 672.)
- [14] Robertson, S.E. (1996). Letter to the Editor. *Information Processing & Management* 32(5), 635-636.
- [15] Robertson, S.E. & Thompson, C.L. (1990). Weighted searching: The CIRT experiment. In: Jones, K.P. (Ed.), *Informatics 10 – Prospects for Intelligent retrieval*. London: Aslib, p. 153-165.
- [16] Salton, G., Fox, E.A. & Wu, H. (1983). Extended Boolean Information Retrieval. *Communications of the ACM* 26(11), 1022-1036.
- [17] Shaw, W.M. (1995). Term-relevance computations and perfect retrieval performance. *Information Processing and Management* 31(4), 491-498.
- [18] Sormunen, E. (2000a). *A method for measuring wide range performance of Boolean queries in full-text databases*. Doctoral Thesis. Acta Electronica Universitatis Tamperensis, URL: <http://acta.uta.fi/pdf/951-44-4732-8.pdf>. Tampere: University of Tampere, 2000.
- [19] Sormunen, E. (2000b). A novel method for the evaluation of Boolean query effectiveness across a wide operational range. In: Belkin, N.J., Ingwersen, P. and Leong, M.-K: eds. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, 2000, 25–32.
- [20] Sormunen, E. (2001) Extensions to the STAIRS Study - Empirical Evidence for the Hypothesised Ineffectiveness of Boolean Queries in Large Full-Text Databases. Submitted for publication in *Information Retrieval*.
- [21] Sormunen, E., Keskustalo, H. & Halttunen, K. (2001a). Query Performance Analyser - a interactive tool for bridging information retrieval research and education. Submitted for publication in *Information Retrieval*.
- [22] Sormunen, E., Kekäläinen, J., Koivisto, J. and Järvelin, K. (2001b). Document text characteristics affect the ranking of the most relevant documents by expanded structured queries. Accepted for publication in *Journal of Documentation*, May 2001.
- [23] Sperer, R. & Oard, D.W. (2000). Structured translation for cross-language information retrieval. In: Belkin, N.J., Ingwersen, P. and Leong, M.-K: eds. *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY: ACM, 2000, 120-127.
- [24] Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. *Information Processing and Management* 28(4), 467-490.

- [25] Tenopir, C. & Cahn, P. (1994). TARGET & FREESTYLE: Dialog and Mead join the relevance ranks. *Online* 18(3), 31-47.
- [26] Tenopir, C. & Shu, M.E. (1989). Magazines in full text: uses and search strategies. *Online Review* 13 (2), 107-118.
- [27] Turtle, H. R. (1990). *Inference networks for document retrieval*. Ph.D. dissertation. Computer and information Science Department, University of Massachusetts. COINS Technical Report 90-92.

Figure 1. The structural dimensions of a query for a search topic containing n identifiable facets.

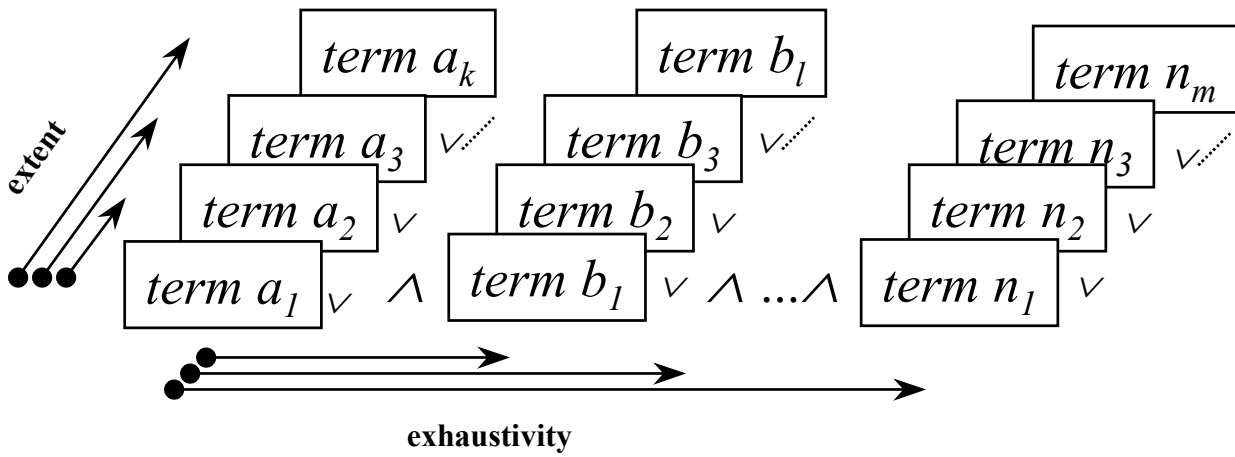


Figure 2. Average performance of optimised Boolean, structured and unstructured queries in the test set (18 search topics).

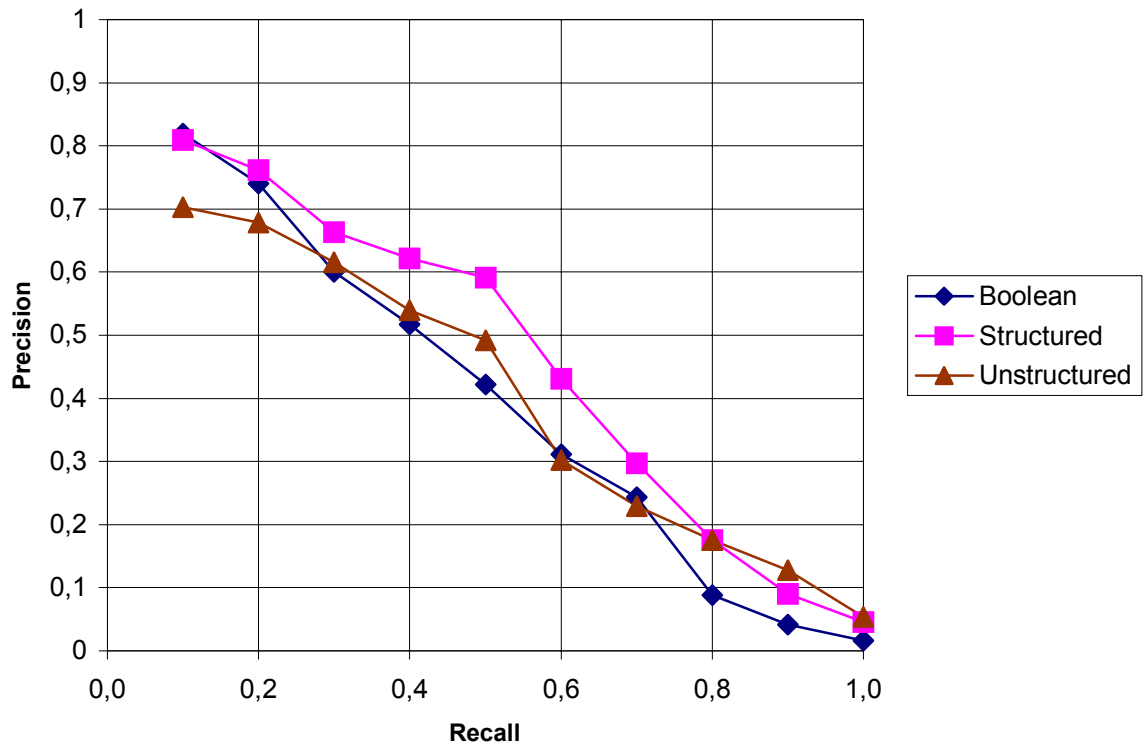


Figure 3. Average exhaustivity and extent of optimised Boolean, structured and unstructured queries (18 search topics).

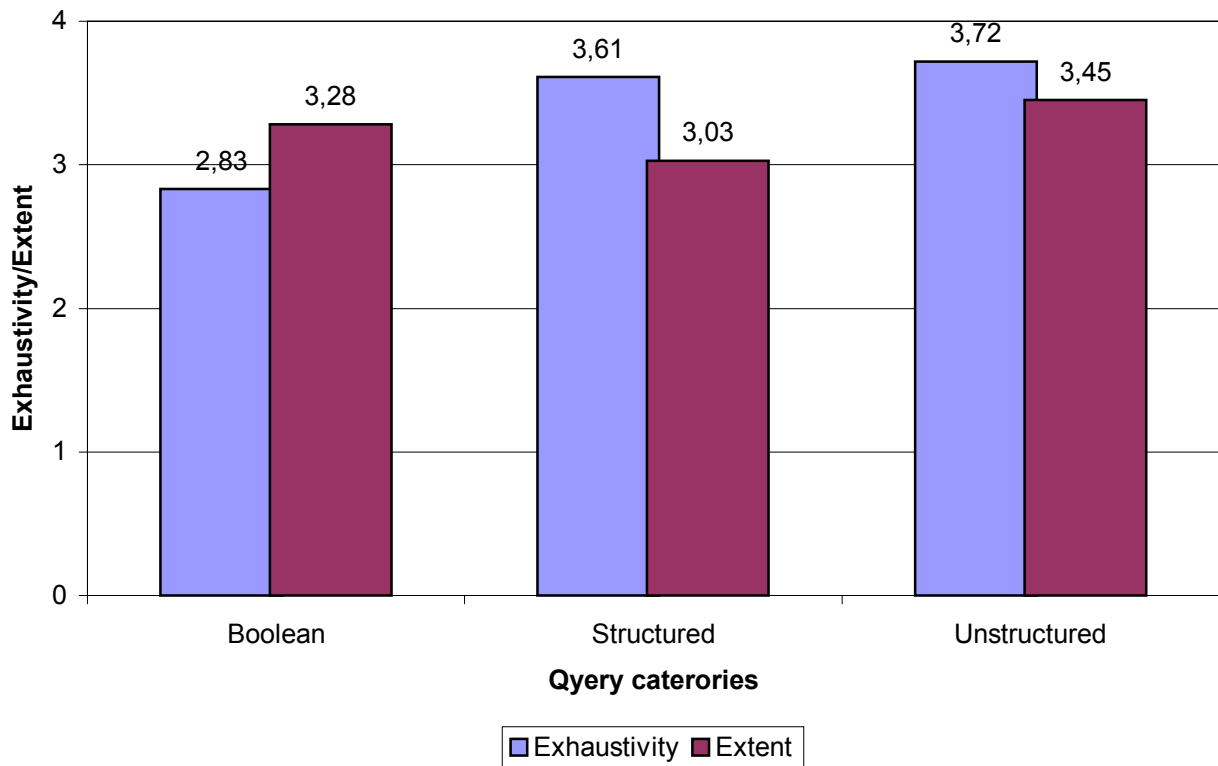


Figure 4. Performance of optimised Boolean, structured and unstructured queries at low exhaustivity levels Exh=1-2 (18 search topics).

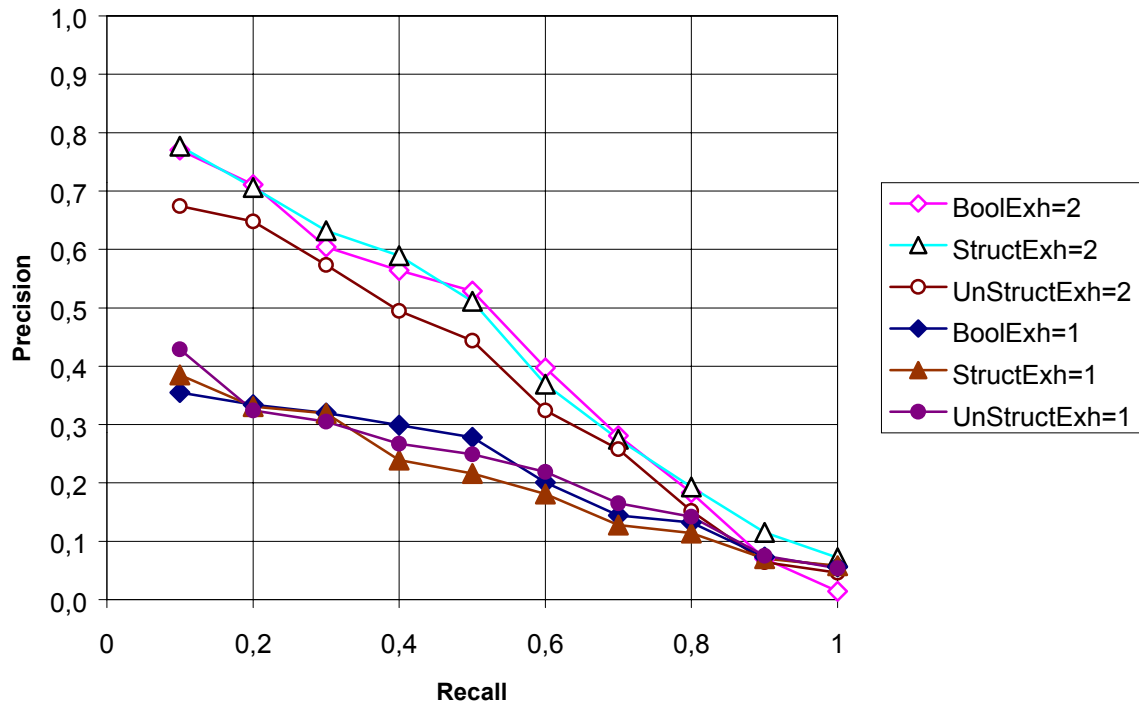


Figure 5. Performance of optimised Boolean, structured and unstructured queries at high exhaustivity levels Exh=3-4 (12-17 search topics).

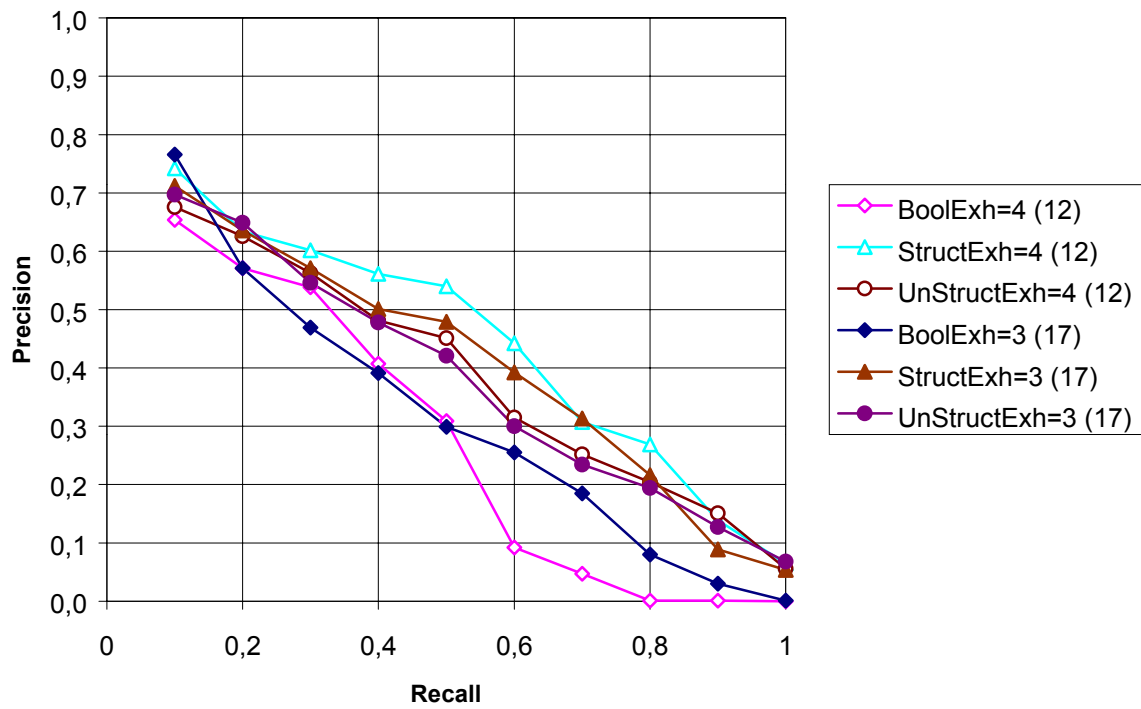


Table 1. The results of the statistical significance test (Friedman) for precision differences. B=Boolean, S=structured best-match, U=unstructured best-match queries. Significance levels: * denotes $p < 0.05$, ** $p < 0.01$, and *** $p < 0.001$, respectively.

Recall	Best Queries Overall	Best Queries (Exh=1)	Best Queries (Exh=2)	Best Queries (Exh=3)	Best Queries (Exh=4)
0.1	-	-	B,S>U**	-	-
0.2	-	-	B,S>U*	-	-
0.3	-	-	-	-	-
0.4	-	-	B>U**	-	-
0.5	-	-	-	-	-
0.6	-	-	-	-	S,U >B**
0.7	-	-	-	-	S,U >B***
0.8	S>B*	-	-	S,U>B***	S,U >B***
0.9	S,U>B*	-	-	S,U>B***	S,U >B***
1.0	S,U>B**	-	S,U>B***	S,U>B***	S,U >B***
Average	-	-	B,S>U*	-	S>B*