A Comparative Evaluation of Full-text, Concept-Based, and Context-Sensitive Search

Robert Moskovitch[1], Susana B. Martins[2], M.D., M.Sc., Eytan Behiri[3], M.D., Aviram Weiss[4], M.D, and Yuval Shahar[1], M.D, Ph.D.

[1]Medical Informatics Research Center
Department of Information Systems Engineering
Ben Gurion University, P.O.B. 653, Beer Sheva 84105, Israel
[2]Stanford University, Stanford, VA Palo Alto Health Care Center, Palo Alto, CA, USA
[3]E&C Medical Intelligence Inc., 41 Madison Ave, New York, NY 10010
[4]Medical Corps, Israel Defense Force, Israel

Contact Person:

Address for correspondence:
robertmo@bgu.ac.il
Medical Informatics Research Center,
Department of Information Systems Engineering,
Ben Gurion University of the Negev, Israel.
P.O.B. 653, Beer Sheva 84105, Israel.
Fax – +972-8-6477161
Phone - +972-52-2668071

Abstract

Objectives*: Study comparatively (1) concept-based search, using documents pre-indexed by a conceptual hierarchy; (2) context-sensitive search, using structured, labeled documents; and (3) traditional full-text search. Hypotheses were: (1) more contexts lead to better retrieval accuracy, and (2) adding concept-based search to the other searches would improve upon their baseline performances.*

Design: *Use our Vaidurya architecture, for search and retrieval evaluation, of structured documents classified by a conceptual hierarchy, on a clinical guidelines test collection.*

Measurements*: Precision computed at different levels of recall to assess the contribution of the retrieval methods. Comparisons of precisions done with recall set at 0.5, using t-tests.*

Results: *Performance increased monotonically with the number of query context elements. Using context-sensitive terms, mean improvement was 11.1% at recall 0.5. With three contexts, mean query precision was 42% ± 17% (95% confidence interval (CI), 31% to 53%); with two contexts, 32% ± 13% (95% CI 27% to 38%); and one context, 20% ± 9% (95% CI, [15% to 24%). Adding context-based queries to full-text queries monotonically improved precision beyond the 0.4 level of recall. Mean improvement was 4.5% at recall 0.5. Adding concept-based search to full-text search improved precision to 19.4% at recall 0.5.*

Conclusions: *The study demonstrated usefulness of concept-based and context-sensitive queries for enhancing the precision of retrieval from a digital library of semi-structured clinical guideline documents. Concept-based searches outperformed free-text queries, especially when baseline precision was low. In general, the more ontological elements used in the query, the greater the resulting precision.*

Keywords*: Information Retrieval, Medical Informatics, clinical practice guidelines, concept-based search, context-sensitive search, Digital Libraries.*

I. Introduction

Full-text search using a set of keywords is probably the best known and most widely used search method, whether in general-interest documents or in medically oriented ones. The general idea behind the method involves searching through the document group, looking for the keywords. This is typically achieved by search engines in an efficient manner, by constructing an index of keywords extracted from the documents. The process of indexing documents refers to a construction of an *index* in which the documents are represented efficaciously, thus enabling an efficient future search-and-retrieval operation [1]. The previously created indices are matched syntactically to the search words when queries are performed. The main advantage of a full-text search is the fast automated indexing process, which extracts the words automatically from the documents, thus requiring no human manual intervention. A general disadvantage of the full-text search is the potentially poor retrieval performance. In many cases, most of the full-text search results are irrelevant or incomplete, since only a statistical match is guaranteed. Not only is the quality of the results poor, but also the number of results may be large.

In specific domains in which a meta-thesaurus is available, such as the medical one (UMLS) [2], documents can be annotated according to the domain concepts. This approach was adopted also in more general document repositories, such as the Web, in which a hierarchical conceptual structure is designed by domain experts and documents are classified according to their contents, along several concepts in the hierarchy. Exploiting this hierarchical organization for search and retrieval purposes is termed *concept-based search* [3, 4].

A major disadvantage of full-text search, which leads to poor results and a high level of recall, is the potential existence of one or more of the keywords in a document in contexts other than the one the searcher had in mind. This disadvantage can be ameliorated by query-

ing for keywords within specific parts of the document, by explicitly stating in which context (e.g., the Introduction) they should be found, an idea that has previously been exploited for improving the precision of the retrieval of medical studies [5,6]. Exploiting a contextual representation of documents for search and retrieval purposes is termed *context-sensitive search*.

In our previous studies, we designed and implemented *Vaidurya* [7], a concept-based and context-sensitive search engine specialized for search and retrieval of documents, such as clinical guidelines, that are represented in a hierarchically organized library that also supports document structuring. The original motivation for designing Vaidurya was to provide a support for the process of effective search and retrieval of *clinical guidelines*, in particular for the purpose of application of one or more of the guidelines most relevant to the patient at hand. Vaidurya is intended for use, for example, by residents or general practitioners who are not necessarily familiar with, or are not experts in, the use of all potentially applicable guidelines. (See the discussion of the DeGeL digital-guideline architecture [8] later in this paper.) However, the design and implementation of Vaidurya are quite generic and have the potential of supporting any type of document content, any classification hierarchy, and any internal hierarchical structure.

Underlying an effective use of the Vaidurya search engine are the following two assumptions: (1) the documents are classified externally by multiple semantic indices, which supports a concept-based search; (2) the documents are structured internally by some (known) hierarchical ontology, which supports a context-sensitive search. One (or more) of the assumptions can be false, in which case the Vaidurya engine uses only the available external or internal ontology, or even falls back on standard full-text search

In this study, we focus on the investigation of the two major types of search, in addition to full-text search, which we implemented within Vaidurya: (1) *concept-based search*,

which relies on the hierarchical classification, or pre-indexing, of the documents in a clinically meaningful fashion by one or more of the concepts from the predefined "external" ontology (essentially, a taxonomy of concepts); and (2) *context-sensitive search*, which relies on the documents first being semistructured by a human editor or automated means, using a predefined "internal" ontology, such as the one used to structure clinical guidelines for purposes of automated application at the point of care; the search for specific terms is then performed only within text that is labeled by a semantically meaningful ontological tag specified by the user or suggested automatically during the structuring phase (e.g., “eligibility conditions,” in the case of clinical guidelines).

In the following paper, we start by surveying the relevant literature, after which we introduce the Vaidurya search and retrieval architecture. We then proceed to describe a set of experiments we have performed, discuss their results, and, based on these experiments, assess the contribution of concept-based and context-sensitive search methods in addition to the traditional full-text retrieval approach.

II. Background

The original motivation for designing the Vaidurya search engine was to support search within the Digital Electronic GuidelinE Library (DeGeL) [8], although Vaidurya is a generic concept-based and context-sensitive search engine, in addition to having standard full-text search capabilities.

DeGeL [8] consists of a distributed architecture and a set of Web-based software tools for incremental conversion ("markup") of Clinical Practice Guidelines (CPGs) into a representation in one or more CPG-specification formats (the equivalent of an internal document ontology). The output of the incremental structuring process is what we term a hybrid repre-

sentation of a CPG, which contains one or more of the following three formats: Semi-structured Text – text fragments assigned to the semantic knowledge-roles of a top-level target CPG ontology, such as GEM [9], Asbru [10], or GLIF [11] (e.g., the "*Filter Condition*" knowledge role, a type of compulsory eligibility condition, in the case of Asbru ontology); Semi-formal representation (which includes control information); and Formal representation (which is fully executable). Only the Semi-structured Text representation, which is exploited by the context-sensitive search, is relevant to this study. (See further discussion of the use of knowledge roles in Section II.B., which describes context-sensitive search.)

The combination of a hierarchical indexing taxonomy and an internal ontological structure made the DeGeL architecture into an ideal platform for the implementation and initial functional evaluation of the Vaidurya search engine. Indeed, although the detailed evaluation described in this study was not performed within the DeGeL architecture, it was carried out within another repository of hierarchically organized, internally structured CPGs, the National Guideline Repository, as we describe in the Methods section.

A. Concept-Based Search

Many digital libraries are indexed in a hierarchical structure; examples include the well known web portal *Yahoo!*[1], using its own hierarchical concepts structure, and the *National Guideline Clearinghouse (NGC*[2]*)* library, using hierarchical conceptual structures*, Disease/Condition* and *Treatment/Intervention*, based on *Medical Subject Headings (MeSH*) [12] and *Unified Medical Language System (UMLS)* [2]. NGC uses *MeSH* produced by the *U.S. National Library of Medicine (NLM)* for both hierarchies, along with other controlled vocabularies, such as the *International Classification of Diseases (ICD)*, incorporated into NLM's

[1] www.yahoo.com

NLM's *UMLS* to classify disease concepts related to NGC guidelines and the U.S. *Health Care Financing Administration (HCFA) Common Procedure Coding System* and *ECRI's Universal Medical Device Nomenclature System (UMDNS)*, incorporated into NLM's UMLS to classify treatment/intervention concepts related to NGC guidelines.

These sites allow browsing through the concepts using the hierarchical structure, starting from the root and ending at the most specific concepts, located at the leaves. Such a browsing method forces the user to navigate the conceptual hierarchical structure. In these directories, searches can be limited to a specific concept [13,14] and its subconcept contents. However, while in Web directories documents are classified along one, two, or a small number of categories, in the medical domain documents are often classified by a multitude of concepts, often as many as a dozen or even tens of concepts, a property which can be further exploited for better retrieval.

In the medical domain, concept-based search refers to a text retrieval approach where the document is mapped to concepts based on its contents. Hersh's SAPHIRE system [3] uses an approach in which concepts used for indexing are automatically extracted from the document. Both documents and queries are often mapped, in the case of documents within biomedical domains into large vocabularies such as MeSH or UMLS, which is one of the major resources offered by the NLM. The concepts in these vocabularies are represented in a hierarchical structure, at the top of which the concepts are general, while their subconcepts are more detailed. However, users are not always familiar with the concepts in these vocabularies, rendering it somewhat limited. (The user's query can also be mapped into a concept query, and matched against all the indexed documents.) It has been previously noted that the particular implementation of concept-based search described above does not necessarily im-

[2] www.ngc.org

prove the retrieval performance [3, 4], compared to traditional text (word-based) retrieval techniques. Other studies have tried to exploit the UMLS meta-thesaurus to expand queries, thus extracting the concepts from the query terms [4]. The authors found that query expansion degraded aggregate retrieval performance, but some specific instances of synonyms and hierarchy-based query expansion improved individual query performance.

B. Context-Sensitive Search

Purcell suggested representing the clinical literature using contextual models [5, 6]. In her study, a group of domain experts marked up the text within a set of clinical journal articles using labels originating from a contextual model specific to the clinical research literature [6]. Purcell then showed that an improvement in retrieval performance could be achieved by using contextual search

In our study, the CPG representation using any particular target ontology is used as a contextual model, and a context-sensitive search, inspired by Purcell's approach, is first specified by the user, and then performed by Vaidurya. For example, the user might wish to search for the term "hypertension" only within the *filter condition* knowledge role, in the case of the Asbru CPG ontology, or within the *target population* element, in the case of the NGC ontology, which was used in this study.

C. The Vaidurya Search and Retrieval Model

Vaidurya offers a highly flexible query structure, in which both very simple and highly sophisticated queries can be built and applied. A query may contain specifications limiting the search to a subgroup of documents that are indexed by a conjunction or disjunction of one or more requested semantic axes within the concept-based search method. The query may in-

clude also ontology elements (contexts) from a selected ontology (in the case that the document was structured by that ontology) in which certain terms should appear within the context-sensitive search. These search methods are offered in addition to the full-text search, commonly used by search engines. In this section we very briefly describe Vaidurya. (A more complete description appears elsewhere [15]).

To perform a concept-based search, Vaidurya enables the user to optionally specify one or more concepts or subconcepts that might potentially be indexing the relevant guideline(s), using two logical operators, one defining the relations between the concepts appearing within the same semantic axis, and one defining the relationship between concepts appearing in different semantic axes. A *concept-based search query* is thus a collection of *concepts* and logical relations between them (conjunction or disjunction), explicitly specified by the user. When a query is executed, *all* of the documents classified by the specified concept and its descendent concepts (subconcepts), exploiting the *is-a* relations within the concept hierarchy, are retrieved. All of the documents retrieved using concepts that belong to the same semantic axis are either unified or intersected, depending on the logical relation specified by the user regarding concepts appearing in the same semantic axis, into a set of documents retrieved for the respective semantic axis. *In our study, documents indexed by concepts belonging to the same semantic axis were always unified.* (That is, all of the documents indexed by any of the internal concepts within the same semantic axis were retrieved for that axis). Regardless of the logical operator used to combine documents indexed by concepts appearing within the same semantic axis, a set of documents is retrieved for that axis.

After retrieving several sets of documents, each indexed by a semantic axis appearing in the query, the documents are combined, based on the logical operator used to combine sets of documents retrieved from different axis. In the case of the *conjunction* (AND) operator, the

document sets are intersected; in the case of the *disjunction* (OR) operator, the document sets are unified. In our evaluation, as we shall see, we examined the effect of both options regarding the method of combining documents retrieved from different semantic axes.

To perform a context-sensitive search, Vaidurya assumes the existence of an internal hierarchical structure of the document (i.e., an ontology that defines an internal contextual model). That is, it assumes that, whether manually or automatically, segments of the text have been labeled by elements from this internal hierarchical structure. The internal structure enables the user to query for keywords appearing only in specific contexts (i.e., within segments of text labeled only by the specified tag), thus potentially improving the search and retrieval accuracy.

Free-text queries, referring either to the full-text or to specified context elements, are ranked using the traditional approach in full-text information retrieval. In Vaidurya, each free-text content element is indexed by the terms appearing in that element content. Each text segment goes through a "*stopping*" procedure, in which common "*stop-words*" such as *AND, OR* are eliminated and the rest of the terms are *stemmed* [16] down to their linguistic root. Each term is assigned a *term frequency-inverse document frequency* (*tfidf*) value representing a combination of the term frequency in the document and in the entire collection [1, 15]. The ranking process is based on similarity functions, which measure the relevancy of each document element's content to the queried element. Similarity functions differ for each search type. We used mostly the *cosine* similarity function, suggested by Salton [1]. Eventually, after a rank is computed for each queried context and for the concept-based query, a total rank of each document is computed, based on the subqueries ranks and their weights (representing their contribution, in terms of retrieval).

In order to implement the ranking algorithm in Vaidurya, in which a query is a collection of different ontology elements, a *weight* was defined for each one. The *Ontology Element Weight* (*OEW*) is defined by an expert and represents the relative contribution and reliability of an ontology element. For example, elements containing a larger segment of text constitute a better presentation of the term frequency than those containing smaller segments of text, in which most of the terms have the same frequency (in such segments all the terms have the same appearance). An element should be more reliable if it has a larger segment of text, and as a result will have a greater weight. There are also other aspects involved in defining the relative ontology element weight, for example how clear the meaning of the element is to any user.

The Vaidurya Ranking Algorithm

Each query Q can be viewed as a collection of queries regarding various elements of the guideline ontology (including the indexing concepts). Thus, Q = $\{q_{elem1}, q_{elem2}\}$, where $q_{elem}i$, is a query on an ontology element whose ID is *i.* Each ontology element *i* has a weight $OEW_{elem}i$. The rank computed for each document is based on a similarity function that measures the relevance of the document to each queried element. Equation 4 shows how the rank of a document is computed based on each single ontology element similarity measure (rank) and its weight in query Q.

$$Rank(CPG,Q)=\sum_{i=1\to|Q|} OEW_i \times Sim(q_i, CPG.elem_i) \qquad (4)$$

A preliminary pre-evaluation of the Vaidurya engine was performed using part of the TREC-6 collection [17] in order to determine whether Vaidurya's full-text baseline retrieval performance was reasonable. The results were comparable to those of other search engines

applied to the TREC-6 collection. However, the TREC-6 collection is not in the medical domain, and is a flat one, supporting evaluation of only the full-text search method. Thus, neither the concept-based search nor the context-sensitive search methods could be evaluated by the TREC-6 collection.

D. The National Guideline Clearinghouse CPGs collection

For our detailed evaluation of the relative value of concept-based and context-sensitive search, we have used the *National Guideline Clearinghouse* (*NGC*) CPGs collection. The NGC website is a repository of CPGs classified by MeSH concepts. The CPGs are classified along two tree-like hierarchical concepts, *Disorders* and *Therapies*. Each concept tree has roughly 2,000 unique concepts in a tree-like structure; overall, 5407 concepts were used at the time of the evaluation. In several regions, the concept trees were 10 levels deep, but the average depth was around 4-6 levels. There were 1136 CPGs; each CPG might have multiple classifications indexed by both of the concept trees, including indices that belong to the same tree but at different levels (nodes), not necessarily leaves. CPGs have on average 10 classifications per guideline; there were CPGs that were classified by 90 concepts.

III. Methods

A. The Evaluation Data-Set

In order to use the NGC collection for an information-retrieval evaluation, we created a set of *information needs*, *queries*, and corresponding *judgments*. Several physicians at the Palo Alto Veterans Administration Health Care System, E&C-Medical Intelligence Ltd and a physician from the Division of Medical Informatics of the Israeli Defense Forces. Company defined 13

information needs (e.g., treatment of hypothyroidism in a particular population subset). Altogether, six physicians participated in the creation of the information needs, queries, and judgments. Each time, a subset of the physicians defined an information need; that group then agreed on the final information need. When the information need definitions were ready, these physicians scanned all of the guideline collection and identified CPGs that were relevant to the information needs (i.e., *judgments*). In order to evaluate the concept-based search and context-sensitive search methods, in addition to the full-text search, each information need was queried through a combination of a **full-text query** (**FTQ**), i.e., a list of keywords searched within the whole document, a **context-sensitive query** (**CSQ**), i.e., a query that searches for terms within three predefined context elements (knowledge roles that exist in the NGC ontology), and a **concept-based query** (**CBQ**), i.e., a list of concepts, all at the k^{th} level of the concept tree. Thus, each query consisted of three components, each in a different format, each of which could be used, on its own or in combination with one or more other components, to query the guideline database. The typical FTQ consisted of two or three terms after stop-word removal. (The FTQ can also be viewed as a private case of the CSQ, in which the context is the whole document.) Thus, each query could be applied by formulating it as an FTQ, CBQ or CSQ, as well as in any of their combinations, including different logical operators (disjunction, conjunction) among the query's terms.

We selected three elements from the NGC document structure: "*Target Population,*" "*Intervention and Practices considered,*" and "*Diseases and condition(s)*" for the CSQ and full-text query. "*Target Population*" defines population of interest such as "Patients undergoing noncardiac surgery" or "Children and adolescents through 18 years residing in the United States" "*Intervention and Practices considered*" covers diagnostic procedures such as "Pelvic ultrasonography" and management strategies such as "pneumoccocos vaccine" "*Diseases and*

condition(s)" lists conditions covered in the guideline such as "Measles" or "Angina." These elements are particularly meaningful when searching for a guideline to answer a clinical question that applies to one's patient, and thus were suggested by the participating clinicians. Each query element included different keywords, while the FTQ sometimes included keywords from all of the elements. The CBQ had four possible formulations applying: (1) concepts from the 2nd level of the conceptual hierarchy, using the disjunction logical operator; (2) concepts from the 2nd level of the conceptual hierarchy, using the conjunction logical operator; (3) concepts from the 3rd level of the conceptual hierarchy, using the disjunction operator; and (4) concepts from the 3rd level of the hierarchy, using the conjunction operator.

For example, a query using only full-text search would simply include the terms: "hypothyroid disease treatment." The context-sensitive formulation of the query included the terms: "hypothyroidism" required for inclusion in the "*Diseases and condition(s)*" context, "hypothyroidism adult dry skin" required for inclusion in the "*Target Population*" context, and "TSH THERAPY" required for inclusion in the "*Intervention and Practices considered*" context.

The concept-based query formulated at the second level of the concept hierarchy included the concepts "Diseases" and "Chemicals and Drugs." When formulated at the third level of the concept hierarchy, the concept-based query included, instead, the more specific concepts "Endocrine Diseases" and "Immunologic Diseases."

B. Evaluation Measures

In order to evaluate the retrieval performance, we used the traditional *precision* and *recall* metrics. Precision is the proportion of relevant CPGs (defined for a specific query within the entire collection, also called judgments) within the set of the retrieved CPGs, and recall is the

proportion of relevant CPGs (judgments) retrieved from the set of relevant CPGs (judgments), for a specific query. We also interpolated the averaged precision at eleven-points of recall levels 0.0, 0.1,..,1.0.

C. Statistical-Significance Tests

In order to perform *hypotheses significance tests* and facilitate a comparison of the results across multiple experiments, we used the 0.5 level of recall. At low levels of recall, the differences in performance are commonly great and unnecessarily indicate a real difference, while at high levels of recall, there is almost no difference in precision. Thus we chose to use the 0.5 level, which measures an intermediate level of recall. In each experiment, we calculated the mean of the 0.5 recall level and compared the different means using *t-tests*, a procedure often recommended for the evaluation of information retrieval systems [18]. The significance-test results of each experiment appear in the appendix.

IV. The Evaluation

In this experimental design, we posed two hypotheses.

A. Hypotheses

Hypothesis I: The more contextual elements used within a *context-sensitive query*, the better the performance.

Hypothesis II: *Concept-based search*, in addition to full-text or context-sensitive search, improves performance, especially when concepts in the query appear deeper within the concept hierarchy, and when using the conjunction logical operator.

B. Experimental Plan

In order to test the hypotheses within a wider scope, we defined subhypotheses within each one of the hypotheses. In *hypothesis I*, we wanted to learn whether the retrieval performance increases as more ontology elements (contexts) are included in the query. Thus, we evaluated it first using only CSQ queries by increasing the amount of elements each time (*hypothesis I.1*), while in a second experiment we used the CSQ in addition to the FTQ (*hypothesis I.2*).

In *hypothesis II*, we wanted to learn whether concept-based search in addition to context-sensitive (full-text is an instance of context-sensitive search) improves performance, especially when concepts in the query appear deeper within the concept hierarchy, and when using the conjunction logical relation rather than the disjunction operator. Thus, we evaluated the contribution of the concept-based search in relationship to several different baselines: full-text (*hypothesis II.1*), single-context query (*hypothesis II.2*), and three-contexts query (*hypothesis II.3*).

To test the set of hypotheses II, in which we wanted to assess the contribution of the concept-based search when using concepts from deeper levels of the hierarchy and when using the conjunction and disjunction logical operators, we formulated the same queries using five querying options. The first was the baseline formulation (1), which is the only search that varies within this set of hypotheses: full-text, single context, or three contexts. The second (2) and third (3) query formulations are CBQs that include concepts from the *second* level of the concept hierarchy (the depth of a concept being measured as levels downwards from level 0, which is the taxonomy root), and which use the disjunction and conjunction operators, respectively, in addition to the baseline querying method. The fourth (4) and fifth (5) options are CBQs that include concepts from the *third* level of the concepts hierarchy, and which use dis-

disjunction and conjunction operators, respectively, in addition to the baseline querying method. Thus, we had three experiments for each hypothesis, in which the four concept-based subqueries were used in addition to the given baseline.

V. Results

A. Context-Sensitive Search Results

In the first phase of the experiments, we wanted to test hypothesis I. At the beginning we ran each element separately, followed by a set of queries including two and three elements, where only context elements were queried. At the second step we used context queries in addition to the full-text search to assess its contribution. The results represent the mean performance over all queries.

1. Results of Queries Based Only on One or More Context Elements

Figure 1 shows precision-recall curves of the averaged single element queries (1 CSQ), in which each query contained terms to be searched within only a single context (i.e., an NGC ontological element). Two-element queries (2 CSQ) and three-element queries (3 CSQ) are shown as well. It can be seen that a *monotonic increase in precision* was achieved as more context elements were used in the query. Comparing the means of the precision values at the 0.5 recall level, within each querying option, showed that queries using three ontological elements outperformed *statistically significantly* the queries using only a single ontological element, as well as the queries using two ontological elements (see Table 1 of the Appendix).

The two-element queries outperformed the single-element query (see Table 1 of the Appendix).

2. Adding Context-Sensitive Search to Full-Text Search

Figure 2 shows the performance achieved in this experiment, in which we assessed the potential contribution of a context-sensitive search when added to an FTQ. It can be seen that when the contexts were queried in addition to the full-text queries (FTQ & 1,2,3 CSQ), the performance improved for all the options, compared to the full-text queries (FTQ) baseline.

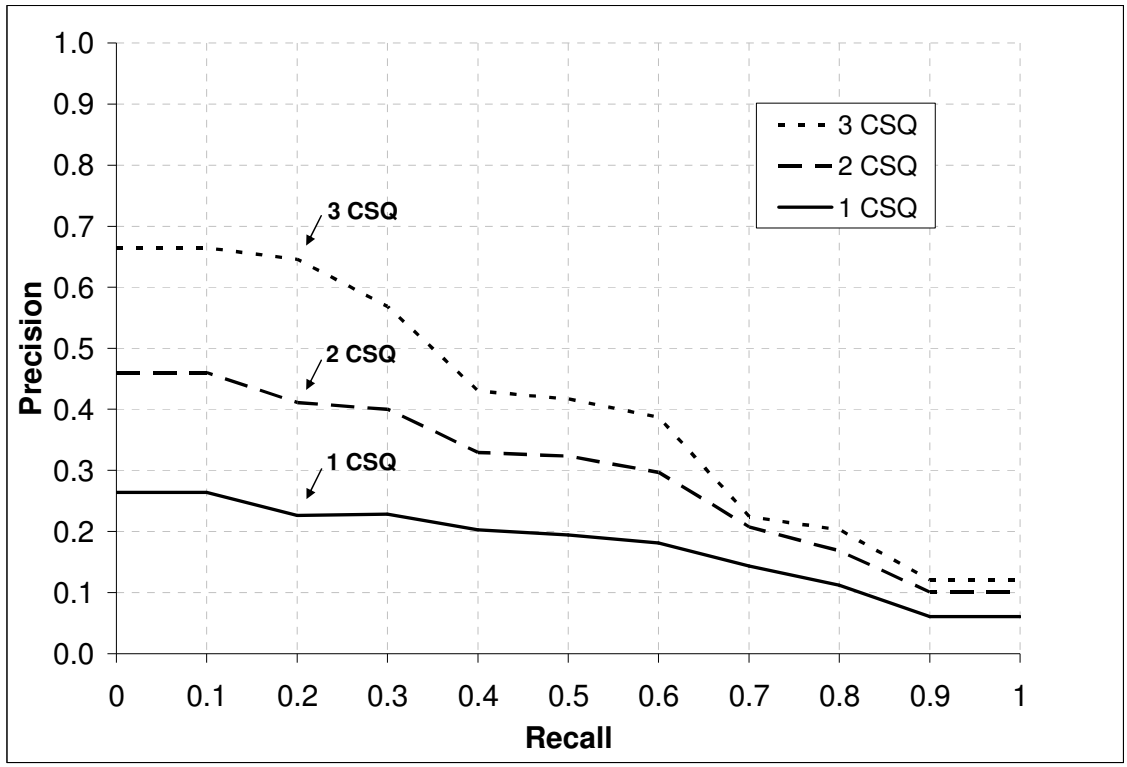

Figure 1. Precision and recall results for the mean performance of single-context queries (Legend: 1 CSQ), two-context queries (Legend: 2 CSQ), and three-context queries (Legend: 3 CSQ), demonstrating that as more ontological elements were used as contexts within the query, a better performance level was achieved.

While all the context-sensitive queries applied in addition to the FTQ outperformed the baseline (full-text), all the context queries which were used in addition to the full-text search performed similarly, and not one of them was significantly superior to the others. A comparison of the means of the precision at a 0.5 recall level, within each querying option, did not reveal any significant difference in performance, as shown in Table 2 of the Appendix.

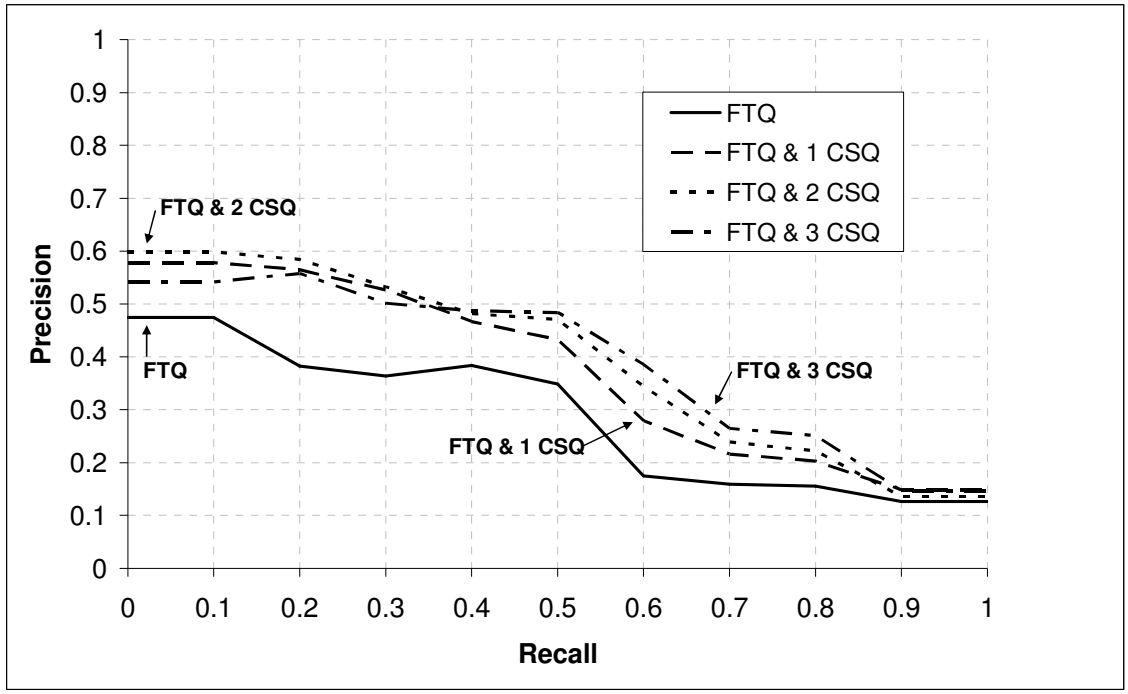

Figure 2. Precision and recall results for various combinations of context-sensitive queries (CSQs) in addition to the full-text query, which is used as the baseline ("FTQ"). Results signify the mean precision over all queries. The second curve ("FTQ & 1 CSQ") represents the mean performance achieved when a single-context element was added to the full-text query. The third curve ("FTQ & 2 CSQ ") represents the mean performance achieved for all two-context element combinations, in addition to the full-text. The fourth curve ("FTQ & 3 CSQ ") represents the mean performance when three-context elements were added to the query in addition to the full-text. As can be seen, at recall levels beyond 0.4, a monotonic improvement is achieved as more context elements are queried in addition to the full text (see text).

3. Intermediate Conclusions: Context-Sensitive Search

In the evaluation of the context-sensitive search mode, we examined the contribution of adding a context to a query, first when the queries included only CSQs, and then in addition to an FTQ. When only CSQ is used, a monotonic increase in performance was achieved as an increasing number of context elements were used in the query, with a mean of 11.1% improvement in the precision at the 0.5 recall level, when using two, as opposed to one, and three, as opposed to two contexts in the query. The highest mean precision at the 0.5 recall level was when using three contexts -- a mean of 41%. When CSQs were used in addition to full-text queries, a monotonic improvement in precision was observed beyond the 0.4 level of recall (i.e., when adding one, two, or three contexts to the query), with a mean of 4.5% at the 0.5 recall level. The highest precision at the 0.5 recall level was for the use of three contexts in addition to full-text search. The mean precision in this case was 48.4%.

B. Concept-Based Search Results

Note: the results of the experiments shown below are represented each time as five curves; in each case, only the meaning of the baseline query-formulation method changes; the other four curves are the same as described above (section IV.B), and are used in addition to the baseline query-formulation method.

1. Adding Concept-Based Search to Full-Text Search

Figure 3 shows the precision-recall curves. Curve (FTQ) presents the baseline performance of the full-text search. Curves (FTQ & 2,3 lvls Conj/Disj CBQ) present the results of a CBQ in addition to the full-text search, as explained earlier in Section IV.B (namely, querying using terms from the top 2 or 3 concept-hierarchy levels and, in each case, using disjunction or conjunction logical operators, respectively). The results shown in Figure 3 demonstrate a monotonic improvement in precision as the CBQ uses terms from a deeper level of the concept hierarchy, and when using the conjunction logical operator. Comparing the means of the precision at a 0.5 recall level within each querying option, showed a significant improvement when the CBQs were used with terms from the third level of the concept hierarchy in addition to a full-text query, compared to the use of terms from the second level and to using full-text queries alone, as shown in Table 3 of the Appendix.

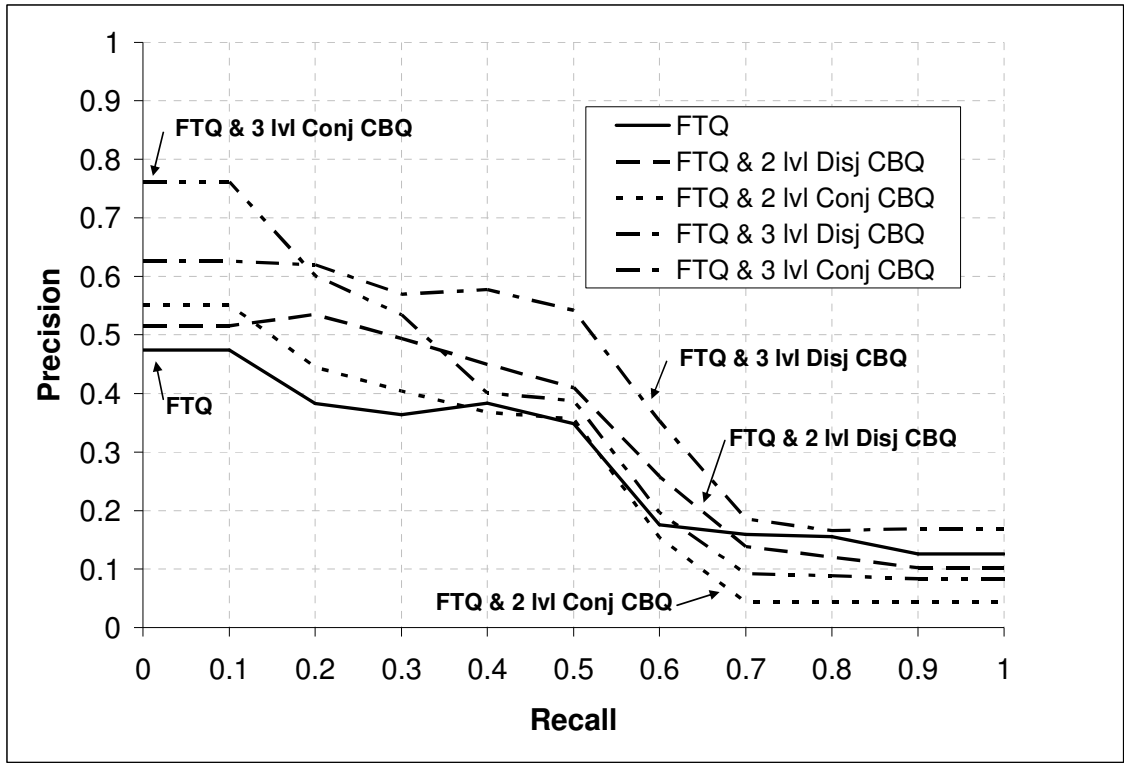

Figure 3. Mean precision and recall curves representing the results of various combinations of concept-based queries (CBQs) in addition to the baseline full-text queries. Curve (FTQ) presents the baseline performance of the full-text search. Curves (FTQ & 2,3 lvls Conj/Disj CBQ) present the result of a CBQ in addition to the full-text search, as explained in the text (namely, querying using the top 2 or 3 concept-hierarchy levels and, in each case, using disjunction or conjunction operators, respectively). A monotonic improvement can be seen as the concept-based queries include concepts from the deeper levels of the concept hierarchy. At low recall levels, queries using the conjunction operator perform better than queries using the disjunction operator.

2. Using Concept-Based Search in addition to a Single-Context Query

In the third set of experiments, our aim was to assess the contribution of the concept-based search to a single ontology element (in addition to a single context). The baseline performance in this case (1 CSQ) was the result of a CSQ including terms to be searched within a sin-

gle-context element. As explained earlier, Figure 4 shows the retrieval performance achieved when adding the CBQ to a single CSQ. In this set of experiments, adding a CBQ using terms from the second level of the concept hierarchy and using either the disjunction or the conjunction logical operators (2, 3, respectively) increased the retrieval performance at the low levels of recall and decreased it at the high levels. Curves (1 CSQ & 3 lvl Disj CBQ) and (1 CSQ & 3 lvl Conj CBQ) show the performance achieved for a query combining CBQ at the third level of the concepts hierarchy, with a specification of disjunction or conjunction within the concepts, respectively, in addition to a single CSQ. It can be seen that when a CBQ used concepts from both levels of the hierarchy (curves (1 CSQ & 2 lvl Disj/Conj CBQ) and (1 CSQ & 2 lvl Disj/Conj CBQ), an increase in retrieval performance was achieved at low levels of recall, and a decrease at high recall levels. Note that at both levels, the use of the disjunction logical operator was superior to the use of the conjunction operator.

Comparing the means of the precision at the 0.5 recall level within each querying option showed a statistically significant improvement in performance when the CBQ included concepts from the third level of the concept hierarchy using the disjunction logical operator, in addition to the single CSQ, compared to the queries using terms from the second level, or to the FTQ, as shown in Table 4 of the Appendix.

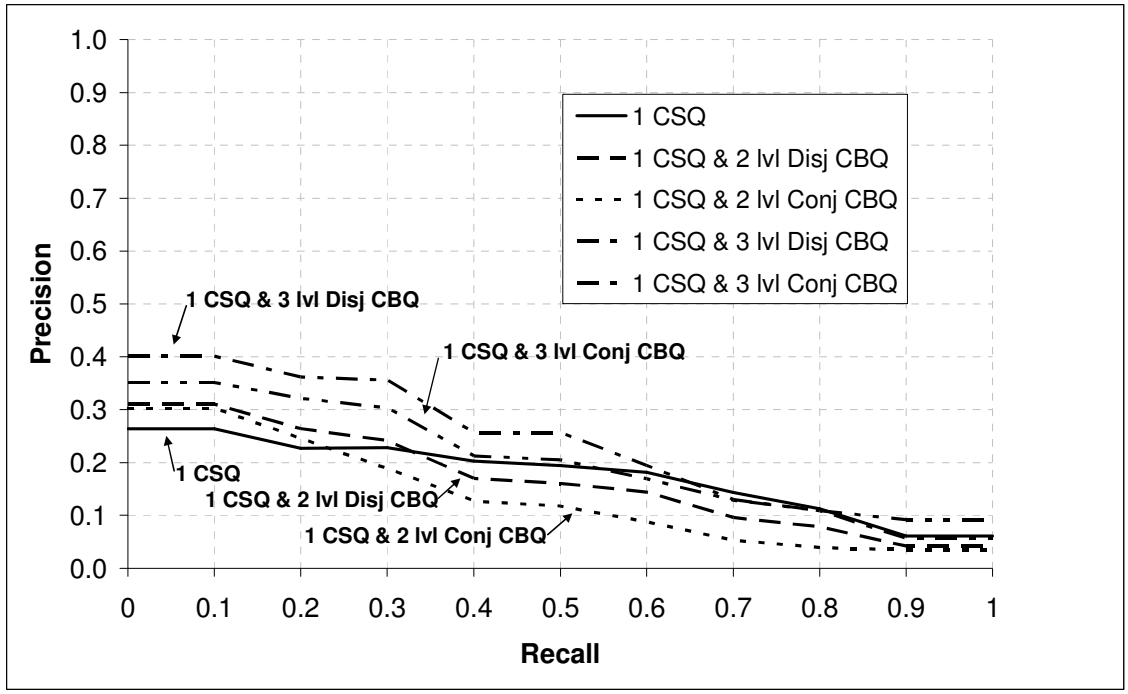

Figure 4. Mean precision and recall curves representing the results for various combinations of concept-based queries with terms coming from various levels of the concept hierarchy and using the conjunction or disjunction logial operators, in addition to a single-context-sensitive query. In general, using a CBQ improved the retrieval performance, when used in addition to the single CSQ (1 CSQ), at low levels of recall, but decreased it at high recall levels., At high levels of recall a decrease can be seen especially when using terms from the second level of the conceptual hierarchy (1 CSQ & 2 lvl Disj/Conj CBQ) (see text).

3. Concept-Based Search in addition to a Three-Element CSQ

In this set of experiments, which tests the part of the second hypothesis which refers to the contribution of the concept-based search, the baseline performance was defined as the precision based on answering a CSQ including terms to be searched within three context elements. Figure 5 shows that in this combination, the CBQ decreased the retrieval performance for *all the CBQ querying options* and at *all recall levels*. (See our comments in the Discussion).

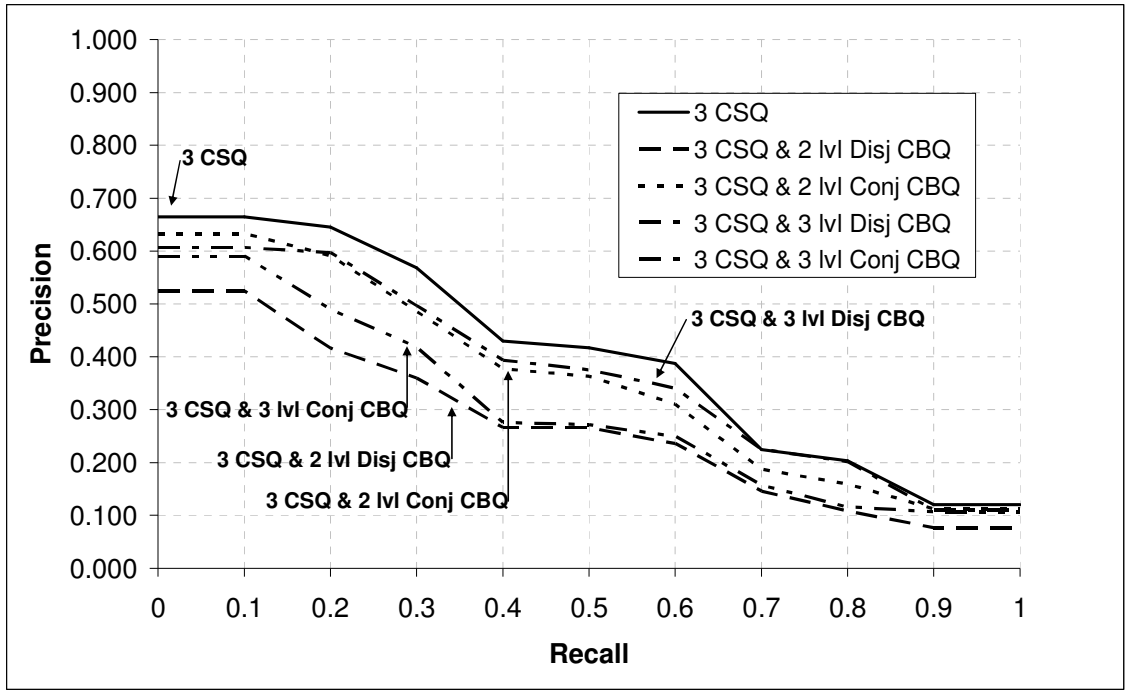

Figure 5. Precision and recall curves representing the results for various combinations of concept-based queries using terms from different levels of the concept hierarchy and the disjunction and conjunction logical operators, in addition to three-context elements queries. The baseline curve (3 CSQ) represents the three-context elements queries' performance. In this case, using concept-based queries decreased the performance relative to this baseline (see text).

4. Intermediate Conclusions: Concept-Based Search

In the evaluation of the contribution of concept-based search, we examined the contribution of concept-based queries, represented by two sets of concepts, one set from the second level of the concept hierarchy and another set from the third level of that hierarchy, using two types of logical operators. We examined the contribution of the CBQs in addition to a full-text search, a single-context query, and a three-contexts query.

Using concept-based search in addition to a full-text search improved precision by up to19.4% at the 0.5 recall level. The highest precision at the 0.5 recall level was when using a concept-based query with concepts from the third level of the hierarchy and with a disjunction operator, in addition to the use of a full-text search. The mean precision in this case was 54.2%. Using concept-based search in addition to only a single context in the query improved the performance only at the low levels of recall, while decreasing it at high recall levels. Using CBQ in addition to a three-context query decreased precision at any level of recall, possibly due to the initial relatively high level of precision when using three contexts in the query (which was 41.7%, in that case), as opposed to the lower initial precision achieved by the single-context query, in which the mean initial precision was 19.5%,.

VI. Discussion

Our main objective in this study was to compare quantitatively free-text, concept-based, and context-sensitive search, in particular for clinical documents. The framework within which we worked, namely the DeGeL digital guideline library and the Vaidurya search engine, enabled us to test explicitly several hypotheses regarding the relative added value of each search type.

The DeGeL guideline library, although highly suitable for the Vaidurya search engine's purposes, does not yet have by itself a sizable collection of clinical guidelines, and so could not be used directly in the evaluation. One option for evaluation of the concept-based search might have been to use the OHSUMED test collection [19], which is based on the clinical literature, and is composed mainly of clinical journal articles. The documents in the OHSUMED collection include a MeSH concepts field extracted from each document, and

thus, could be used for evaluation of the concept-based search. However, the OHSUMED documents are not structured; thus, we could not evaluate the context-sensitive search method. We preferred a test collection of semi-structured documents, which are also classified along multiple hierarchical concepts. The NGC guideline repository was quite suitable for that purpose. Thus, we downloaded guidelines from that repository into the DeGeL architecture to test our various research hypotheses, using the Vaidurya search engine.

The markup process of a document into a predefined ontology that represents a meaningful structure of the document, as in the case of the NGC repository or the DeGeL architecture, presents us with an exceptional opportunity to query a digital medical-document library using context-sensitive methods. Furthermore, indexing the documents along multiple hierarchical semantic axes, either manually, as in the case of the DEGEL framework, or automatically, as we have shown in another study [20], enabled us to apply the concept-based search, where the user explicitly indicates one or more concepts from one or more semantic axes, in the query.

Our experimental results show that concept-based search improved the retrieval performance mainly when used in addition to full-text queries, as well as leading to partial improvement when used in addition to context-sensitive queries. Even in the context of a naturally limited small set of clinician queries and judgments, several of the results were found statistically significant (see appendix), especially when comparing concept-based queries using concepts from the third level of the hierarchy, to queries using concepts from the second level of the hierarchy, and when comparing both cases to the results of the baseline full-text query. When concept-based queries were used in addition to context-sensitive queries, the results were typically not statistically significant. However, meaningful trends were observed;

in particular, concept-based queries consistently improved the baseline level of single element context-sensitive queries at low levels of recall.

Furthermore, as hypothesized, a monotonic increase in performance was achieved as an increasing number of context elements were used in formulating the query, up to the use of three contexts in the query (see Figure 1). When context-based queries were used in addition to full-text queries, a monotonic improvement in Precision (with respect to adding one, two, or three contexts to the query) was also observed, as more contexts were used in the query formulation (see Figure 2).

In the case of adding concept-based search to a context-sensitive query using three ontological elements, the result seemed actually to *decrease* the precision at all levels of recall; this phenomenon was observed to a much lesser degree when adding a CBQ to a single-element context-sensitive query, appearing especially at high levels of recall. The reason for the difference between these two scenarios might be that the initial, baseline precision in the first case (i.e., when using the three context elements) was much higher than that when using a single-element context query (see Figures 4 and 5).

It should be noted that studies examining concept-based retrieval have shown that it does not necessarily improve, and might even worsen, the search-engine's performance [21, 22]. This phenomenon might be due to the fact that current automated extraction modules are not yet sufficiently accurate, and users usually are not familiar with all the concepts when entering keywords. In the case of our concept-based search and retrieval setup, the CPGs of the NGC repository were indexed *manually* by a multiple-hierarchy indexing scheme, and the query concepts were chosen *explicitly* from the conceptual hierarchy by the user, demonstrating significant improvements in the retrieval performance in several cases.

A previous study by Purcell [6] of contextual modeling for retrieval of clinical literature demonstrated slight improvements in the precision of the retrieved documents when contexts were used in the query. In the current study, we used the knowledge roles (e.g., *entry conditions*) of the ontologies which were used to structure the clinical guidelines as the contextual elements. The results seem to validate and extend the intuition suggested by Purcell.

Our context-sensitive evaluation indicated that a search based on a single ontology element alone results in a poorer performance than one based on the full text; however, for specific queries, we noticed significant improvement in precision. As mentioned, the more contexts were used in formulating the query, the better the resulting precision.

Thus, given the experience gained through the evaluation, our suggestion is that initially a full-text search should be used. When this option fails, or results in low performance, it seems beneficial to use the concept-based search, which usually improves performance, especially when the baseline precision seems low. Further use of context-sensitive queries may then improve the precision significantly when several elements (knowledge roles) of the ontology are queried simultaneously, especially in addition to the full-text search.

Our experience indicates that this querying approach enables performance of a more specific search that may limit the search and retrieval process while avoiding the multiple irrelevant results from which most search engines suffer. The disadvantages of a context-sensitive query are the need for a more complex query interface and a higher demand on the user's time. The concept-based search, however, enables the user to limit the search to a specific subset of the repository, which increases the precision of the returned results, while the necessary interface, although not trivial, is not overly cumbersome.

The limitations of this study are mainly caused by the test collection. Our test collection, compared to common information retrieval test collections, such as the TREC repository [17], in which there are millions of documents, is relatively small. However, in contrast to the case of using huge test collections, such as TREC, in which judgments (i.e., decisions of which documents are relevant to a query) are specified *automatically*, based on an ensemble of search engines, in our test collection the reviewers browsed *manually* each of the CPGs in the test collection and indicated their relevance to the query. Another limitation was the number of queries, which was relatively low; however, this amount of queries was usually sufficient for performance of statistically meaningful comparisons, although we expect that additional queries would have resulted in more statistically significant results. Although the use of structured clinical guidelines might limit somewhat the ability to generalize the results of the context-sensitive search, the results of the concept-based search can be quite easily generalized, since they are based on a generic indexing approach that is independent of the explicit contents of the documents.

VII. Conclusions

We have demonstrated the usefulness of concept-based and context-sensitive queries for enhancing the precision of retrieval of documents from a large digital library, in this case a repository of clinical guidelines semi-structured by predefined ontological knowledge roles. A concept-based search usually improved performance over the use of free-text queries, especially when the baseline precision was low. Further use of context-sensitive queries might significantly improve the retrieval precision when several elements (knowledge roles) of the ontology are queried simultaneously. In general, the more ontological elements used, the greater the resulting precision.

Although this study has focused on searching within a library of procedural clinical knowledge, namely clinical guidelines, its methodology and results are potentially relevant to other types of clinical documents. In particular, it is relevant whenever a concept-based search is applicable, namely, when a hierarchical indexing scheme exists (such as when indexing medical research papers using MeSH), or when a manual or automated internal markup (structuring) of the document by a hierarchy of meaningful context elements was performed.

We are in the process of experimenting with a possible generalization of the concept-based search to search in libraries indexed along domain taxonomies, in which the user is not required to specify explicitly the concepts by which the documents she is looking for might be indexed, to avoid the complicated query interface and the familiarity with the concepts hierarchies. We plan to evaluate this approach on a significantly wider test collection, such as the *TREC-Genomics* repository.

Acknowledgements

This research was supported in part by NIH award No. LM-06806. We thank Mr. Tu and Dr. Peleg, from Stanford BioMedical Informatics, for useful discussions regarding the need for supporting the use of multiple guideline ontologies. Drs. Shiffman and Karras from Yale University and the University of Seattle assisted us in using the GEM ontology. Drs. Goldstein, Basso, Kaizer, and Advani, from Stanford University and the Veterans Administration, Palo Alto Health Care System, and Dr. Lunenfeld from the Soroka Medical Center, were extremely helpful in assessing the various interfaces and search. We want to thank the physicians from E&C Intelligence Ltd., who did this extremely time consuming work in the creation of the test collection, Drs Ben-Yosef, Sidikmam, and Zaltsman. We thank The Ben Gurion University Information Systems Department undergraduate students A. Litmanovitz, O. Bohanna, and C. Sasson, who worked on the IR and HCI evaluation tools. We thank Drs. Shapira, Kuflik and Goren-Bar from Ben Gurion University and Haifa University for useful discussions at the initial steps of the development of Vaidurya. Finally, we would like to thank the anonymous reviewers for their significant and helpful comments.

References

[1] Salton G and McGill MJ. Introduction to Modern Retrieval. McGraw-Hill Book Company, 1983

[2] Humphreys BL and Lindberg DA. The UMLS project: making the conceptual connection between users and the information they need. Bull Med Libr Assoc. 1993 Apr;81(2):170-177.

[3] Hersh WR, and Greens RA. SAPHIRE – an information retrieval system featuring concept matching, automatic indexing, probabilistic retrieval and hierarchical relationships. Computers and Biomedical Research 1989;23:410-25.

[4] Hersh WR, Price S, Donohoe L, Assessing thesaurus-based query expansion using the UMLS Metathesaurus. Proceedings of the 2000 Annual AMIA Fall Symposium, 2000, 344-348.

[5] Purcell GP, Contextual document models for searching the clinical literature [dissertation] Stanford Medical Informatics; 1996.

[6] Purcell GP, Rennels GD, and Shortliffe EH. Development and evaluation of a context-based document representation for searching the medical literature. International Journal of Digital Libraries 1997; 1:288-296.

[7] Moskovitch R, Hessing A, and Shahar Y (2004) Vaidurya – A concept-based, context-sensitive search engine for clinical guidelines. Proceedings of Medinfo-04, San Francisco, CA. 2004.

[8] Shahar Y, Young O, Shalom E, Galperin M, Mayaffit M, Moskovitch R and Hessing A. A framework for a distributed, hybrid, multiple-ontology clinical-guideline library and automated guideline-support tools. Journal of Biomedical Informatics 2004;37:325-344.

[9] Shiffman RN, Karras BT, Agrawal A, Chen R, Marenco L, Nath S. GEM: a proposal for a more comprehensive guideline document model using XML. Journal of the American Medical Informatics Association 2004;7(5):488-498]

[10] Shahar Y, Miksch S, and Johnson P. The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. Artificial Intelligence in Medicine 1998;14:29-51.

[11] Peleg M, Boxwala A, Bernstam E, Tu SW, Greenes RA and Shortliffe EH. Sharable representation of clinical guidelines in GLIF: Relationship to the Arden syntax. Journal of Biomedical Informatics 2001;34:170-181.

[12] Medical subject headings (MeSH). National Library of Medicine, 2003. http://www.nlm.nih.gov/mesh.

[13] Concard J, Yang C, and Claussen J. Effective collection metasearch in a hierarchical environment: Global vs. localized retrieval performance. *In Proc. of SIGIR 02, 371-372.*

[14] Wang W, Meng W, and Yu C. Concept hierarchy based text database categorization in a metasearch engine environment. *In Proc. Of WISE '00, June 2000.*

[15] Moskovitch R. A concept-based, context-sensitive, multiple-ontology search engine for clinical guidelines. [M.Sc. dissertation]. Ben Gurion University; 2005.

[16] Porter MF. An algorithm for suffix stripping. Program 1980; 14(3):130-137.

[17] Text REtrieval Conference [http://trec.nist.gov/]

[18] Sanderson M and Zobel J. Information retrieval system evaluation: effort, sensitivity, and reliability. Proceedings of the 28th ACM SIGIR Conference, 2005.

[19] Hersh WR, Buckley C, Leone TJ, Hickam DH, OHSUMED: an interactive retrieval evaluation and new large test collection for research, Proceedings of the 17th Annual

ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, 192-201.

[20] Moskovitch R, Cohen-Kashi S, Dror U, Levy I, Maimon A, and Shahar Y. Multiple hierarchical classification of free-text clinical guidelines. Artificial Intelligence in Medicine Journal, 2004b. In print (Expected July 2006).

[21] Hersh WR and Hickam DH. A comparison of retrieval effectiveness for three methods of indexing medical literature. The American Journal of the Medical Sciences 1992a; 303(5):292-300.

[22] Hersh WR and Hickam D.H. A comparison of two methods for indexing and retrieval from a full text medical database. Medical Decision Making 1993; 13(3):220-26.

APPENDIX

Significance Tables for the Precision of Different Query Types

The tables below refer to the results figures mentioned in the text. Each table presents the results of the t-test comparisons performed. Each column and row refers to a search mode, including the mean performance and standard deviation at the 0.5 level of recall in brackets (mean ± SD). In addition we display below the mean and SD also the 95% confidence interval. Each cell presents the t-test comparison result as described in the index below:

<<<(*pv*) - denotes that the method listed at the header of the column of the cell outperforms the one listed at the header of the row of the cell, having a p-value lower than 0.05, indicating the p-value in the brackets.

<<(*pv*) - denotes that the method listed at the header of the column of the cell outperforms the one listed at the header of the row of the cell, having a p-value lower than 0.1 (and higher than 0.05), indicating the p-value in the brackets.

<(*pv*) - denotes that the method listed at the header of the column of the cell outperforms the one listed at the header of the row of the cell, indicating the p-value in the brackets.

>- denotes that the method listed at the header of the row of the cell outperformed the method listed at the header of the column of the cell.

Table 1 – Context-sensitive search using three, two and single contexts

Options	**2 CSQ** (0.32 ± 0.13) [0.27, 0.38]	**3 CSQ** (0.42 ± 0.17) [0.31, 0.53]
1 CSQ (0.2 ± 0.09) [0.15, 0.24]	<<(0.05)	<<<(0.02)
2 CSQ (0.32 ± 0.13) [0.27, 0.38]		<(0.2)

Table 2 – Context-sensitive query in addition to full-text queries

Options	**FTQ & 1 CSQ** (0.43 ± 0.1) [0.38, 0.48]	**FTQ & 2 CSQ** (0.47 ± 0.13) [0.41, 0.53]	**FTQ & 3 CSQ** (0.48 ± 0.17) [0.37, 0.60]
FTQ (0.35 ± 0.06) [0.28, 0.41]	<(0.19)	<(0.13)	<(0.15)
FTQ & 1 CSQ (0.43 ± 0.1) [0.38, 0.48]		<(0.30)	<(0.32)
FTQ & 2 CSQ (0.47 ± 0.13) [0.41, 0.53]			<(0.45)

Table 3 – The effect of adding a concept-based query to a full-text query

Options	**FTQ & 2 lvl Disj** (0.41 ± 0.07) [0.34, 0.48]	**FTQ & 2 lvl Conj** (0.35 ± 0.14) [0.26, 0.46]	**FTQ & 3 lvl Disj** (0.54 ± 0.07) [0.47, 0.62]	**FTQ & 3 lvl Conj** (0.38 ± 0.16) [0.28, 0.50]
FTQ (0.34 ± 0.06) [0.28, 0.41]	< (0.27)	< (0.47)	<<(0.03)	< (0.38)
FTQ & 2 lvl Disj (0.41 ± 0.07) [0.34, 0.48]		>	<(0.11)	>
FTQ & 2 lvl Conj (0.35 ± 0.14) [0.26, 0.46]			<(0.15)	<(0.43)
FTQ & 3 lvl Disj (0.54 ± 0.07) [0.47, 0.62]				>

Table 4 – The effect of adding a concept-based query to a query including a single context element

Options	**1 CSQ & 2 lvl Disj** (0.16 ± 0.06) [0.12, 0.2]	**1 CSQ & 2 lvl Conj** (0.11 ± 0.06) [0.08, 0.15]	**3 CSQ & 3 lvl Disj** (0.25 ± 0.09) [0.21, 0.31]	**3 CSQ & 3 lvl Conj** (0.2 ± 0.1) [0.15, 0.26]
1 CSQ (0.19 ± 0.09) [0.15, 0.24]	>	>	<(0.18)	<(0.44)
1 CSQ & 2 lvl Disj (0.16 ± 0.06) [0.12, 0.2]		>	<<<(0.06)	<(0.24)
1 CSQ & 2 lvl Conj (0.11 ± 0.06) [0.08, 0.15]			<<<(0.01)	<<(0.08)
1 CSQ & 3 lvl Disj (0.25 ± 0.09) [0.21, 0.31]				>

Table 5 – The effect of adding a concept-based query to a query including three contexts query

Options	**3 CSQ & 2 lvl Disj** (0.26 ± 0.16) [0.16, 0.38]	**3 CSQ & 2 lvl Conj** (0.36 ± 0.15) [0.26, 0.47]	**3 CSQ & 3 lvl Disj** (0.37 ± 0.13) [0.28, 0.48]	**3 CSQ & 3 lvl Conj** (0.27 ± 0.16) [0.16, 0.38]
3 CSQ (0.41 ± 0.17) [0.31, 0.53]	>	>	>	>
3 CSQ & 2 lvl Disj (0.26 ± 0.16) [0.16, 0.38]		<(0.26)	<(0.23)	<(0.48)
3 CSQ & 2 lvl Conj (0.36 ± 0.15) [0.26, 0.47]			< (0.46)	>
3 CSQ & 3 lvl Disj (0.37 ± 0.13) [0.28, 0.48]				>