# Vaidurya: A Multiple-Ontology, Concept-Based,

# Context-Sensitive Clinical-Guideline Search Engine

**Robert Moskovitch and Yuval Shahar**

*Medical Informatics Research Center*

*Department of Information Systems Engineering*

*Ben Gurion University, P.O.B. 653, Beer Sheva 84105, 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*

*We designed and implemented a generic search engine (*Vaidurya*), as part of our Digital clinical-Guideline Library* (DeGeL) *framework. Two search methods were implemented in addition to full-text search: (1) concept-based search, which relies on pre-indexing the guidelines in a clinically meaningful fashion, and (2) context-sensitive search, which relies on first semi-structuring the guidelines according to a given ontology, then searching for terms within specific labeled text segments. The Vaidurya engine is fully functional and is used within the DeGeL system. We describe the Vaidurya ontological and algorithmic framework; we also briefly summarize the results of a detailed evaluation in the clinical-guideline domain, demonstrating that both concept-based and context-sensitive ontology-independent search are highly feasible and significantly improve on free-text-search retrieval performance. We conclude by analyzing the limitations and advantages of the approach, and the steps that we have started to take to extend it based on user feedback.*

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

## Introduction

Effective search and retrieval of relevant documents has become crucial with the recent growth in digital literature, in general, and the medical literature, in particular [1]. The requirement for effective retrieval is especially crucial when searching for a clinical guideline to manage a particular patient or a group of patients at the point of care, or when assessing retrospectively the quality of care that should have been administered according to one or more established state-of-the-art guidelines. The need has become further emphasized by the recent trend towards provision of automated support to guideline-based clinical care, and search engines must consider the special requirements introduced by the process of structuring guidelines for such a purpose. As we demonstrate in our study, exploitation of that structuring is both quite feasible and beneficial.

Typically, documents are retrieved using traditional full-text search, as originally described by Salton et al [2], however this approach is not accurate enough and doesn't exploit the potential existence of rich domain-specific knowledge, which is common in the medical domain, and especially when searching for clinical guidelines appropriate for application to a specific patient.

For example, the medical domain offers controlled vocabularies and various tools for using them, such as the Unified Medical Language System (UMLS) [3]. Current search methods, such as the one used in Medline, index documents according to the MeSH conceptual hierarchy [4], either manually or by automatic means, such as the MetaMap tool [5]. Thus, a textual query is also converted into a set of representative concepts, which are matched to the indexed documents, leading to a *concept-based search*. Such approaches, however, do not fully exploit the hierarchical relations among the concepts, such as the semantics of parent and child links, and thus do not necessarily lead to an improvement over traditional full-text search [6].

Furthermore, medical documents often have an explicit and meaningful internal structure. For example, medical documents usually include sections such as "Introduction" and "Discussion," while clinical-guideline documents often contain even more semantically meaningful segments, such as "Description of the eligible population" or "Objectives". Such a structure can support a context-sensitive search, in which terms are searched only within certain meaningful segments of the documents, potentially enhancing the precision of the retrieval process.

Both of these avenues for enhancement of the search precision have been explored in our study and were found to indeed be fruitful. In this paper, we describe in detail Vaidurya – a search engine that uses concept-based and context-sensitive search in addition to the traditional full text search. For example, Vaidurya explicitly capitalizes on the existence of external conceptual hierarchies, exploits the semantics of their inner structure, and enables the user to specify certain logical relations among them.

Vaidurya [7] is a search engine originally developed for retrieval of clinical guidelines within our digital guideline library, DeGeL [8], a set of Web-based tools for structuring, search, retrieval, and application of clinical guidelines. As we describe below, retrieval of suitable guidelines for care of particular chronic patients is an important and highly motivating objective. However, the Vaidurya search engine was designed to be quite generic, and is appropriate for search and retrieval within any collection of structured and indexed documents for which certain ontological assumptions hold, a situation that is most common in medical domains. Other examples include collections of articles in biomedical domains, as we had recently demonstrated in the case of the Trec Genomics collection [9]. Throughout this paper, we will exemplify our new approach to concept-based and context-sensitive search using the example of searching within a clinical-guideline library, a sub-domain in which we had rigorously and quantitatively evaluated our framework. For the purposes of that evaluation, reported elsewhere [10, 11], we used the *National Guideline Clearinghouse* (*NGC*[1]) clinical-guideline collection, and its internal hierarchical structure, which we briefly describe in the Evaluation section.

In this paper, we present in detail the Vaidurya system's computational infrastructure and present briefly an integrated view of the results of its evaluation. We start by introducing the domain of supporting clinical-guideline retrieval and application; then we proceed with introducing concept-based and context-sensitive search; we continue by describing Vaidurya's computational architecture and by summarizing the results of its evaluation in the guideline domain. Although we had evaluated the methodology within a particular guideline ontology and set of conceptual indices, the results are independent of the particular guideline ontology and conceptual hierarchy used. We conclude by presenting our overall insights regarding the limitations and advantages of the Vaidurya framework, and the specific measures that we had already taken to overcome some of its limitations.

## Background

We start by introducing the need for effective search and retrieval of electronically represented clinical practice guidelines. We then introduce in more detail the ideas of concept-based search and context-sensitive search, before focusing on our specific methods.

[1] http://www.guideline.gov

## Clinical Practice Guidelines

*Clinical Practice Guidelines* (*CPGs*) , are defined by the Institute of Medicine as “systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances” [12,13]. CPGs are most useful at the point of care, in particular, when the care provider prescribes a specific care plan. Extensive evidence exists that conforming to state-of-the-art guidelines improves the quality of medical care [14,15,16], while potentially, at least in the longer term, reducing its escalating costs.

Currently, CPGs are usually represented in free text, which provides little support for automated application of the CPGs at the point of care. Several structured representation formats, or CPG *ontologies*, have been proposed; their objective is to represent clinical guidelines in a semi-structured, and eventually a completely structured, machine-comprehensible format [12, 17, 18, 19, 20, 21, 22, 23]. However, currently most guidelines exist in a free-text or a semi-structured textual format.

Even if the CPGs are organized within a digital library, it is of paramount importance to search and retrieve them effectively and with high precision relative to the user's needs, whether these include therapy of a particular patient, structuring of a certain CPG, or a retrospective assessment of the quality of care of a group of patients relative to a particular guideline or a set of guidelines.

The growing interest in the use of CPGs and their adoption had led to the creation of several CPG web sites and search engines. The *Guideline International Network* (G-I-N[2]), which includes CPGs from a wide list of countries, enables its members to browse and perform full

[2] http://www.g-i-n.net

text search. The *Guidelines Finder*[3] contains CPGs from the UK and offers a basic text-based search to access free-text CPGs. The *NGC*, an initiative of the *Agency for Health Research and Quality* (AHRQ), contains CPGs from different web portals in a uniform internal structure that are indexed along MeSH concepts; as we describe later in this paper, we used the NGC resource extensively in our evaluation. However, unlike the DeGeL library, most digital libraries provide mainly unstructured free-text CPGs and for the most part offer basic search techniques, based on terms and keywords.

The DeGeL library [8] represents CPGs using a *meta-ontology* format as well as a set of specific guideline ontologies. Ontology independent elements, such as documentation details and semantic classification indices, are common to every CPG, regardless of the specific ontology, such as Asbru, which is used to represent the CPG's content.

To support *concept-based* indexing and search, seven *semantic axes* (concept hierarchies) are defined within DeGeL, based on MeSH and UMLS, For example, *disorders* (e.g., ischemic heart disease), *therapies* (e.g., irradiation), *symptoms and signs* (e.g., abdominal pain and elevated rectal temperature, respectively), etc. Each CPG can be classified, using the *IndexiGuide* tool [8], along multiple subconcepts—one or more within each semantic axis.

To support the conversion of free-text CPGs into a machine-comprehensible representation, and to support *context-sensitive* search and retrieval of the guidelines, the DeGel architecture includes a set of web-based tools for incremental conversion of a CPG into one of a several CPG-specification ontologies. The DeGeL tools enable a team of a clinical editor and a knowledge engineer to gradually structure a guideline into increasingly formal formats at several representation levels, working separately and/or together (depending on the representa-

[3] http://libraries.nelh.nhs.uk/guidelinesFinder

tion level). The CPG specification process starts with a semantic markup process, using a structuring (*mark-up*) tool called URUZ [8], in which segments of the CPG's text are labeled by semantic tags from the chosen guideline ontology (e.g., "eligibility conditions" or "outcome objectives") of one of the CPG ontologies implemented in DeGeL. These labeled segments, each representing a context, are highly useful for context-sensitive retrieval of the guideline (as well as for supporting its automated application). The DeGeL library's four guideline representation levels (formats) include a full text format, a semi-structured text format (labeled by guideline-ontological semantic tags), a semi-formal representation (including control operators such as *In Parallel*) and a fully-structured, machine-executable representation. The Vaidurya search engine uses the full-text and the semi-structured representations.

## Concept-Based Search

Many digital libraries are indexed in a hierarchical structure; examples include the well known web portal *Yahoo!*, which uses its own hierarchical concepts structure, and the NGC library, which uses MeSH- and UMLS-based hierarchical conceptual structures: *Disorders* and *Therapies*. These sites allow the user to browse through the concepts using the hierarchical structure, starting from its roots, and ending at the most specific concepts, located at the leaves. Thus, such a browsing method forces the user to navigate the conceptual hierarchical structure. In these directories searches can be limited to a specific concept and its sub-concept contents. However, while in Web directories documents are classified into 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 tens of concepts, existing within potentially quite different branches of the hierarchical tree (e.g., a CPG might be indexed by several intermediate or leaf-nodes within Disorders as well as within Therapies and within Anatomical sites).

Traditional text-retrieval systems use the *Vector Space Model* introduced by Salton [2], implemented within the SMART system, in which terms are extracted from the document and represented by their term frequency as a bag of words. More modern representations for full text retrieval presented by Singhal et al [24]. The limitation of this approach is that humans search in terms of *concepts* [25]. However, for practical reasons since most of the search engine search expect a keywords search phrase the users search for keywords which represent their conceptual search. In the medical domain, concept-based search refers to a text-retrieval approach in which the document is mapped to concepts based on its contents. Hersh's SAPHIRE system [6] uses an approach in which concepts used for indexing are automatically extracted from the document. Gobeill et al present a framework for multi-lingual retrieval based on the multi-lingual version of MeSH [26]. In the case of documents within biomedical domains, documents and queries are often mapped into large vocabularies such as the *Medical Subject Headings* (MeSH) [4] or *Unified Medical Language System* (UMLS) [3], which is one of the major resources offered by the *National Library of Medicine* (NLM). The concepts in these vocabularies are represented in a hierarchical structure, in which the concepts at the top are general and their sub-concepts are more detailed. This approach is somewhat limited, since users are not always familiar with the concepts in these vocabularies. Furthermore, previous studies have shown that, compared to traditional text (word-based) retrieval techniques, such an implementation of concept-based search does not necessarily improve the retrieval performance [27, 28]. Other studies tried to exploit the UMLS meta-thesaurus to expand queries, thus extracting the concepts from the query terms [29, 8]. 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. Aronson [5]compared his methods to Srinivasan's [30] and showed an improvement by expanding text-based queries with both phrases and concepts from the UMLS Metathesaurus. Neither used the hierarchical relationships of the Metathesaurus, yet reported an improvement over a non-query-expanded baseline. Rada [31] developed an algorithm to estimate the conceptual distance between documents and queries using MeSH and suggested that MeSH can be utilized to improve retrieval performance.

**<FIGURE 1>**

**Figure 1**: The concept "Asthma" in MeSH, decomposed to tree-numbers, appearing in several locations in different concept trees.

## Medical Subject Headings

The MeSH thesaurus is organized hierarchically and includes 15 concept "trees", consisting of descriptors (concepts). Examples of such trees include *Anatomy* and *Diseases*. At the most general level of the hierarchical structure there are very broad concepts (e.g, 'Anatomy', 'Mental Disorders'). More specific headings are found at lower levels of the eleven levels hierarchy (e.g., 'Ankle' and 'Conduct Disorder'). The same concept may appear as multiple *tree-numbers* within MeSH, thus, in effect, having multiple ancestors. For example, the concept *Asthma* has *four* tree-numbers in MeSH, having three different parents (figure 1). Note that intermediate concepts such as *Respiratory Hypersensitivity* can also appear more than once as different *tree numbers*, possibly at different levels (figure 1).

## Context-Sensitive Search

It is theoretically possible to search for certain terms only when these appear within a specific context, or structured type, of the text. Purcell suggested representing the clinical literature using contextual models [32, 33]. 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. Purcell then showed that an improvement in retrieval performance could be achieved by using contextual search [33].

To perform a context-sensitive search in Vaidurya, documents are expected to be structured according to a relevant [typically, hierarchical] contextual model. For example, in the case of clinical guidelines, a clinical editor created these contexts by labeling the free text using a predefined set of labels (e.g., eligibility conditions) chosen from one of several standard hierarchical guideline representation formats. Such structuring, in the case of clinical guidelines, uses tools such as DeGeL's markup tool, URUZ [8], the guideline element model (GEM) ontology-specific GEM CUTTER tool [22], the Guideline Markup Tool (GMT) [34], or the Stepper tool [35]. Semantic structuring can also be performed for general medical documents, using mostly automated methods, as is the objective of the MnM framework [36]. In the case of the DeGeL framework, each marked up CPG is represented using semantic tags from the target guideline ontology that was selected for it by the URUZ user, and is represented eventually as an XML structure based on the structure of the target ontology. Thus, several different instances of the same guideline can exist, each referring to the same source(s), but each of them marked up (structured) using a different guideline ontology. Therefore, Vaidurya context-sensitive queries in DeGeL enable the users to specify within which ontological structure and within which, respectively, specialized knowledge roles within that ontology, should the terms be searched, when performing a context-sensitive search.

## Methods

## The Vaidurya Search and Retrieval Engine

The Vaidurya system is a search and retrieval engine including, concept-based, and context-sensitive search within a hierarchically organized, semantically structured document library, such as DeGeL (and, to a lesser extent, NGC). In this section we present its detailed structure.

A query in Vaidurya is composed of a concept-based component and a context-sensitive component, whose results are weighted according to a predefined formula.

## Concept-Based Search in Vaidurya

The Vaidurya search model includes an optional specification, within a document-retrieval query, of one or more concepts or sub-concepts, using a set of logical operators defining the relations among the concepts. For example, a concept-based query can consist of the chosen list of concepts, "Endocrine Diseases" and "Immunologic Diseases" from the "*Disorders*" axis, and the logic operator between them, is a *conjunctive* (*and*). A query can specify also concepts from other axes; in this case there will be also a logic operator among the axes. An example of a concept-based query is shown at the top in Figure 2.

Our assumption is that a conceptual hierarchy is always composed of one or more concept trees (e.g. Disorders, Therapies), determined by the roots at level 1 of the hierarchy (level 0 being the root concept). For example, the 15 concept trees in MeSH. Thus, medical documents, and in particular, CPGs, are assumed by Vaidurya to be indexed by concepts from a set of conceptual hierarchies, such as the sub-hierarchies of the MeSH super-hierarchy. A CPG can be classified into multiple sub-concepts within the same conceptual hierarchy (e.g., by several different therapy types) and at varying levels of the conceptual hierarchy. This multiple indexing enables users to query for guidelines indexed by multiple concepts, including cases in which the resulting instances are required be indexed by both concepts (i.e., a conjunction).

A concept-based query includes an optional specification of one or more queried concepts or sub-concepts, using the logical operators conjunction (AND) and disjunction (OR), defining the constraints on the desired relations between the queried concepts, explicitly specified by the user.

Informally, a concept-based query includes two operator types. The first, called the *inner operator*, defines a constraint applied among documents within the same concept tree; the second, called the *outer operator*, defines a constraint to be applied among the documents returned after application of the inner operators within each concept trees (see Figure 2).

# <FIGURE 2>

**Figure 2** - The inner operator refers to the relation among concepts within the queried concepts tree; the outer operator refers to the relation between the trees, in the final retrieval stage.

Formally, given $m$ hierarchical concept trees indexing a set of documents, a *concept-based query* $Q^{cb}$ is a structure $\{outer^{op}\ [t^1<inner^1, c^1_1, c^1_2,.., c^1_{n1}>, t^2<inner^2, c^2_1,.., c^2_{n2}>,.., t^k<inner^k, c^k_1,.. c^k_{nk}>], >$, in which the first element defines the *outer* logical operator, and the second element is a list that specifies a collection of $k$ inner queries, $k \le m$, where each inner query $j$ ($j = 1..k$) refers to one of the $m$ concept trees. Each inner query is a set of queried concepts $c^j_i$, in which $j$ is the tree id and $i$ is the queried concept id.

During the retrieval process, first, the documents, classified along each queried concept $c^j_i$ *and its descendents*, are retrieved. (We had considered the option of retrieving also the documents indexed by the parents of the required indexing concepts; however, note that in this case documents that are indexed by a much broader set of concepts will be retrieved; these documents are often likely to be irrelevant for the specified search.) Then, based on the application of the inner operator $inner^j$, a set of documents is retrieved for each queried hierarchy. In the case of an AND inner oprator, the lists of documents indexed by each sub-tree of concept tree $j$ are intersected. In the case of an OR inner operator the lists are unified. Eventually, the outer operator is applied to the $k$ sets of documents retrieved (one for each queried tree), intersecting the sets in the case of an AND outer operator, and unifying them in the case of an OR operator, resulting in the final set of documents retrieved for the CBS. The resultant set is later integrated with additional types of queries, such as a full-text or a context-sensitive search, using a weighted-average formula.

## Context-Sensitive Search in Vaidurya

The context-sensitive search exploits the semi-structuring markup process that a document, such as a CPG, undergoes in DeGeL. To support the context-sensitive Search, we defined two properties that should be defined for each element of the hierarchical contextual model (e.g., a guideline ontology) whose labels were used to tag each segment of the text: *Search Type* and *Search Scope*. These properties, or *aspects*, define the way in which an element will be indexed, queried and retrieved (Table 1,2) and can be considered as a *meta-annotation* of the contextual model. For example, a context query might search for the terms "age greater than 18" within the context element "*eligibility condition*". An example of a context- sensitive query is shown in Figure 3.

### The Search Type

The search type aspect characterizes the type of a contextual element for the purposes of a search engine. An element can have only a single search type based on its contents character. Table 1 describes, for each *Search Type*, its definition and properties.

Table 1 - *Definitions of the possibilities for the search type value of* a contextual element

**<TABLE 1>**

For example, the *free text* search type refers to a chunk of text, such as the text of the section describing the *entry conditions* or the *abstract* of the CPG. The *text value* and *multiple text value* refer to a context that must be a text string, whose value is taken as a single term, which might be of either single or multiple cardinality, respectively. For example, the website from which the CPG was downloaded would be a [single] text value (since there is only one value within that field for each guideline), while the authors' names (which often includes names of multiple authors) is a multiple text value. These [single or multiple] text values are "fixed" in the sense that we are not interested in their textual content, but we would like to retrieve each of them as a single symbol. *Unsearchable* is assigned to a context in the mark up ontology, which is just a place holder that does not have any explicit content, or that cannot be generalized from its descendents; an example is the *Identity* element in the Source ontology, which is a parent node that has the descendents: *Title*, *Guideline Length* and *Release Date*. The .Identity element itself does not relate in any logical way to its descendents' contents (it is only an aggregator for these child nodes) and thus, its content cannot be searched.

**The Search Scope**

Since Vaidurya assumes that documents are represented within different hierarchical contextual models, in which each element may have descendents, both parent and children elements may be targeted when searching. Sometimes there are elements that function as headers and have no content; these elements may sometimes use their descendents' contents when being queried. Using an element's descendents will be relevant only when its descendents are of the same Search Type as the element and when its descendents' contents are semantically relevant and have a retrieval value to the element. Thus, the *Search Scope* aspect has several possible values (Table 2).

It is possible to query an element's descendents and score-up the result to the queried element. Using the *Search Scope* approach allows broadening the query when appropriate, and taking advantage of the descendents' contents, a crucial option within a library such as DeGeL, since

CPGs are often in the process of being marked up, and the elements' contents are not necessarily instantiated--possibly because they have not yet been marked-up.

Table 2 – *Definitions of the possibilities for the search scope value of* a contextual element

# <TABLE 2>

## The Vaidurya Query Interface

The current Vaidurya full-query interface (in this case, as it is used within the DeGeL system), is shown in Figure 3. The left side of the interface contains the optional conceptual axes (e.g., Signs & Symptoms) at the top and the contextual models (the selected CPG markup ontology) at the bottom (e.g., Entry Conditions), which enables the user to query them. The right side of the interface includes the user-formulated concept-based aspect of the user's query at the top, and the user-formulated context-sensitive aspect of the query at the bottom. To formulate the concept-based aspect of the query, the user selects intermediate or leaf concepts, at any semantic level, from the conceptual axes, and defines the *internal* logical relations among the concepts within each conceptual axis (conjunction [*all*] or disjunction [*any*]) and the *external* logical relation among the concepts that belong to different conceptual axes. The context-sensitive query is formulated by choosing one or more ontological elements from the selected CPG ontology, and querying for terms appearing within them, as shown in Figure 3.

# <FIGURE 3>

**Figure 3** – The Vaidurya full-query interface, shown here in the context of the DeGeL guideline library. The query interface enables the user to search for guidelines indexed by concepts from the different concept taxonomies and to query for the presence of certain terms within the context of different guideline-ontology elements. In this case, the user has specified for her concept-based

search three conceptual axes (*Treatment*, *Disorders*, and *Guideline Types*) by all of whom, i.e., an external conjunction (*all*) the guideline should be indexed; within each axis, the guideline should be indexed by one of, i.e., an internal disjunction (*any*) of its concepts. She is going to select a concept from the *Signs & Symptoms* conceptual axis. She has already specified terms that should appear in one of (*any*) of the two guideline-ontology contexts, *filter condition* and *setup condition***.**

The results returned by Vaidurya are displayed by VisiGuide (Figure 4), a viewer which enables users to browse the retrieved documents, including their contents and the classification of the returned results. VisiGuide can display the results using any given hierarchical contextual model.

**<FIGURE 4>**

**Figure 4** - The VisiGuide interface for displaying the results of searching for an answer to an information need, in this case, within the DeGeL library. Here, VisiGuide displays several CPGs returned as a response to a query in Vaidurya. The set of relevant CPGs can be viewed and even browsed individually in detail according to the contents of their ontology-specific knowledge elements (e.g., Title or Entry Condition), and according to their concept-based classifications (e.g., those relevant to a particular Symptom or Sign). The concept hierarchy on the left displays how many CPGs relevant to the query were found for each concept class or sub-class in the concept hierarchy, at any desired level.

## The Vaidurya Indexing and Ranking Algorithm

Guidelines are indexed in Vaidurya by their document ID, contextual (e.g., ontological model) element ID, and their content. (Note that elements are labeled text segments that have been marked up previously, such as, for example, by the semi-structuring phase in the case of the DeGeL library, or by editors in the case of the NGC library). Each element's content is indexed according to its predefined *search type*. A query in Vaidurya is converted into a collection of queried elements grouped by their *search type*. Each document is ranked according to the query, representing its relevancy to the query. In this section we describe the ranking algorithm.

## Indexing Free Text

Free text queries are ranked based on the traditional approach in text retrieval. In Vaidurya, each free text content element is indexed by the terms appearing in that element. 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* [37] down to their linguistic root. Each term gets a *term frequency-inverse document frequency* (*tfidf*) value representing a combination of the term frequency in the document and in the entire collection. Equation 1 shows how the term frequency of a term is calculated. The *tf* value represents the relative frequency of a term in a document or a text segment by normalizing its frequency relative to the one of the most frequent term(s) in the document or text segment. Equation 2 shows how the *tfidf* value is calculated based on the *tf* value, *N*, the number of items in the entire collection, and *n,* the number of items in which the term appears. As can be seen, the *tfidf* score is high when the term *i* is very frequent in the document and infrequent over the document collection as a whole.

$$tf_i = \frac{N(term_i)}{\max(N(term))} \qquad (1)$$

$$tfidf_i = tf_i \cdot \log(N/n) \qquad (2)$$

**Similarity Functions**

The ranking process is based on similarity functions, which measure the relevance of each document element's content to the queried element. Similarity functions differ for each search type. All of the similarity functions are Boolean, which means that the document content is either relevant or not. The exception is the FreeText search type, whose relevance measure is a continuous value [0,1]. Most of the search types' similarity functions behave as selection operators in the database Standard Query Language (SQL).

Concept-based queries include several concepts with boolean operators between them, defining a subgroup of documents classified along the combination of concepts, as specified in the query. The similarity function in this search type returns 1 when the document appears in this group and 0 when it does not.

When a query is executed, the relevance of each of the elements defined by the query is computed. This is done using the *Cosine similarity function.* Equation 3 shows how the similarity is computed using the Cosine function, which computes the similarity between the query (i.e., the terms listed for each query element), where each term in the query is $q_i$, and the document (i.e., the terms listed in each document element), where each corresponding term is $D.elem_i$, where $D$ is a document, $elem$ is a queried context element in the document and $i$ is a term in the element query $q$.

$$Sim(C.elem, q) = \frac{\sum_{i=1}^{N} (C.elem_i \cdot q_i)}{\sqrt{\sum_{i=1}^{N} C.elem_i^2 \cdot \sum_{i=1}^{N} q_i^2}} \qquad (3)$$

**Ranking Weights**

In order to implement the ranking algorithm in Vaidurya, where a query is a collection of different contextual (ontology) elements of different search types, two kinds of weights were defined. A *Search Type Weight* (STW) defines the relative *search type weight*. We assume that each search type has differing contribution and reliability factors relative to the retrieval task. For example, in a TextValue search type element the user chooses from a given finite list, which we consider as very reliable in terms of representing the user needs. In the FreeText search type, the user may not enter the correct keywords, and, moreover, the document may contain synonyms. In FreeText the user has to provide the keywords from his own mind, and thus we consider this search type querying to be sometimes less effective. Thus, FreeText will have a relatively lower weight than TextValue. The second weight is the *Ontology Element Weight* (OEW); this weight represents the relative contribution and reliability of a context (ontology) element. For example, elements containing a larger segment of text have a better presentation of the term frequency than those containing a smaller segment 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 bigger segment of text, and as a result will have a heavier weight. There are also other considerations in determining the relative weight of each ontology element, such as how clear is the meaning of the element to any user. Weights are acquired from a domain expert or are based on empirical experiments and, in any case, are set within Vaidurya.

### The Ranking Algorithm

Each query $Q$ can be viewed as a collection of queries referring to various elements of the document ontology (including the indexing concepts). Thus, $Q = \{st_1 = \{q_{elem\_1}, q_{elem\_2}\}, st_2 = \{q_{elem\_3}\},.., st_k = \{q_{elem\_n}\}\}$, where $q_{elem}i$, is a query on the ontology element whose id is *i,* and where the elements are grouped by their *search type* $st_j$. Each ontology element *i* has a weight $OEW_{elem}i$ and each search type has a weight *STWj*. 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 each *search type* (*ST*) in query *Q* is calculated. Equation 5 shows how the final rank of a document is evaluated based on the search types ranks. Note that currently, |Q| = *k* = 3 for the three search types, but in principle, new search types can be added in the future.

$$Rank(STj) = \sum_{i=1}^{|ST_{jj}|} OEW_i \cdot Sim(q_i, D.elem_i) \qquad (4)$$

$$Rank(D,Q) = \sum_{j=1}^{|Q|} STW_j \cdot Rank(ST_j) \qquad (5)$$

## Evaluation of the Vaidurya Search Engine

An initial *feasibility* evaluation was performed by implementing the Vaidurya engine and its interfaces. The Vaidurya engine was then incorporated within the DeGeL architecture [8]. We also asked our DeGeL clinical users for qualitative feedback regarding its usability and functionality in that context.

A preliminary *functional* evaluation of Vaidurya was performed using part of the TREC[4] 6 collection, which determined that Vaidurya's baseline full-text retrieval performance was reasonable, compared to other known retrieval systems.

A description of the full details of the evaluation and the complete results is outside the scope of this paper, which focuses on the Vaidurya search engine and its architecture, and is described elsewhere [38,10]. We shall only provide a brief sketch of the evaluation methodology and of its core results; when presenting the results, we shall integrate these results

[4] Text Retrieval Conference (TREC), http://trec.nist.gov

with several more recent results that were briefly introduced elsewhere [11], which have extended the use of the inner and outer operators to all four possible combinations.

The DeGeL library is not yet a sufficiently large CPG repository. Thus, to fully and quantitatively evaluate the contribution of the several search types implemented within the Vaidurya search engine, we created a test collection based on the NGC textual CPGs collection, which is a particularly appropriate collection for evaluation of Vaidurya, since each CPG is classified along several concepts, and is internally structured. For the evaluation of the Vaidurya system's performance within the NGC collection, we used the NGC repository's internal structure, which is designed for easy reading and browsing. This internal structure includes: Clinical Specialty, Guideline Category, Implementation Tools, Intended Users, IOM Care Need, IOM Domain, Method of Guideline Validation, Methods Used to Analyze the Evidence, Methods Used to Assess the Quality and Strength of the Evidence, Methods Used to Collect/Select the Evidence, Methods Used to Formulate the Recommendations, Organization Type, Target Population, Treatment/Intervention.

The NGC website repository contained at the time the evaluation was performed 1136 CPGs classified along the multitude MeSH and UMLS hierarchical concepts: Disease/Conditions and Treatment/Intervention. Each concept tree has roughly 2,000 unique concepts in a tree-like structure.

To create a test collection from the NGC collection five physicians defined thirteen information needs, queries and the corresponding judgments. Judgments are the CPGs expected to be retrieved upon a specific query. To evaluate the three search methods, each query had three representations: *full-text query* (*FTQ*), *concept-based query* (*CBQ*), and *context-sensitive query* (*CSQ*). We performed a detailed formal evaluation of the Vaidurya search engine, comparing the results of performing full-text search, concept-based search, and context-sensitive search.

To measure the retrieval performance, we used the traditional precision and recall metrics. Precision is the proportion of relevant documents (defined for a specific query within the entire collection, also called *judgments*) within the set of the retrieved documents, and recall is the proportion of relevant documents (judgments) retrieved from the set of relevant documents (judgments), for a specific query. We also used eleven-point average precision, which is a common measurement of the combination of precision and recall. Eleven-point average precision is computed by averaging the precision, or interpolated, of the retrieved documents at the recall levels 0.0, 0.1,..,1.0.

## Experiments and Results

### Results of the Vaidurya Search-Engine Evaluation

With respect to the *pragmatic* evaluation, the Vaidurya engine software design and implementation were successful, and the preliminary tests within the DeGeL architecture have demonstrated the engine's usability and functionality. The Vaidurya system is currently routinely used within the DeGeL project to support modules such as the Uruz markup module [8] and the Spock runtime application engine [39]. Thus, the pragmatic evaluation showed the system to be successful.

Even at that early phase, we received useful feedback from DeGeL users that included comments regarding two key potential obstacles: (1) there was some difficulty in the manual concept-based indexing of guidelines within the IndexiGuide guideline-classification tool; and (2) the full Vaidurya interface, including all types of searches, although providing considerable functionality is too complex. We briefly mention our proposed solutions to these problems and some specific results already achieved with respect to solving them, in the Discussion section.

Before proceeding with the functional evaluation, we performed a set of evaluation runs on the TREC-6 collection to assess just the full-text search and to verify that its performance is not too poor; the results were comparable with other search engines. In addition, to tune the search engine, we ran a preliminary evaluation with several queries on the NGC collection, changing the relative weights of the free-text, concept-based and context-sensitive search types (Equation 5), which had demonstrated little dependence on the particular weight values; thus, we left the weights at their default values, 1/3, 1/3, 1/3.

Figure 5 shows a summary of the performance achieved in the first evaluation [10] for each search method. It presents a graph of precision-recall curves of the retrieval performance achieved for the three different search methods implemented within the Vaidurya engine, namely *full-text*, *context-sensitive*, and *concept-based queries* (FTQ, CSQ, CBQ respectively). The first method is an FTQ search, which was considered to be the baseline. The best performance achieved when adding a CSQ search to the FTQ was when the query included two context elements (FTQ+2CSQ). The best performance achieved when adding a CBQ search to a FTQ was when querying for concepts selected by the user from the third level of the conceptual hierarchy and when the disjunction logical operator was used (FTQ+CBQ3OR). When only a CSQ search was used, the allover best performance was achieved when using three context elements (3CSQ). The 3CSQ also achieved the best retrieval performance at low levels of recall. However, using the concept based queries at the third level, with a disjunction operator, in addition to the full text (FTQ+CBQ3OR) outperformed the other methods at the 0.5 level of recall. Note that, for most recall levels, all of the concept-based search and context-sensitive search methods outperformed the traditional full-text search.

To compare the methods' performance we chose the mean precision at the 0.5 recall level and compared the different means using t-tests, a procedure often recommended for evaluation of information retrieval systems [40]. Typically, the difference in the precision among different search methods at the low recall level was high (and statistically significant; for all methods versus FTQ), while at the high recall levels it was low; thus, we decided to choose the 0.5 level of recall.

At that level of recall the trend was clear, but nonsignificant. The 95% confidence intervals for precision at the 0.5 level of recall were: FTQ -- [0.28, 0.42], FTQ+3AND -- [0.28, 0.50], 3CSQ -- [0.30, 0.53] and FTQ&2CSQ -- [0.41, 0.53].

**<FIGURE 5>**

**Figure 5** – A summary of the results of the Vaidurya search-engine evaluation in its three search modes. Precision and recall results are displayed, summarizing several key combinations of search types. The first curve ("FTQ") represents the full text search. The second curve ("FTQ+2CSQ") represents the mean performance achieved when a context-sensitive query (CSQ) including two context elements was added in addition to the full-text query (FTQ), as described in the algorithm, (see Section III.A.5.). The third curve ("FTQ+CB3OR") represents the mean performance when a concept-based query (CBQ) was used, with concepts selected by the user from the third level of the concepts hierarchy, using a disjunction logical operator, in addition to the full-text. The fourth curve ("3CSQ") represents the mean performance achieved for a CSQ including three context elements (without an FTQ). The results indicate that using a CSQ or a CBQ improved the retrieval performance compared to the traditional full text search, especially when a CBQ is used in conjunction with the FTQ.

## Full Concept-Based Search Evaluation

In addition to the initial evaluation, we have recently performed an evaluation focusing on the concept-based search [9]. Unlike in the case of the first set of experiments described above, in which, within the concept based search, the *outer operator* was always set to OR, in this study we performed a broader evaluation of the concept-based search, including all four possible combinations of the two operators, the *inner* and the *outer*, each set to be either OR or AND. In the new study, we wanted to examine the contribution of the CBQ to the FTQ, CSQ and 3CSQ baselines separately. Additionally, we wanted to investigate in detail the best operator (*inner* and *outer*) settings and the concept query level that optimizes the contribution of the concept based search for each baseline.

Three main experiments were designed, in which we used CBQ in addition to a given baseline: (1) FTQ, (2) single CSQ, and (3) three CSQs. In each experiment we evaluated eight combinations of the CBQ resulting from three variables: (1) the queried level of the hierarchy second or third, (2) the outer logic operator having AND or OR, and (3) the inner logic operator having AND or OR, resulting in the four options: OR-OR, OR-AND, AND-OR and AND-AND. Table 3 presents all the eight combinations and the acronyms by which we refer to the operator and level combinations when reporting the results.

Table 3 – The eight concept based queries combinations.

**<TABLE 3>**

Figure 6 presents a summary of the mean retrieval performance, in a precision-recall curves graph. For each baseline we present the best added CBQ option.

In general, a significant improvement was achieved when a CBQ was used in addition to an FTQ. The best CBQ setting when querying in addition to an FTQ was CBQ-3OO, that is, querying at the third level of the concept hierarchy and having a disjunction relation in both the inner and the outer operators.

When a CBQ was used in addition to a single CSQ (1CSQ), the best mean performance was achieved for CBQ-3OA, which means querying at the third level with disjunction in the outer operator and conjunction in the inner operator. As in the case of FTQ, here a significant improvement was observed too. Querying CBQ in addition to three CSQs decreased the mean performance of the baseline for all settings. However, the best settings were for querying the third level with disjunction in the inner and the outer operators (3CSQ+3OO). Note that the 3CSQ baseline was higher than the FTQ and 1CSQ baselines. It can be seen that as the baseline performance is higher the CBQ contribution is lower. Additionally, when the baseline is low, it is better to use conjunction in the operators; and when it is higher, it is better to use disjunction, especially in the outer operator.

**<FIGURE 6>**

**Figure 6.** The best concept-based query (CBQ) settings are presented in addition to three baselines: A free-text query (FTQ), a one-context context-sensitive query (1CSQ) and a three-contexts CSQ (3CSQ). The dashed lines represent the respective baselines, while the full line represents the performance with the additional CBQ. Among the baselines, the 3CSQ (though not the 1CSQ) outperforms the FTQ. The CBQ enhances the mean retrieval performance for the FTQ and 1CSQ baseline, and decreases it in the case of the 3CSQ baseline, which starts at a very high level. As the baseline is lower, the contribution of the CBQ is higher.

## Discussion

We have demonstrated the feasibility of designing and implementing a functional full-text, concept-based, and context-sensitive search engine for clinical, semi structured documents that are classified into several levels of several conceptual hierarchies. In the case of the concept-based search, the queries use concepts from multiple hierarchies; the documents classified to these concepts and to their ontological descendents are retrieved based on internal and external logical operators defined by the query among the concepts. Furthermore, we have demonstrated the potential value of concept-based and context-sensitive search in addition to full-text search, at least within a relatively significant-size electronic clinical-guideline repository, and especially when base-line precision is low.

Although the described search-and-retrieval framework is a general one, we evaluated the approach on a collection of textual CPGs, whose effective search and retrieval is one of our main objectives, as explained in the introduction.

Indexing the documents along multiple hierarchical semantic axes, such as is done by the NGC *Disorders* and *Therapies* axes, or by the seven indexing hierarchies currently implemented in the DeGeL framework, supports a concept-based search, in which the user explicitly indicates in the query multiple concepts from one or more semantic axes.

The markup process that a CPG goes through, for example within the DeGeL architecture, using the URUZ hybrid (multiple-format) mark-up and specification tool for the gradual conversion of the CPG into an executable representation; or as part of the segmentation manifested through the XML representation format within the NGC repository, presents us, besides the capability to use *concept-based* search, with an exceptional opportunity to query a digital guideline library using *context-sensitive* methods. Further opportunities will undoubtedly present themselves due to the recent work on automated semantic markup of documents, in particular clinical ones, as mentioned earlier. In the case of Vaidurya, the context-sensitive search is based on a contextual model created by the mark-up process, i.e., by labeling the text using a particular target guideline ontology.

Combining several types of search based on weighting schemas, such as in text retrieval, and Boolean chemas in the case of conceptual retrieval were used also in Text Categorization [41], knowledge based text retrieval [42, 43], question-answering [44]. Our use in this study of concept-based search is somewhat different from previous work, which has not often demonstrated an improvement; this difference in method might explain the enhancement in results achieved here. Concept-based indexing in the medical domain often uses concepts that are extracted automatically from the text contents, usually referring to the MeSH conceptual hierarchy. Similarly, the user's query (also a document) is converted automatically into a concept query, and matched against all the indexed documents, as in the case of the PubMed tool. In contrast, the concept-based queries in Vaidurya are specified explicitly by the user. In addition, the libraries we have applied it to (DeGeL and NGC) use mostly manual indexing. Our experimental results show that the concept-based search improved the retrieval performance when used in addition to full-text and context sensitive queries [10,11].

One of the limitations of the current study is that we had performed before the main experiments a preliminary experimentation with multiple different settings, using an automated test and a small set of judgments; changing the relative weights of the three search types did not seem to change the performance, so the weights were left as uniform. However, future experiments with multiple additional queries might be needed to better validate these preliminary default settings.

During the performance of our study, there were two potential obstacles to the use of concept-based and context-sensitive search that were mentioned by the DeGeL and Vaidurya users; we have since then attempted to address both. The first obstacle mentioned in our preliminary experiments was that the CBQ and CSQ search interface, although powerful, was highly overloaded from the point of view of a casual clinical user. It displays both the semantic axes used by the concept-based search and the target ontology's semantic roles, which are used by the context-sensitive search. To overcome this potential problem, we developed and presented elsewhere new interfaces that are simpler and that we expect to be significantly more user-friendly. These interfaces offer several versions of the search screen, each at a different sophistication level, all customizable by the user [45]. In addition to these customizable interfaces, which mainly select contexts from the selected document's contextual model, we have also developed a query interface in which clinicians assisted by a medical informatics expert can create query templates for end users, who are not familiar with the underlying context model, and possibly not even with the specific medical domain in which they wish to search [46]. A second potential obstacle to the use of concept-based search, mentioned by IndexiGuide users within the DeGeL library, was the need for the manual classification of each new document added to the digital library (unless originally classified by the author using the same semantic axes; even in that case, additional indices are often useful for the Vaiduyra engine purposes), potentially a time-consuming task even for an expert mark-up tool editor. Non-satisfactory indexing may conceivably harm the precision of the concept-based retrieval method. Thus, we have developed an automatic classifier that classifies new guidelines along multiple semantic hierarchical axes, based on a given set of guidelines previously classified using the same axes. The guideline classifier thus proposes to the human editor, given a new guideline, a set of paths (i.e., both intermediate and leaf nodes) in the concept hierarchy, by which the new guideline might be indexed, thus significantly facilitating the task of multiple-hierarchical indexing of the new guideline. We have rigorously evaluated the new automated-classification approach within the NGC framework, with encouraging results [47].

Our conclusions are thus that the Vaidurya engine's concept-based and context-sensitive search and retrieval methods are both usable and useful, and that the potential obstacles to their effective use are likely to be overcome with appropriate user friendly interface designs and by automated support to the process of concept-based classification.

## Acknowledgements

This research was supported in part by NIH award No. LM-06806. We thank S. Tu and Dr. Peleg,, for useful discussions regarding the need for supporting the use of multiple guideline ontologies. Drs. Shiffman and Karras assisted us in using their GEM ontology. Drs. Goldstein, Basso, Kaizer, Advani, and Lunenfeld, were extremely helpful in assessing the various interfaces and search. We thank all the graduate students in the Medical Informatics Lab, Especially A. Hessing and O. Young, in their contribution to the creation of Vaidurya's interfaces. We thank A. Litmanovitz, O. Bohanna, & C. Sasson, who worked on the IR and HCI evaluation tools; S. Nave, L. Bar-on, A. Efrati and S. Hatan, and O. Perets, and Y. Schlamm, who developed the new framework of customized and template based query interfaces; and I. Dahari, D. Pretselman, and U. Gobi, who developed the CPGs web wrappers. We wish to thank Dr Kumar for his contribution in the extraction rules and the assessment of the query interfaces. We also thank T. Lavie for her contribution in the HCI aspects of the query interfaces and Dr. Leibowitz for his contribution in designing and assessing several template-based query interfaces.

## References

[1] Benjamin G. Druss, Steven C. Marcus, Growth and decentralization of the medical literature: implications for evidence-based medicine, Journal of Medical Library Association, 93(4): 499–501, 2005.

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

[3] Bodenreider O, The Unified Medical Language System (UMLS): integrating biomedical terminology, Nucleic Acids Research, 2004, Vol. 32, 267-270.

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

[5] Aronson A, Demner-Fushman D, Humphrey S, Lin J, Liu H, Ruch P, Ruiz M, Smith L, Tanabe L, Wilbur J, (2005) Fusion of knowledge-intensive and statistical approaches for retrieving and annotating textual genomics documents. Proceedings of TREC 2005, Gaithersburg, MD, USA.

[6] 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.

[7] Moskovitch R., Hessing A. and Shahar Y., Vaidurya - A Concept-Based, Context-Sensitive Search Engine For Clinical Guidelines , Proceedings of MedInfo 2004, San Fransico, USA.

[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] Sa'adon R, Moskovitch R, Shahar Y, Hierarchical Expansion for Concept-Based Search, Medinfo 2007, Brisbane, Australia, August 2007.

[10] Robert Moskovitch, Susana B. Martins, Eytan Behiri, Aviram Weiss, and Yuval Shahar, A Comparative Evaluation of a Full-text, Concept Based, and Context Sensitive Search Engine, Journal Of American Medical Informatics Association, 14: 164-174, March 2007.

[11] Moskovitch R., Saadon R., Behiri E., Martins E., Weiss A., Shahar Y., Experiments with Hierarchical Concept-Based Search, Medinfo 2007, Brisbane, Australia, 2007.

[12] Field M and Lohr K, Clinical practice guidelines: directions for a new program, ed. I.o.M. Committee to Advise the Public Health Service on Clinical Practice Guidelines. 1992.

[13] 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.

[14] Grimshaw JM and Russel IT. Effect of clinical guidelines on medical practice: A systematic review of rigorous evaluations. Lancet 1993;342:1317–1322.

[15] Qualigini S, Ciccarese P, Micieli G, and Cavallini A. Non-compliance with guidelines: motivations and consequences in a case study. Proceedings of the symposium on Computerized Guidelines and Protocols (CGP 2004), 2004: p. 75-87.

[16] Micieli G, Cavallini A, and SQ. Guideline compliance improves stroke outcome - A preliminary study in 4 districts in the Italian region of Lombardia. Stroke 2002;33:1341-1347.

[17] Tu, SW, Kahn MG, Musen MA, Ferguson JC, Shortliffe EH and Fagan LM Episodic Skeletal-plan refinement on temporal data. *Communications of ACM* 1989; 32:1439–1455.

[18] Musen MA, Tu SW, Das AK, and Shahar Y. EON: A component-based approach to automation of protocol-directed therapy. *Journal of the American Medical Information Association* 1996;*3*(6):367–388.

[19] 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.

[20] Fox J, Johns N, and Rahmanzadeh A. Disseminating medical Knowledge: the PRO-forma approach. Artificial Intelligence in Medicine 1998;14:157-181.

[21] Peleg M, Boxwala AA, Tu S. Zeng Q, et al. The InterMed approach to sharable computer-interpretable guidelines: A review. J. Am. Med. Inform. Assoc. 2004;11(1):1-10.

[22] 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]

[23] Qualigini S, Ciccarese P, Micieli G, and Cavallini A. Non-compliance with guidelines: motivations and consequences in a case study. Proceedings of the symposium on Computerized Guidelines and Protocols (CGP 2004), 2004: p. 75-87.

[24] Singhal A, Modern Information Retrieval: A Brief Overview. IEEE Data Eng. Bull. 24(4): 35-43 (2001)

[25] Savoy J, Bibliographic database access using free-text and controlled vocabulary: an evaluation. Inf. Process. Manage. 41(4): 873-890 (2005)

[26] Gobeill J, Muller H, Ruch P, Translation by Text Categorization, *Proceedings of ImageCLEF*, 2006.

[27] 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.

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

[29] Hersh WR, P.S., Donohoe L,. Assessing thesaurus-based query expansion using the UMLS Metathesaurus. in 2000 Annual AMIA Fall Symposium. 2000.

[30] Srivansan.P, Retrieval feedback in MEDLINE. Journal of American Medical Informatics Association, 1996. 3(2): p. 157-167.

[31] Rada, R., Bicknell, E., Ranking documents with a thesaurus. Journal of the American Society for Information Science, 1989. 40: p. 304-310.

[32] Purcell G P, 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.

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

[34] Votruba P, Structured knowledge acquisition for Asbru. Master's Thesis, Institute of Software Technology and Interactive Systems, Vienna University of Technology, Vienna, Austria, 2003

[35] Svatek V and Ruzicka M. Step-by-step mark-up of medical guideline documents. In: Medical Informatics Europe, Budapest. IOS Press, 2002

[36] Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. and Ciravegna, F. MnM: Ontology Driven Tool for Semantic Markup, ECAI Workshop on Semantic Authoring, Annotation & Knowledge Markup (SAAKM 2002). Lyon, France, 2002.

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

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

[39] Young, O. and Y. Shahar (2005). Applying Hybrid-Asbru Clinical Guidelines Using the Spock System. Proceedings of the 2005 AMIA, Washington, USA

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

[41] Ruch P, Automatic Assignment of Biomedical Categories: Toward a Generic Approach. Bioinformatics, 22(6):658-64, 2006.

[42] Lin J, Demner-Fushman D, The role of knowledge in conceptual retrieval: a study in the domain of clinical medicine. SIGIR 2006: 99-106

[43] Huang X, Lin J, Demner-Fushman D. Evaluation of PICO as a Knowledge Representation for Clinical Questions. Proceeding of the 2006 Annual Symposium of the American Medical Informatics Association (AMIA 2006).

[44] Demner-Fushman D, Lin J, Answering Clinical Questions with Knowledge-Based and Statistical Techniques. Computational Linguistics 33(1): 63-103 (2007)

[45] Moskovitch R and Shahar Y. A multiple ontology customizable search interface for retrieval of clinical guidelines, Clinical Guidelines, Symposium on Computerized Guidelines and Protocols (CGP-2004), Prague.

[46] R. Moskovitch, T. Lavie, A. Leibowitz, Y. Denekump, Y. Shahar, A Multiple-Ontology Template-Based Query Interface for a Clinical-Guidelines Search Engine, Proceedings

of ECAI06 workshop on AI techniques in healthcare: evidence-based guidelines and protocols, Italy, 2006.

[47] 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 (2006) 37, 177-190.