Experiments with Hierarchical Concept-Based Search

Robert Moskovitch[a], Roee Sa'adon[a], Eytan Behiri[b], Susana Martins[c], Aviram Weiss[d], and Yuval Shahar[a]

[a]*Medical Informatics Research Center, Ben Gurion University, Beer Sheva, Israel*
[b]*E&C Medical Intelligence Inc., 41 Madison Ave, New York, NY 10010*
[c]*Medical Corps, Israel Defense Forces, Israel*
[d]*Stanford University, Stanford, VA Palo Alto Health Care Center, Palo Alto, CA, USA*

Abstract

Many digital libraries use hierarchical indexing schema, such as MeSH to enable concept based search in the retrieval phase. However, improving or outperforming the traditional full text search isn't trivial. We present an extensive set of experiments using a hierarchical concept based search retrieval method, applied in addition to several baselines, within the Vaidruya search and retrieval framework. Concept Based Search applied in addition to a low baseline is outperforming significantly, especially when queried on concepts in the third level and using disjunction within the hierarchical trees.

Keywords:

Medical Text Retrieval, Concept Based Search.

Introduction

Many digital libraries are indexed using a hierarchical conceptual structure; examples include PUBMED, in which documents are classified along the Medical Subject Headings (MeSH) concepts, and the National Guideline Clearinghouse (NGC[1]) library, each of whose documents is classified using multiple concepts from MeSH and the Unified Medical Language System (UMLS) [1]. Several sites allow browsing through the concepts using the hierarchical structure from the root to the most specific concepts (leaves), which forces the user to navigate the hierarchy. Others enable to query for concepts from MeSH relying on the pre-indexing of the documents along the concepts. Some studies proposed limiting the search to a specific concept (category) [2, 3] and its subconcept contents. In the medical domain, unlike the web, 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.

The NIH had invested huge amount of money during the past decades in building a set of controlled vocabularies and accessory tools to enable the implementation of concept indexing and retrieval. However, no study had shown that using the conceptual structures outperforms or improves the traditional full text search. One of the famous studies was made by Hersh, in which an attempt to adopt Salton's *tfidf* approach to the conceptual resulted unsuccessfully [4]. Recently we presented *Vaidurya*, a concept based and context sensitive search engine, developed originally, within the Digital Electronic GuidEline Library (DeGeL) [5], to search for textual and marked up clinical practice guidelines, however, we extended it recently to handle general clinical documents. A detailed description of Vaidurya is provided in [6], as well as an extensive and rigorous evaluation, in which a small portion of this study results appear, however, in this study a wider evaluation is provided, in which both concepts based logic operators are evaluated as we will elaborate later. Vaidurya enables the user to query explicitly for concepts given a logic relation between them. In this study we present a novel hierarchical concept based retrieval method including a wide and detailed evaluation of the approach.

We start with a background review of concept based search (CBS) and MeSH. Then we describe briefly the search methods implemented in Vaidurya. We describe our research hypotheses, the experimental plan, and the results. Eventually we discuss the results and conclude.

Background

Concept Based Search

In the medical domain, CBS refers to a text retrieval approach, in which documents are mapped to concepts, representing a meaningful abstract subject, based on its contents. Hersh's SAPHIRE system [4] uses an approach, in which concepts used for indexing, are automatically extracted from the document. Commonly both documents and queries are mapped, in the case of the biomedical domain into vocabularies such as MeSH and UMLS. However, users are not always familiar with the concepts in these vocabularies, rendering it somewhat limited. It has been previously noted, that the particular implementation of concept-based search described above does not necessarily improve the retrieval performance [7] compared to traditional text retrieval methods. Other studies have tried to exploit the UMLS meta-thesaurus to expand queries, thus extracting the concepts from the query terms [7]. The authors found that query expansion degraded aggregate retrieval performance, but some specific instances of synonyms

[1] www.ngc.org

and hierarchy-based query expansion improved individual query performance. Aronson [8] compared his methods to Srinivasan's [9] 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 [10] 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.

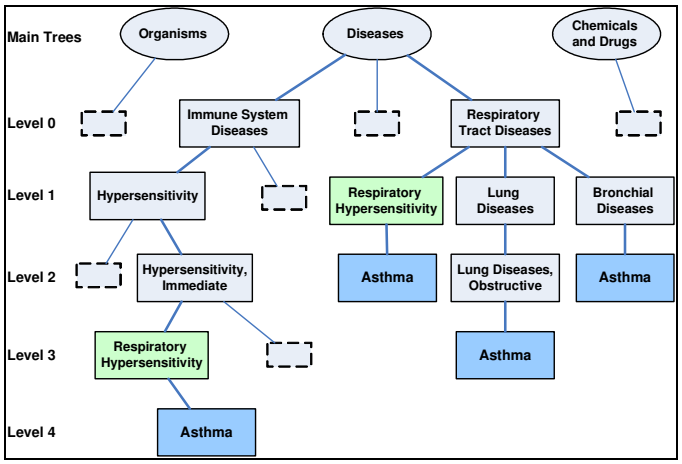

Figure 1 – The concept "Asthma" in MeSH, decomposed to tree-numbers, appearing in several locations in different concept trees.

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).

Vaidurya

Vaidurya is a concept based and context sensitive search engine, developed originally, within the Digital Electronic GuidEline Library (DeGeL) [5], to search for textual and marked up clinical practice guidelines, including three types of implemented search methods: (1) full-text search, using standard key terms; (2) context-sensitive search, which exploits the semantic markup performed on guidelines in *DeGeL*, can be used also for search in a structured document; and (3) concept-based search, consisting on hierarchical concepts indexing structure. The documents can be classified manually or automatically using machine learning based method as we proposed in [11].

Full Text Search in Vaidurya

Documents are represented and indexed using the *vector space model* introduced by Salton [12], commonly used in free text search, in which each document is represented by a *bag-of-words*. After the document terms are extracted, stop-words (e.g., "and", "the", "are") are removed and the left terms are stemmed to their root using porter algorithm,, a vector of terms is created, such that each index in the vector represents a term frequency in the document, known as ***term-frequency-inverse-document-frequency*** (**TF*IDF**). Term frequency (*tf*) represents the term appearances in the specific document, normalized by the most appeared term (tf^{max}), while the *idf* = *log(N/n)* represents the appearance of the term in the entire documents collection, where *N* is the size of the collection, and *n* is the number of documents in which the term appears. The free text retrieval is based on the *cosine similarity* [12] which measures the distance between a query and a document within the Euclidean space of the terms. More details about the free text retrieval in vaidurya is at [6].

Context Sensitive Search in Vaidurya

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). 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. Thus a contextual query includes a set of keywords for each queried context element. In the retrieval process each document gets a rank according to the match for each queried context. This study focuses on the concept based search in Vaidurya, however, more details are provided at [6].

Materials and Methods

Concept Based Search in Vaidurya

In general, our study focused on a very broad class of search methods, which we refer to as *double-operator* methods. Our assumption is that a conceptual hierarchy is always composed of one or more *concept trees*, determined by the roots at level 1 of the hierarchy (level 0 being the root concept). For example, the 15 concepts trees in MeSH.

To perform a concept-based search, Vaidurya includes optional specification of one or more concepts or subconcepts, 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. The first, called *outer-op*, defines the relation among *different* concept trees; the second, called *inner-op*, is defined within the *same* concept tree (Figure 2). Formally, a concept based query Q^{cb} = $\{[t^1\langle inner^1, c^1_1, c^1_2,.., c^1_{n1}\rangle, t^2\langle inner^2, c^2_1,.., c^2_{n2}\rangle,..,t^m\langle inner^m, c^m_1,.. c^m_{nm}\rangle], outer^{op}\rangle$ is denoted by a pair, in which the first element specifies a collection of the queried concepts trees, $T = \{t^1, t^2,.., t^m\}$, in each one of the *m* concept trees t^1 to t^m. Each queried tree t^i is defined by a set of queried concepts c^i_k, in which *i* is the tree id and *k* is the queried concept id, and a

local $inner^j$ operator AND or OR. The second element defines the *outer* logic operator.

During the retrieval process, first, the documents, classified along each queried concept c^i_j and *its descendents*, are retrieved. Then, based on the application of *inner-op*, a set of documents are retrieved for each queried hierarchy, in the case of AND the documents of each concept are intersected and in the case of OR they are unified. Eventually, the application of the *outer-op* logical operator on the documents retrieved for each tree, intersecting in the case of AND, and unifying in the case of OR, results in the final set of documents retrieved for the CBS having the same rank, which is later integrated with additional types of queries, such as full text or context sensitive search, using a weighted average formula.

Four concept based retrieval algorithms were used in this study, defined by the *outer* and *inner* logical operators: (1) *OR-OR*, in which both logic operators set to disjunction. (2), OR-AND, in which the outer is set to disjunction and the inner to conjunction (3) *AND-OR*, in which the outer operator is set to conjunction and the outer operator is set to disjunction, and (4) *AND-AND*, in which both logic operators set to conjunction. In this study the inner operator in all the trees were set to the same value.

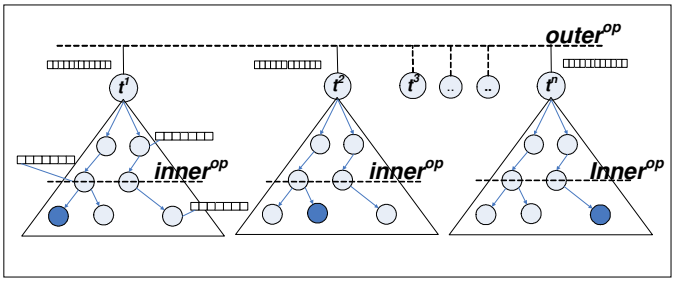

Figure 2- The inner-op is located within the queried concepts tree and the outer-op defines the relation between the trees, in the final retrieval stage.

Evaluation

Test Collection

For our detailed evaluation of the relative value of concept-based and context-sensitive search required a test collection in which documents are structured and classified along hierarchical concepts, we have therefore used the 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.

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 Company and a physician from the Medical Crops of the Israeli Defense Forces 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. Eventually, these physicians scanned (*manually*) 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.

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. 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. For the CBQ two types of queries were formulated: (1) concepts from the 2nd level of the conceptual hierarchy, and (2) concepts from the 3rd level of the conceptual hierarchy. In this study we queried the CBQ using the four combinations of the *outer* and *inner* logic operators.

Evaluation Measures

In order to evaluate 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-retrieved documents (judgments) retrieved from the set of relevant documents (total judgments), for a specific query. We also interpolated the averaged precision at eleven-points of recall levels 0.0, 0.1,..,1.0.

Experimental Plan

In this study we wanted to examine the contribution of the CBQ to the FTQ and CSQ. We also wanted to examine the best level of querying, as well as estimating the best logic operators settings.

Hypothesis I – *adding concept-based search to full-text or context-sensitive search will improve the respective baseline performance*.

Hypothesis II – *Querying concepts from the third level will outperform querying at the second level.*

Hypothesis III – *There are significant differences among the results of the four types of searches*.

To examine these hypotheses three main experiments were designed, in which we used CBQ in addition to a given base-

line: (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 1 presents all the eight combinations including an acronym which we will use in the report of the results.

Table 1 – The eight concept based queries combinations.

Level	Outer	inner	Acronym
2	OR	OR	2OO
2	OR	AND	2OA
2	AND	OR	2AO
2	AND	AND	2AA
3	OR	OR	3OO
3	OR	AND	3OA
3	AND	OR	3AO
3	AND	AND	3AA

Experiments and Results

We present the results of three experiments, in which the eight CBQs were applied in addition to a given baseline. As a result of the limited length of the paper we present only the four best CBQs in the figures, while report the order of the others performance in the text. We sorted the CBQs according to their average precision at 0, 0.5 and 1 recall level.

Experiment 1 - CBQ in addition to FTQ

In experiment 1 we evaluated each of the eight combinations, in addition to full-text queries. Figure 3 presents the four outperforming CBQs, in addition to the FTQ baseline. Generally at most of the recall levels all the four CBQs, including 3OO, which was significantly greater at the 0.05 significance level when compared at 0.5 recall level, 3AO, 3OA and 2OO in decreasing order, outperformed the FTQ baseline, while beyond the 0.6 recall level the FTQ outperformed. In addition the additional four CBQs (not appearing in figure 3) were 2OA, 2AO, 3AA and 2AA. The four outperforming CBQs can be characterized by querying the 3rd level of the hierarchy, which outperformed the 2nd level. Three of them have OR set to the *outer*, and to the *inner* logic operators. Note that the 3OO outperformed at most of the recall levels while the 3AO outperformed at the low recall levels.

Experiment II - CBQ in addition to single CSQ

In experiment 2 we evaluated each of the eight combinations, in addition to a single context query. Figure 4 presents the four outperforming CBQs, in addition to the single CSQ baseline. Generally, at most of the recall levels all the four CBQs, including 3OA, 3OO, 3AO and 2OO, in decreasing order, outperformed the single CSQ baseline (1CSQ), while beyond the level of 0.3 recall the 2OO was bellow the baseline. In addition the additional four CBQs (not appearing in figure 4) were 2OA, 2AO, 3AA and 2AA. The four outperforming CBQs can be characterized by querying the 3rd level of the hierarchy, which outperformed the 2nd level. Three of them have OR set to the *outer*, and to the *inner* logic operators. Note that while all the four outperforming CBQs were the same, as in experiment 1, but in a slight different order, the best CBQ in the experiment two was 3OA.

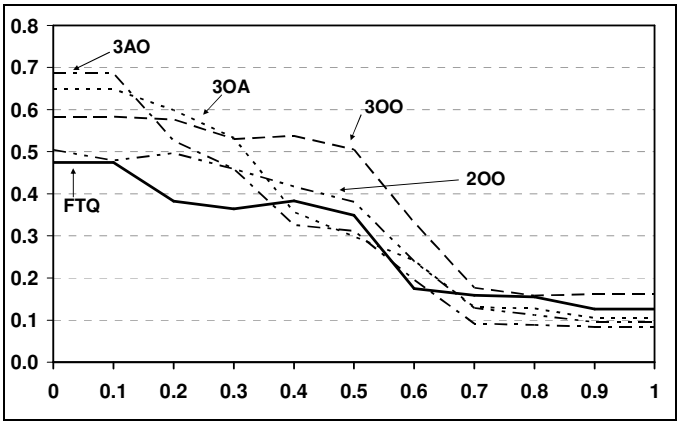

Figure 3 – CBQs applied in addition to the full text queries baseline, in which 3OO, 3AO, 3OA and 2OO outperform the baseline.

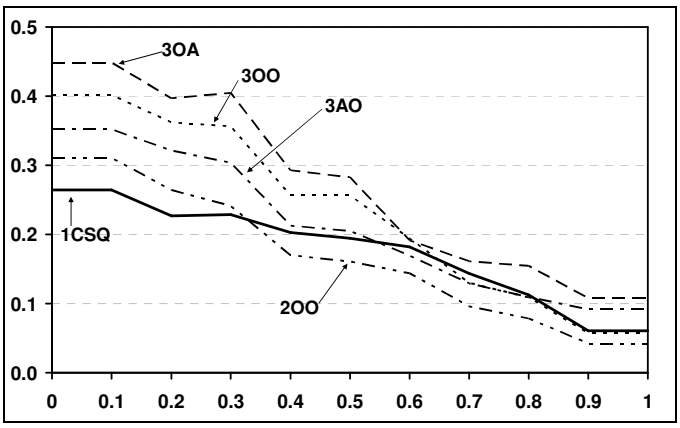

Figure 4 – CBQs applied in addition to single context queries baseline, in which 3OA, 3OO, 3AO and 2OO outperform the baseline at most of the recall level

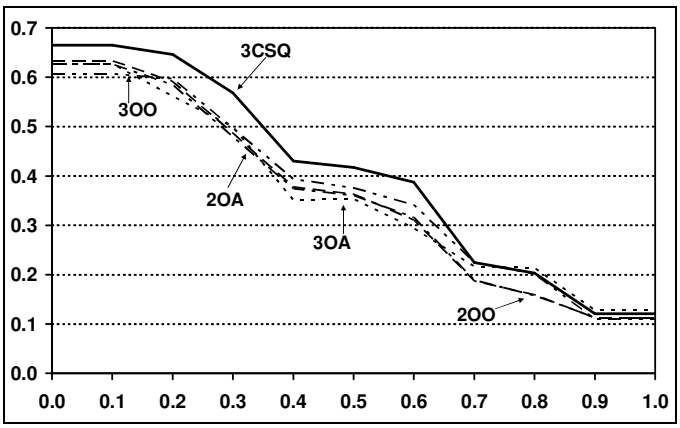

*Figure 5 – CBQs applied in addition to three context queries baseline, in which the CBQs (*2OO, 3OA, 2OA and 3OO*) constantly decreased the baseline (3CSQ).*

Experiment III - CBQ in addition to three CSQs

In experiment 3 we evaluated each of the eight combinations, in addition to three context queries. Figure 5 presents the four

outperforming CBQs, in addition to three CSQs baseline. Generally, along all the recall levels all the four CBQs, including: 2OO, 3OA, 2OA and 3OO, in decreasing order, performed lower than the three CSQs baseline (3CSQ). In addition the additional four CBQs (not appearing in figure 5) were 3AO, 3AA, 2AO and 2AA. Unlike the previous experiments, in which CBQs improved the baseline, here it decreased. We will refer to this in the discussion and conclusion section; however, again three of the four outperforming CBQs appear in the first and second experiments. In this experiment while there were two CBQs querying at the 2nd and 3rd hierarchy level, as well as the inner operator, the outer operator is OR at all the four top CBQs. Note that while 2OO was the fourth at both previous experiments, it is the first here (within the CBQs).

Discussion

We presented Vaidurya, focusing on its concept based search methods, the hypotheses of the research, the corresponding experiments, in which CBQs in eight settings were applied to varying baselines. Testing hypothesis 1, in which we expected to have an improvement when applying CBQs in addition to varying baselines, shown a significant improvement when applied in addition to FTQ. An improvement was found in addition to a single CSQ, while decreased when used in addition to three CSQs. The reason for the difference between these three scenarios might be that the initial baseline precision in the last case (i.e., when using the three CSQs) was much higher than that when using FTQ or a single CSQ (see figures 3, 4 and 5). In addition, note that the improvement was *greater* when the baseline was *lower* (see figures 3 and 4). Referring to hypothesis 2, querying at the third level outperformed the second level, especially in experiments 1 and 2, while in the third it was even. Referring to hypothesis 3, three CBQs 3OA, 3OO and 2OO appeared within the four outperforming CBQs repeatedly within the three experiments. While 2OO was the last (within the four outperforming CBQs) in experiments 1 and 2, it was the first in experiment 3, which may be explained as well by the high level of the baseline performance.

To summarize, an improvement by applying CBQ in addition to a textual search can be achieved, especially when querying at the third level, setting the *outer* logic operator to OR or AND, and setting OR to the *inner* logic operator. Note that using AND-AND achieved the lowest performance.

Previous studies examining CBS have shown that it does not necessarily improve, and might even worsen, the search-engine's performance [4]. 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 CBS the user can manually specify queried concepts from a given predefined ontology of concepts. The limitations of this study are mainly caused by the size of the test collection, which is relatively small. However, in contrast to huge test collections, in which judgments are specified automatically, based on an ensemble of search engines, in our test collection the reviewers browsed manually each of the CPGs and indicated their relevance to the query. We are currently in the process of extending this method to enable a user enter *simply* a textual query which will be converted to a conceptual representation and will be queried in addition to the FTQ. Preliminary evaluation results on the Trec-Genomics are encouraging.

References

[1] Humphreys BL and Lindberg DA. The UMLS project: making the conceptual connection between users and the information they need. Bulletin Medical Library Association, 81(2): p. 170-177, 1993.

[2] Concard J, Yang C, and Claussen J. Effective collection metasearch in a hierarchical environment: Global vs. localized retrieval performance. Proc of SIGIR, 2002.

[3] Wang W, Meng W, and Yu C. Concept hierarchy based text database categorization in a metasearch engine environment. Proc of WISE '00. 2000.

[4] Hersh WR, Hickam DH, Haynes RB, and McKibbon KA. A performance and failure analysis of SAPHIRE with a MEDLINE test collection, JAMIA, Vol 1: p. 51-60, 1994.

[5] 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, 37: p. 325-344, 2004.

[6] Robert Moskovitch, Martins SB, Behiri E, Weiss A, and Shahar Y. A Comparative Evaluation of Full-text, Concept-Based, and Context-Sensitive Search. JAMIA, 14: p. 164-174, 2007.

[7] Hersh WR, Price S., and Donohoe L. Assessing thesaurus-based query expansion using the UMLS Metathesaurus. In Proceedings of AMIA, 2000.

[8] Aronson AR and Rindflesch TC. Query expansion using the UMLS Metathesaurus. In Proceedings of AMIA Annual Fall Symposium, p. 485-489, 1997.

[9] Srinivasan P. Retrieval feedback in MEDLINE. JAMIA, 3(2): p. 157-167, 1996.

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

[11] 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, 37: p. 177-190, 2006.

[12] Salton G and McGill M. Introduction to Modern Information retrieval, New York: McGraw-Hill, 1983.

Address for correspondence

robertmo@bgumail.bgu.ac.il

Department of Information Engineering,
Ben Gurion University of the Negev, Israel
P.O.B. 653, Beer Sheva 84105, Israel.