SkylineSearch: Semantic Ranking and Result Visualization for PubMed *

Julia Stoyanovich[2] Mayur Lodha[1] William Mee[1] Kenneth A. Ross[1]
jstoy@cis.upenn.edu {mdl2130, wjm2107, kar}@cs.columbia.edu
[1]CS Department, Columbia University, New York, NY, USA
[2]CIS Department, University of Pennsylvania, Philadelphia, PA, USA

ABSTRACT

Life sciences researchers perform scientific literature search as part of their daily activities. Many such searches are executed against *PubMed*, a central repository of life sciences articles, and often return hundreds, or even thousands, of results, pointing to the need for data exploration tools. In this demonstration we present *SkylineSearch*, a semantic ranking and result visualization system designed specifically for *PubMed*, and available to the scientific community at `skyline.cs.columbia.edu`. Our system leverages semantic annotations of articles with terms from the *MeSH* controlled vocabulary, and presents results as a two-dimensional skyline, plotting relevance against publication date. We demonstrate that *SkylineSearch* supports a richer data exploration experience than does the search functionality of *PubMed*, allowing users to find relevant references more easily. We also show that *SkylineSearch* executes queries and presents results in interactive time.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Scientific databases*; H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User Interfaces*

General Terms

Algorithms, Human Factors, Performance

Keywords

Data Exploration, Skyline

1. INTRODUCTION

Literature search is a central task in scientific research. *PubMed* (`www.pubmed.gov`) is the most significant bibliographic source in life sciences, and many researchers, practitioners, and students search *PubMed* as part of their daily activities. *PubMed* currently indexes over 20 million articles that date back to 1865, and the number of new articles increases steadily from year to year.

PubMed articles are annotated by a staff of indexers with terms from the Medical Subject Headings (*MeSH*) controlled vocabulary (`www.nlm.nih.gov/mesh`). *MeSH* organizes terms into a hierarchy, allowing searching at various levels of specificity. *MeSH* annotations are currently used for query expansion: a query that matches a term will return all articles annotated with that term or with its descendants in *MeSH*. Due to the size of *PubMed*, and to the comprehensive query expansion strategy, many queries return hundreds, or even thousands, of results.

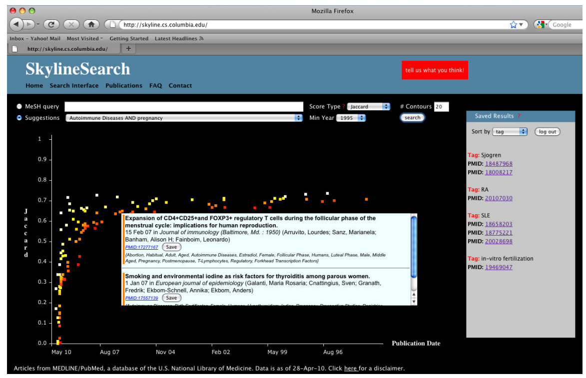

Figure 1: A screenshot of the *SkylineSearch* interface.

For example, the query *Autoimmune Diseases AND Pregnancy* returned over 11,000 results on October 28, 2010. The *PubMed* search interface supports the sorting of results by publication date, first or last author, title, and publication venue. Navigating a result set that contains 11,000 matches in sorted order, by a single meta-data field, has several limitations. First, while publication date is a valuable dimension in scientific literature search, a user may want to find relevant articles even if they are not among the most recent few. Second, while author and journal names can help the user find a reference of which he is already aware, sorting on these meta-data fields does not conveniently support true information discovery.

In [5] we proposed to use *MeSH* annotations for relevance ranking of *PubMed* search results. We found that *MeSH* has a unique structure, and we termed it a *scoped polyhierarchy*. Figure 2 presents a portion of *MeSH* that models autoimmune and connective tissue diseases. The hierarchy is a tree of *nodes*, with one or several nodes mapping to a single *term*. Thus, *MeSH* is a *polyhierarchy*. For example, the term “Arthritis, Rheumatoid” (RA) is represented by two nodes in *MeSH*. A term may span a subtree of different shape in different branches, and so the polyhierarchy is *scoped*. So, RA has six descendants in the “Rheumatic Diseases” subtree and four descendants in the “Autoimmune Diseases” subtree, representing that not all types of RA are autoimmune.

In [5], we developed several notions of relevance that are appropriate for scoped polyhierarchies, and showed how result relevance

*This research was supported in part by NSF grant 0937060 to the Computing Research Association for the CIFellows Project, NSF grant IIS-0915956, and NIH grant 5 U54 CA121852-05.

SIGMOD’11, June 12–16, 2011, Athens, Greece.
ACM 978-1-4503-0661-4/11/06.

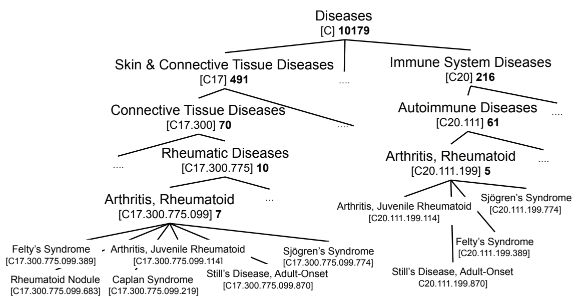

Figure 2: A portion of *MeSH*: a scoped polyhierarchy.

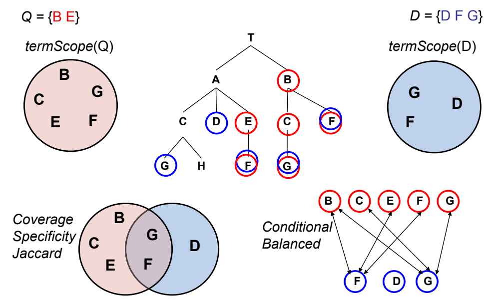

Figure 3: An illustration of the similarity measures.

can be computed efficiently on the scale of *PubMed* and *MeSH*. We also presented a novel algorithm that efficiently computes a two-dimensional skyline of results, plotting relevance against publication date. In this demonstration we present *SkylineSearch*, an on-line system that implements and extends the techniques of [5], and integrates them into a complete result visualization system. A screenshot of our system is presented in Figure 1.

Our system improves on our prior work in two important ways. First, in [5] we showed that ranking and skyline computation that use our proposed measures of relevance can be executed efficiently, *given a set of candidate matches*. In this demonstration we will show that the *end-to-end computation*, described in Section 2, can also be executed in interactive time. Second, in our prior work we presented results of a user study that explored the effectiveness of our relevance measures *for ranking*. The *SkylineSearch* system goes further, and shows that our relevance measures can lead to a richer user experience when used to construct *a two-dimensional skyline*. We will demonstrate that *SkylineSearch* allows users to find relevant references more easily than does the *PubMed* search interface. Demonstration scenarios are outlined in Section 3.

2. SYSTEM OVERVIEW

2.1 User Interface

Figure 1 presents a screenshot of the *SkylineSearch* user interface. The user interacts with the system by issuing a keyword query to be executed against *PubMed*. A typical query will consist of one or several keywords, optionally connected with AND or OR, e.g., *autoimmune diseases AND pregnancy* and *Alzheimer disease*. Individual keywords, or groups of keywords, may also be designated as *MeSH* terms, for example *connective tissue diseases[MeSH Terms] AND autoimmune diseases[MeSH Terms]*. The user also chooses one of the similarity measures, described in Section 2.2, and sets the number of skyline contours, defined below.

SkylineSearch evaluates the query, computes the skyline of results, and displays the skyline, using the process described in Section 2.3. In addition to computing the skyline of the result set, our system computes up to 20 *skyline contours*. Skyline contours are useful for highlighting points that are close to the skyline, and that might be of interest to the user. A skyline contour is defined inductively as follows: (i) a point belongs to the 1^{st} contour iff it belongs to the skyline of the whole data set; (ii) a point belongs to the k^{th} contour iff it belongs to the skyline of the data set obtained by removing points from the 1^{st} through $k - 1^{st}$ contours.

Candidate results are processed by *SkylineSearch* in batches, and skyline points are presented to the user as soon as the first batch has been processed. The user may mouse over a point that represents a result, displaying a description of the corresponding *PubMed* article. The description lists the title, authors, publication venue and date, and *MeSH* annotations, and allows to navigate to the article in *PubMed*. The user may *save* and *tag* results for future reference.

2.2 Similarity Measures

At the heart of *SkylineSearch* lie the similarity measures used to access the relevance of a *PubMed* article to a query. In [5] we introduced three measures that exploit the unique structure of *MeSH*. We now briefly describe these measures, using a small scoped polyhierarchy in Figure 3. In defining the measures we use the notion of a *term-scope*. The term-scope of a query, denoted by $termScope(Q)$ is the set of *MeSH* terms that the query matches, along with all descendants. The term-scope of a document, $termScope(D)$, is defined analogously. In Figure 3, $Q = \{B, E\}$ and $D = \{D, G, F\}$; $termScope(Q) = \{B, C, G, F, E\}$ is represented by a red circle, $termScope(D) = \{D, G, F\}$ is represented by a blue circle.

SkylineSearch implements five similarity measures, illustrated in Figure 3. *Coverage* measures how exhaustively the document answers the query, and is computed as $\frac{|termScope(Q) \cap termScope(D)|}{|termScope(Q)|}$. *Specificity* measures how specific the document is to the query, and is computed as $\frac{|termScope(Q) \cap termScope(D)|}{|termScope(D)|}$. *Jaccard* is the harmonic mean of coverage and specificity. These are adaptations of the $TermSim$ measure from [5]. *Conditional* counts the number of ancestor-descendant pairs (s, t), where $s \in termScope(Q)$ and $t \in termScope(D)$; it implements $CondSim$ from [5]. *Balanced* is similar to *Conditional*, but it normalizes the contribution of each query term to the score; it implements $BalancedSim$ from [5].

2.3 System Architecture

SkylineSearch is implemented in Java and uses an Oracle 10.2 database for persistent storage. Figure 4 describes the system architecture and the data flow. The main components of *SkylineSearch* are the *Query Manager*, the *In-Memory DB* and the *Oracle DB*. The *PubMed* dataset is external, and is accessed by our system at query time and at data load time.

The *In-Memory DB* stores *MeSH* annotations of all *PubMed* articles, and currently requires about 8GB of RAM. The database was implemented by us for this project; it includes our custom indexing structures and implements the optimized ranking and skyline computation algorithms presented in [5]. The database resides on the server, and there are no additional memory requirements for the client, which communicates with the database via a browser-based application. The system can take advantage of multiple physical cores by satisfying multiple requests concurrently.

The *In-Memory DB* is incrementally updated each month, when information about newly published articles becomes available. This

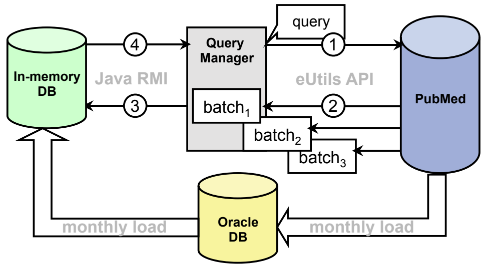

Figure 4: System architecture of *SkylineSearch*.

information is downloaded via FTP from the *PubMed* repository, parsed by a collection of custom scripts, and stored in our local Oracle database. A full load of the *In-Memory DB* from Oracle takes about 3 hours, and an incremental load takes several minutes. The size of the database is linear in the number of *PubMed* articles.

Query processing is coordinated by the *Query Manager* that receives a query from the user and communicates with *PubMed* via the eUtils API (arrow 1). The *Entrez* search engine evaluates the query against the live *PubMed* database, and returns ids of results in batches, sorted in decreasing order of publication date (arrow 2). *Query Manager* ships result ids over to the *In-Memory DB* via Java RMI, submitting a skyline computation request (arrow 3). *In-Memory DB* handles outstanding computation requests asynchronously. *Query Manager* picks up available results (arrow 4) and passes them over to the visualization component.

3. DEMONSTRATION SCENARIO

We demonstrate the functionality of *SkylineSearch* by executing queries provided by us and by members of the audience, and by discussing the results. As we execute the queries, we observe that *SkylineSearch* responds in interactive time. The total run-time is dominated by the round-trip time to *PubMed* (arrows 1 and 2 in Figure 4), over which we do not have direct control. Nonetheless, all queries we considered so far returned the first set of results within several seconds. We now discuss two demonstration scenarios.

Use Case 1: *Diabetes Mellitus AND Myocardial Infarction* is a general query that returns survey articles, studies of particular aspects of the disorders, and case reports. The query matched 8109 *PubMed* articles on October 28, 2010. Of these, 2559 were published in the last five years, and 481 in the last year. The query matches two *MeSH* terms — *Diabetes Mellitus* and *Myocardial Infarction*, each mapping to two *MeSH* nodes and exhibiting polyhierarchy features. However, scoping is not pronounced for this query: *Myocardial Infarction* spans subtrees of size 5 and 6, and *Diabetes Mellitus* spans two subtrees of size 8 each.

SkylineSearch, when executed with *Jaccard* similarity and 20 skyline contours, produces a skyline of an interesting shape. Many articles *published in the last year* are shown, with relevance score reaching 0.7. Recent case reports, descriptions of cohort studies, and treatment outcomes with particular drugs are shown here, with relatively low scores (e.g., *PubMed* ids 19330466, 20181922, 20584880). One of the highest-scoring articles in this group, with *PubMed* id 19910536, deals with the relationship between diastolic dysfunction and cardiovascular failure in diabetes, following myocardial infarction. This article is both very relevant to the query and fairly recent, appearing in January 2010. However, it cannot be easily found via a sorted list provided by *PubMed*, where it appears in position 363 when the list is sorted by publication date.

Articles *published over a year ago* are shown in the higher-scoring part of the skyline, and include mostly survey articles (e.g., *PubMed* ids 16275205 and 15928277). These articles are not easily accessible via a sorted list provided by *PubMed*, since they do not start appearing until position 482 in the ranked list, and are not easy to tell apart from case reports and pharmacological studies.

With *Conditional* and *Balanced* similarity, results follow a similar trend, because scoping is not pronounced in this query. The actual set of skyline points is somewhat different than with *Jaccard*, because of different score normalization (see Section 2.2).

Use Case 2: *Autoimmune Diseases AND Pregnancy Complications* matches two *MeSH* terms, spanning subtrees of size 64 and 91, respectively. Similar trends are observed for this query with *Jaccard* similarity as for use case 1. Additionally, because both query terms exhibit scoped polyhierarchy features in their subtrees, and because they span large subtrees of different size — *Conditional*, and particularly *Balanced*, relevance measures produce very interesting results on the skyline. The skyline and early contours computed with the *Balanced* measure contain several articles about pregnancy complications in Systemic Lupus Erythematosus and in Antiphosopholipid Syndrome, e.g., *PubMed* ids 19897518 and 17283586. These autoimmune disorders are commonly associated with pregnancy complications, and so these are high-quality answers. The answers appear prominently on the skyline, but would not have been easy to find in *PubMed*'s sorted list, because they are not among the most recent few results.

4. RELATED WORK

Rada and Bicknell [4] considered ranking *MEDLINE* documents using *MeSH*, and modeled the distance between the query and the document as the mean path-length between all pairs of document and query terms. This is one of several distance-based measures; see also Lee and Kim [3]. In contrast, we compare sets of terms via common descendants, not via common ancestors. We believe that descendant-based similarity is more appropriate, because query expansion in *PubMed* also incorporates descendants of a query term. Query expansion and ranking are parts of the same ranked retrieval process, and should agree on the semantics of relevance.

Several systems for bibliographic search in life sciences have been developed, see Kim and Rebholz-Schuhmann [2] for a review. The system closest ours, *GoPubMed* [1], uses three ontologies — *GO*, *MeSH* and *Uniprot*, to organize *PubMed* query results in a faceted hierarchy. When multiple ontology terms appear in the hierarchy, the system allows navigation by each of the terms. Unlike in our work, contributions of multiple terms are not reconciled into a single score, and no skyline visualization of results is provided.

5. REFERENCES

[1] A. Doms and M. Schroeder. GoPubMed: Exploring PubMed with the GeneOntology. *Nucleic Acid Research*, 33, 2005.

[2] J.-J. Kim and D. Rebholz-Schuhmann. Categorization of services for seeking information in biomedical literature: a typology for improvement of practice. *Brief Bioinform*, 9(6), 2008.

[3] J. Lee and M. Kim. Information retrieval based on a conceptual distance in is-a hierarchy. *J Doc*, 49, 1993.

[4] R. Rada and E. Bicknell. Ranking documents with a thesaurus. *JASIS*, 40(5), 1989.

[5] J. Stoyanovich, W. Mee, and K. A. Ross. Semantic ranking and result visualization for life sciences publications. In *ICDE*, 2010.