Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Effective Navigation of Query Results for Knowledge on Data Engineering

M. Sreedevi
Assistant Professor, Dept. of Comp. Science,
Sri Venkateswara University, Tirupati , India.

Abstract--- The MEDLINE database, on which the PubMed search engine operates, contains over 18 million citations, and the database is currently growing at the rate of 500,000 new citations each year. Other biological sources, such as Entrez Gene and OMIM, witness similar growth. As claimed in previous work, the ability to rapidly survey this literature constitutes a necessary step toward both the design and the interpretation of any large scale experiment. Biologists, chemists, medical and health scientists are used to searching their domain literature – such as PubMed– using a keyword search interface. Currently, in an exploratory scenario where the user tries to find citations relevant to her line of research and hence not known a priori, she submits an initially broad keyword- based query that typically returns a large number of results. Subsequently, the user iteratively refines the query, if she has an idea of how to, by adding more keywords, and re-submits it, until a relatively small number of results are returned. This refinement process is problematic because after a number of iterations the user is not aware if she has over-specified the query, in which case relevant citations might be excluded from the final query result.

Keywords: MEDLine, PubMed, MeSH.

I. Introduction

EEDS of sinternet or networks are cropped from that day onwards different users are increases rapidly. Obviously number of applications also increases. As many friendly interfaces, applications are available in the networks of networks, then the users are added speedily. When number of users are increases there are issues will be arises namely database management, security, traffic problem, etc.As an example, a query on PubMed for "cancer" returns more than 2 million citations. Even a more specific query for "prothymosin", a nucleoprotein gaining attention for its putative role in cancer development, returns 313 citations. The size of the query result makes it difficult for the user to find the citations that she is most interested in, and a large amount of effort is expended searching for these results. Many solutions have been proposed to address this problem –commonly referred to as information overload. These approaches can be broadly classified into two classes: ranking and categorization, which can also be combined.

BioNav belongs primarily to the categorization class, which is ideal for this domain given the rich concept hierarchies (e.g., MeSH) available for biomedical data. We augment our categorization techniques with simple ranking techniques. BioNav organizes the query results into a dynamic hierarchy, the navigation tree. Each concept (node) of the hierarchy has a descriptive label. The user then navigates this tree structure, in a top-down fashion, exploring the concepts of interest while ignoring the rest. An intuitive way to categorize the results of a query on PubMed is using the MeSH static concept hierarchy [18], thus utilizing the initiative of the US National Library of Medicine (NLM) to build and maintain such a comprehensive structure. Each citation in MEDLINE is associated with several MeSH concepts in two ways: (i) by being explicitly annotated with them, and (ii) by mentioning those in their text (see Section 7 for details). Since these associations are provided by PubMed, a relatively straightforward interface to navigate the query result Would first attach the citations to the corresponding MeSH concept nodes and then let the user navigate the navigation tree. Fig. 1 displays a snapshot of such an interface where shown next to each node label is the count of distinct citations in the subtree rooted at that node. A typical navigation starts by revealing the children of the root ranked by their citation count, and is continued by the user expanding on or more of them, revealing their ranked children and so on, until she clicks on a concept and inspects the citations attached to it. A similar interface and navigation method is used by e-commerce sites, such as Amazon and eBay. For this example, we assume that the user will navigate to the three indicated concepts corresponding to three independent lines of research related to prothymosin.

II. Related Work

Hierarchical Navigation

Providing access to every page on your website. Hierarchical navigation ties your website together Often referred to as Web navigation the Hierarchical navigation model goes from the general to the specific; from a homepage to main sections to subsections and databases. It is a way to tie together many areas of information into a working website structure. A visitor could easily go from the homepage to other areas of the website and back again. Providing clear and

simple access. The goal of any hierarchical website navigation system is to offer the user a clear and simple way to access all pages in a site easily and to do so quickly from anywhere in the website.

Hierarchical navigation is the most popular navigation model on the web

This type of navigation is most commonly used by most webmasters and is often referred to as web navigation. It is a combination of linear and database structures which are interconnected to provide access to any page in the website.

Hierarchical navigation and sitemaps

A good example of hierarchical navigation is a sitemap. Proper sitemap construction insures that the links are arranged into a pattern by the webmaster and are easily accessible by the visitor. Any page on your website can be easily called up.

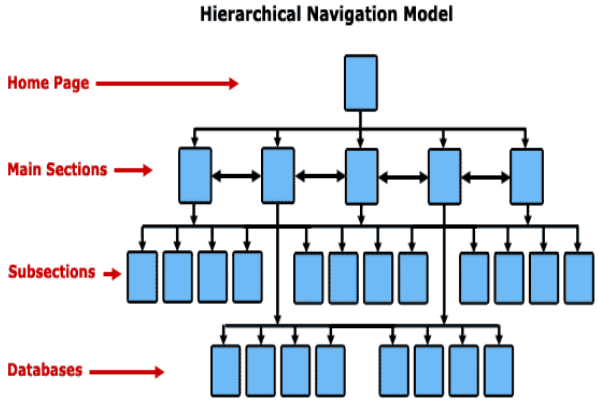


Fig.: Hierarchical Navigation Link Diagram

Clustering Systems

A major problem faced in biomedical informatics involves how best to present information retrieval results. When a single query retrieves many results, simply showing them as a long list often provides poor overview. With a goal of presenting users with reduced sets of relevant citations, this study developed an approach that retrieved and organized MEDLINE citations into different topical groups and prioritized important citations in each group.

A text mining system framework for automatic document clustering and ranking organized MEDLINE citations following simple PubMed queries. The system grouped the retrieved citations, ranked the citations in each cluster, and generated a set of keywords and MeSH terms to describe the common theme of each cluster.

Several possible ranking functions were compared, including citation count per year (CCPY), citation count (CC), and journal impact factor (JIF). We evaluated this framework by identifying as "important" those articles selected by the Surgical Oncology Society. Our results showed that CCPY outperforms CC and JIF, i.e., CCPY better ranked important articles than did the others. Furthermore, our text clustering and knowledge extraction strategy grouped the retrieval results into informative clusters as revealed by the keywords and MeSH terms extracted from the documents in each cluster. The text mining system studied effectively integrated text clustering, text summarization, and text ranking and organized MEDLINE retrieval results into different topical groups.

III. Approaches For Concept Hierarchies

A. Query Search process module (or) Biomedical Search Systems module

PubMed– using a keyword search interface. Currently, in an exploratory scenario where the user tries to find citations relevant to her line of research and hence not known a priori, she submits an initially broad keyword- based query that typically returns a large number of results. Subsequently, the user iteratively refines the query, if she has an idea of how to, by adding more keywords, and re-submits it, until a relatively small number of results are returned. This refinement process is problematic because after a number of iterations the user is not aware if she has over-

specified the query, in which case relevant citations might be excluded from the final query result.

Query on PubMed is using the MeSH static concept hierarchy, thus utilizing the initiative of the US National Library of Medicine (NLM) to build and maintain such a comprehensive structure. Each citation in MEDLINE is associated with several MeSH concepts in two ways: (i) by being explicitly annotated with them, and (ii) by mentioning those in their text. Since these associations are provided by PubMed, a relatively straightforward interface to navigate the query result would first attach the citations to the corresponding MeSH concept nodes and then let the user navigate the navigation tree.

B. Concept Hierarchy

A *Concept Hierarchy H (V, E, r)* is a labeled tree consisting of a set V of concept nodes, a set E of edges and is rooted at node r. Each node n€V has a label l and a unique identifier *id*.

C. Dynamic navigation tree module

Navigation tree. Fig displays a snapshot of such an interface where shown next to each node label is the count of distinct citations in the subtree rooted at that node. A typical navigation starts by revealing the children of the root ranked by their citation count, and is continued by the user expanding on or more of them, revealing their ranked children and so on, until she clicks on a concept and inspects the citations attached to it. A similar interface and navigation method is

used by e-commerce sites, such as Amazon and eBay. For this example, we assume that the user will navigate to the three indicated concepts corresponding to three independent lines of research related to prothymosin.

Fig.: Dynamic navigation tree module

BioNav introduces a dynamic navigation method that depends on the particular query result at hand and is demonstrated in Fig The query results are attached to the corresponding MeSH concept nodes as in Fig. but then the navigation proceeds differently. The key action on the interface is the expansion of a node that selectively reveals a ranked list of descendant (not necessarily children) concepts, instead of simply showing all its children.

***C. Navigation Tree*:**

A *Navigation T (V, E, r)* is the maximum embedding of an initial navigation tree *T (V1, E2, r)* such that no node n€V is labeled with an empty results list L (N), excluding the root (in order to maintain the tree structure and avoid the creation of a forest).

D. Hierarchy navigation web (interface) search module

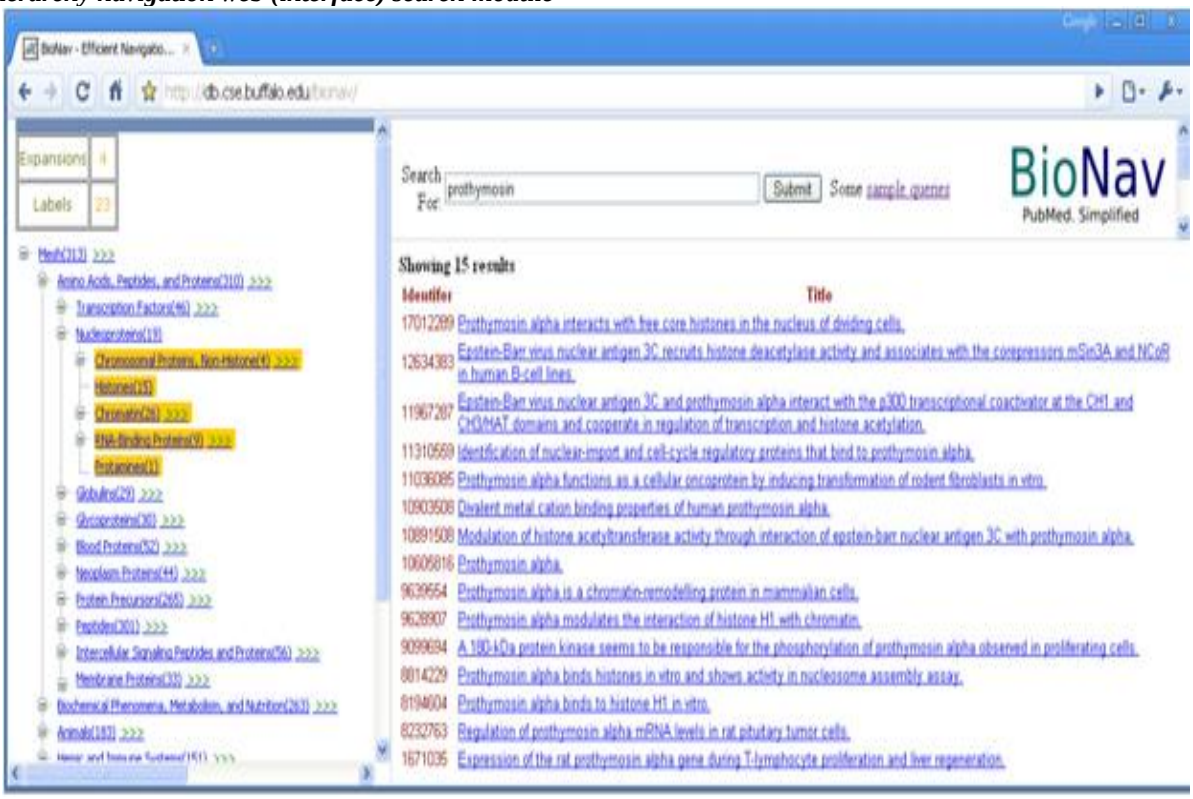


Fig.: Hierarchy navigation web (interface) search module

BioNav belongs primarily to the categorization class, which is ideal for this domain given the rich concept hierarchies (e.g., MeSH) available for biomedical data. We augment our categorization techniques with simple ranking techniques. BioNav organizes the query results into a dynamic hierarchy, the navigation tree. Each concept (node) of the hierarchy has a descriptive label. The user then navigates this tree structure, in a top-down fashion, exploring the concepts of interest while ignoring the rest.

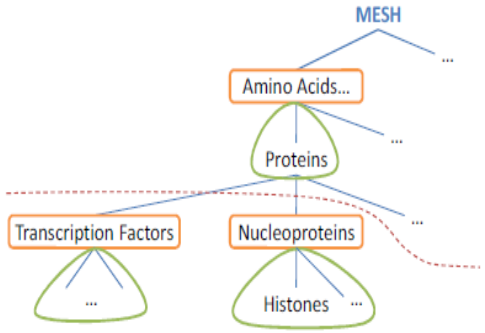
E. Query Workload online operation module

On-Line Operation. Upon receiving a keyword query from the user, BioNav executes the same query against the MEDLINE database and retrieves only the IDs

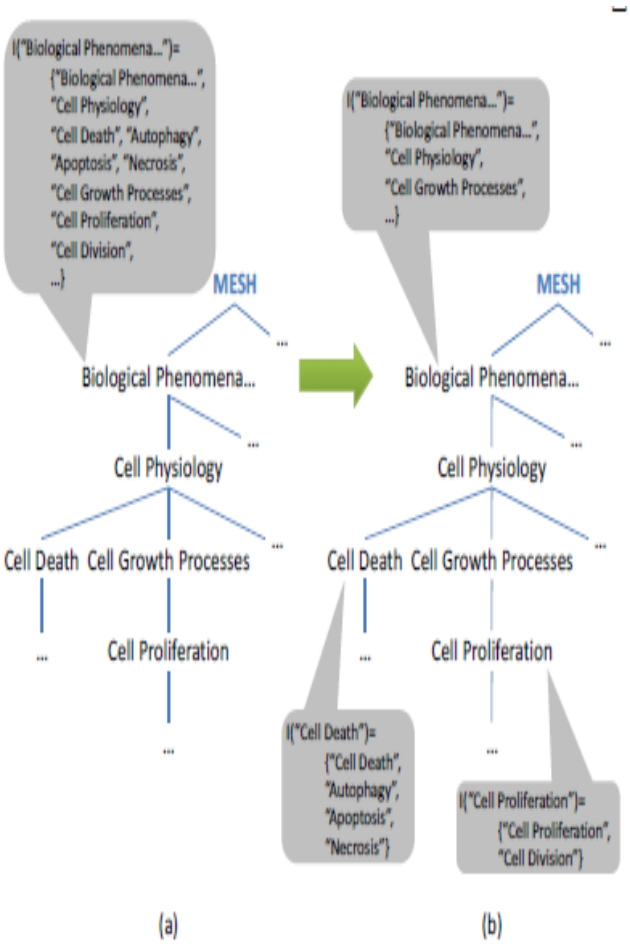
(Pub Med Identifiers) of the citations in the query result. This is done using the ESearch utility of the Entrez Programming Utilities (eUtils). eUtils are a collection of web interfaces to PubMed for issuing a query and downloading the results with various levels of detail and in a variety of formats. Next, the navigation tree is constructed by retrieving the MeSH concepts associated with each citation in the query result from the BioNav database. This is possible since MeSH concepts have tree identifiers encoding their location in the MeSH hierarchy, which are also retrieved from the BioNav database. This process is done once for each user query.

F. Valid EdgeCut

A valid EdgeCut of a tree *T (V, E, r)* is an EdgeCut CCE such that no two edges in C appear in a path from the root to a leaf node.



Navigation Tree,EdgeCut and Component sub trees



IV. Motivation

The proposals dynamically categorize SQL query results by inferring a hierarchy based on the characteristics of the result tuples. Their domain is the tuple attributes and their problem is how to organize them hierarchically in order to minimize the navigation cost. They also decide the value ranges for each attribute, for both categorical and numerical ones, and how to rank them. One of the systems takes into consideration the user's preferences during the inference for a more personalized experience. Once the hierarchy is inferred, they follow a static navigation method. BioNav is distinct since it offers dynamic navigation on a predefined hierarchy, as is the MeSH concept hierarchy. Hence, BioNav is complementary to these systems, since it can be used to optimize the navigation, after these systems construct the navigation tree.

V. Conclusion

Information overload is a major problem when searching biomedical databases such as PubMed, where typically a large number of citations are returned, of which only a small subset is relevant to the user. In this paper, we presented the BioNav system to address this problem. Our solution is to organize the query results according to their associations to concepts of the MeSH concept hierarchy, and provide a dynamic navigation method that minimizes the information overload observed by the user. When the user expands a MeSH concept on our web interface, BioNav reveals only a selective list of descendant concepts, instead of simply showing all its children, ranked based on their estimated relevance to the user's query. We formally stated the underlying framework and the navigation and cost models used for the evaluation of our approach. Our complexity result proved that the problem of expanding the navigation tree in a way that minimizes the user's navigation cost is NP-complete. A feasible (for small trees) optimal algorithm and an efficient heuristic were developed. Experimental results validated the effectiveness of the proposed heuristic for diverse sets of queries and navigation trees, when compared to categorization systems using a static navigation method. The architecture of the BioNav system was implemented and is available at http://db.cse.buffalo.edu/bionav.

References:

[1] J S. Agrawal, S. Chaudhuri, G. Das and A. Gionis: *Automated Ranking of Database Query Results*. In Proceedings of First Biennial Conference on Innovative Data Systems Research (CIDR), 2003.

[2] K. Chakrabarti, S. Chaudhuri and S.W. Hwang: *Automatic Categorization of Query Results*. SIGMOD Conference 2004: 755-766.

[3] Z. Chen and T. Li: *Addressing Diverse User Preferences in SQLQuery-Result Navigation*. SIGMOD Conference 2007: 641-652

[4] T. Zhang, R. Ramakrishnan and M. Livny: *BIRCH: An Efficient Data Clustering Method for Very Large Databases*. SIGMOD Conference 1996: 103-114

[5] V. Hristidis and Y. Papakonstantinou: *DISCOVER: Keyword Search in Relational Databases*. In Proc. of VLDB Conference, 2002

[6] R. Hoffman and A. Valencia: *A gene network for navigating the literature.* Nature Genetics, 36(7):664, 2004.

Author :

M. Sreedevi
Assistant Professor
Dept.ofComp.Science
Sri Venkateswara University Tirupati
A.P India.
Cell : 9440571597
E-mail: msreedevi_svu2007@yahoo.com

