Minimizing Navigation Cost Using Genetic Algorithm Based Best Edge Cut Algorithm

Jisha Jamal

Abstract-- With the dramatically quick and explosive growth of information available over the Internet, World Wide Web has become a powerful platform to store, disseminate and retrieve information as well as mine useful knowledge. Due to the properties of the huge, diverse, dynamic and unstructured nature of Web data, Web data research has encountered a lot of challenges. The most important problem when searching queries on biomedical databases is information overload. In this paper, the proposed approach gives the solutions for information overload and also ensures that the navigation is effective. Ranking and categorization manage the information overload. The proposed technique provides a novel search interface that facilitates end users to have effective navigation of query results that are presented in the form of concept hierarchies. Moreover the query results are presented in such a way that the navigation cost is minimized and thus giving rich user experience in this area. And also propose an efficient genetic algorithm based Best-Edge cut algorithm for relatively small trees. The empirical results revealed that the proposed navigation system is effective and can be adapted to real world systems where huge number of tuples is to be presented. In contrast, previous works expand the hierarchy in a predefined static manner, without navigation cost modeling.

Index Terms - Interactive data exploration and discovery, Search process, Interaction styles, Automatic categorization, biomedical data set, clustering, navigation cost, tree navigation

1. INTRODUCTION

The past decade has witnessed the modern advances of high-throughput technology and rapid growth of research capacity in producing large-scale biological data, both of which were concomitant with an exponential growth of biomedical literature. However, the acquisition of such information is becoming increasingly difficult due to its large volume and rapid growth. The goal of searching the literature is to find the right facts and the right references. In that endeavor, getting 100,000 hits is not better than retrieving 50,000 hits, when there were only 100 documents that were actually relevant. This has led to users to refine query with other keywords and get the desired results after many trials. Here it has to be observed that user time is wasted in refining search criteria and also the navigation of query results which are abundant and bulky. [1]The navigation cost is more as user has to spend lot of time in finding the required subset of rows from the bulk of search results. This problem has been researched in [1], [2], [3] and the problem is identified as information overload. And when you do a literature search to find an article to use as a reference, refining the results to get to the right reference can also be a hit-or-miss effort. Refining the search too much can exclude the key reference; not refining enough leaves you with a large haystack. This means you need a search technology that covers the relevant content and enables you to quickly hone in on the papers you need. This problem can be solved using concept hierarchies. Knowledge representation in the form of concepts and the relationships among them (Ontology) allows effective navigation.

For experiments, PubMed database which is in the public domain is used. The PubMed data is medical in nature and organized as per the annotations provided that is instrumental in making concept hierarchies to represent the whole dataset of PubMed. The PubMed contains over 18 million citations, and the database is growing at the rate of 500,000 new citations each year [4]. Other biological sources, such as Entrez Gene [17] and OMIM [20], witness similar growth. Users are often overwhelmed by the search results. Even though PubMed has number of advantages they are as follows,

□Users get an overview of the whole search result.
□They can choose the number of categorized results.
□It enables to derive the general keyword relevant to the search even though they are not mentioned in the article.

- [1]*Jisha Jamal is currently pursuing master's degree program in Department of CSE, KMCT College of Engineering,Calicut,India PH-+914792336218. E-mail: jishajml@gmail.com*

But there are two major challenges to address the user interests, (Chen and Li, 2007).

□ How to summarize the user interests from the behavior of all user already in the system.
□ How to decide the subset of user interests associated with a specific user.

The solutions are of two types namely categorization and ranking. However, these two can be combined to have more desired results. The proposed system is specially meant for presenting results in such a way that the navigation cost is reduced. For this purpose categorization techniques is used and concept hierarchies are built. The categorization techniques are supported by simple ranking techniques. The queries are first categorized and then ranking according to the user interests and then finally constructing a tree for the navigation of databases. The tree navigation is similar to a decision tree. That is the proposed solution effectively constructs a navigation tree that can reduce cost of navigation and user's experience is much better when compared with existing systems that do not use these techniques. The proposed system uses a cost model that lets it estimate the cost of navigation and make decisions in providing concept hierarchies. The cost of navigation is directly proportionate to the navigation sub tree [10] instead of the whole results in the tree. Earlier work on dynamic categorization of query results are in [2], [3], [5] and [6]. They made use of query dependent clusters based on the unsupervised technique. However, they neglect the process of navigation of clusters. In this aspect the proposed system is distinct and provides dynamic navigation on a pre-defined concept hierarchy. Another telling difference between existing systems and the proposed one is that the proposed system uses navigation cost model that minimizes navigation cost no matter what the bulky of search results is. Overall, the contributions are development of a framework for effective navigation of query results; a formal model for cost estimation; algorithm to optimize the results' navigation cost.

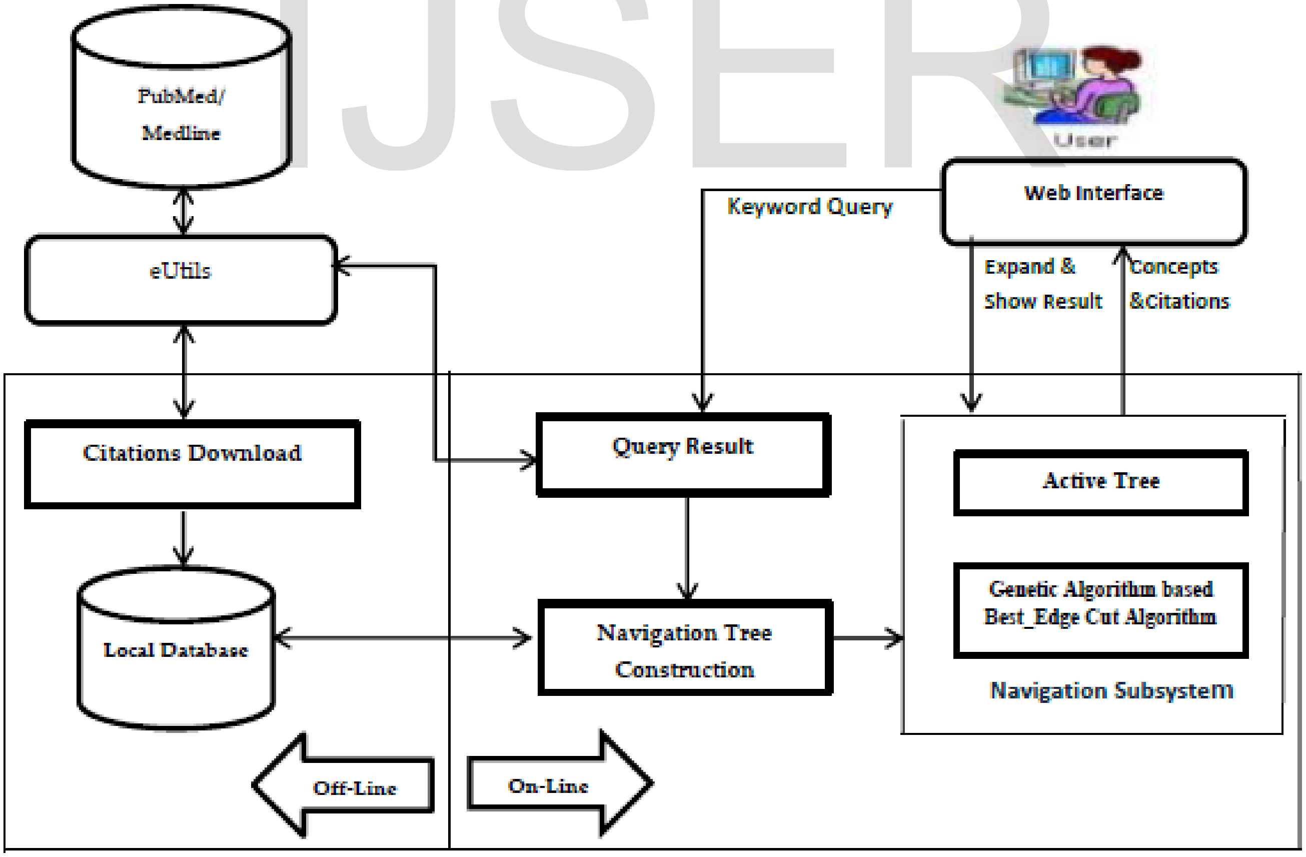


Fig.1 Architecture of Proposed Framework

2. Architecture Of Proposed Framework

The proposed framework is meant for making navigation of query results as effective as possible. The results of this project enable end users save lot of time as the proposed framework reduces the time taken to reach valuable content in the hierarchy.

As can be seen in fig. 1, the proposed framework has two phases such as Online and Offline. The offline phase performs operations in which user's active presence is not required. The Online phase is responsible to perform operations that are direct responses to user queries and also navigation operations made by user. The Medline DB citations can be loaded into local database using utility programs which are provided by the DB vendors. The concepts thus downloaded are stored in local database. In online phase, user enters a query. The query gets processed and results are obtained from database. The results then are used to construct concept hierarchies. The navigation sub system is responsible to take care of fine-tuning navigation tree so as to reduce the time for viewing desired results only. User is provided with a web based interface though which users can determine giving queries and the results get presented.

2.1 Navigation Model

Once user issues a query keyword, the proposed system generates an initial active tree and displaying its root to the user. Subsequently, the user navigates the tree by performing one of the following actions on a given component sub tree *I(n)* rooted at concept node *n*:

1. EXPAND I(n): The user clicks on the">>>" hyperlink next to node n and causes an Edge Cut(I(n)) operation to be performed on it, thus revealing a new set of concept nodes from the set I(n).
2. SHOWRESULTS I(n): By performing this action, the user sees the results list L(I(n)) of citations attached to the component subtree I(n).
3. IGNORE I(n): The user examines the label of concept node n, ignores it as unimportant and moves on to the next revealed concept.
4. BACKTRACK: The user decides to undo the last Edge Cut operation.[9]

User continues these operations until he gets the intended results. In order to define a cost model, this paper focuses on a simplification of the general navigation model, and call it TOPDOWN, where only EXPAND, SHOWRESULTS and IGNORE are the available operations, that is, the user follows a top-down only navigation starting from the root. TOPDOWN is common in practice. Note that when the user encounters a leaf node in TOPDOWN the only available option is SHOWRESULTS. The TOPDOWN navigation model is formally presented in Fig. 2. It is a recursive procedure and is initially called on the root of the initial active tree.

```
EXPLORE (I(n))
        if n is the root
                S← EXPAND I(n)
                For each ni  in S
                EXPLORE (I(ni))
        else, if n is not a leaf-node, choose one of the
following:
                1. SHOWRESULTS I(n)
                2. IGNORE I(n)
                3. S← EXPAND I(n)
                        For each ni in S
                        EXPLORE (I(ni))
        else, choose one of the following:
// n  is a leaf node
              1. SHOWRESULTS I(n)
                2. IGNORE I(n)
```

Fig 2. TOPDOWN Navigation Model

2.2 TOPDOWN Cost Model

The cost model, which is inspired by a previous work [2], takes into consideration the number of concept nodes revealed by an EXPAND action, the number of EXPAND actions that the user performs and the number of citations displayed for a SHOW RESULTS action. In particular, the cost model assigns

□Cost of 1 to each newly revealed concept node that the user examines after an EXPAND action.

□Cost of 1 to each EXPAND action the user executes.

□Cost of 1 to each citation displayed after a SHOWRESULTS action.

Since the exact sequence of actions of a user cannot be known *a priori*, *estimate* the cost based on the following two probabilities:

- EXPLORE probability P_E *(I(n))* is the probability that the user is interested in the component subtree *(I(n))* and will hence explore it. The IGNORE probability is 1 - P_E *(I(n))*
- EXPAND probability P_C *(I(n))* is the probability that the user executes an EXPAND action on component Subtree *(I(n))*. The SHOWRESULTS probability is 1 - P_C *(I(n))*

The cost of exploring a component sub tree I(n) rooted at node n is ,

- *Cost(I(n))* = P_E^N *(I(n)).* *(1-Pc(I(n)).|L(I(n))|* + *Pc(I(n)).(B+|S|* + *Σs*∈S cost(Ic(S)))),

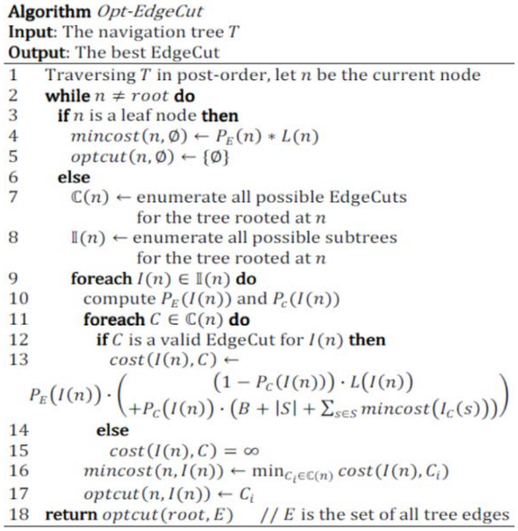
where P_E^N *(I(n))* is the normalized P_E *(I(n))*, such that the sum of P_E^N ’s of the component subtrees after an EdgeCut equals 1. P_E^N of the original tree is 1. The intuition for this normalization is that the probability that the user wants to explore a node *n* should not depend on the specific expansions sequence that revealed *n*.

3. Algorithms For Best Edge Cut

Given the cost equation in Section 2.2, we can compute the optimal cost by recursively enumerating all possible sequences of valid EdgeCuts, starting from the root and reaching every concept in the navigation tree, computing the cost for each step and taking the minimum. However, this algorithm is also prohibitively expensive. Instead we propose an alternative algorithm, Genetic algorithm based *Best-EdgeCut* that makes use of the genetic algorithm technique to reduce the computation cost.

3.1 Opt Edge Cut Algorithm

The Opt-Edge Cut algorithm shown in fig. 3 which is responsible to calculate the minimum expected navigational cost.



```
Algorithm Opt-EdgeCut
Input: The navigation tree $T$
Output: The best EdgeCut
Traversing $T$ in post-order, let $n$ be the current node
while $n \neq root$ do
  if $n$ is a leaf node then
    $mincost(n, \emptyset) \leftarrow P_E(n) * L(n)$
    $optcut(n, \emptyset) \leftarrow \{\emptyset\}$
  else
    $\mathbb{C}(n) \leftarrow$ enumerate all possible EdgeCuts
                for the tree rooted at $n$
    $\mathbb{I}(n) \leftarrow$ enumerate all possible subtrees
                for the tree rooted at $n$
    foreach $I(n) \in \mathbb{I}(n)$ do
      compute $P_E(I(n))$ and $P_c(I(n))$
      foreach $C \in \mathbb{C}(n)$ do
        if $C$ is a valid EdgeCut for $I(n)$ then
          $cost(I(n), C) \leftarrow$
  $P_E(I(n)) \cdot \begin{pmatrix} (1 - P_C(I(n))) \cdot L(I(n)) \\ + P_C(I(n)) \cdot (B + |S| + \sum_{s \in S} mincost(I_C(s))) \end{pmatrix}$
        else
          $cost(I(n), C) = \infty$
      $mincost(n, I(n)) \leftarrow \min_{C_i \in \mathbb{C}(n)} cost(I(n), C_i)$
      $optcut(n, I(n)) \leftarrow C_i$
return $optcut(root, E)$    // $E$ is the set of all tree edges
```

Fig. 3 Opt-EdgeCut Algorithm [9]

3.2 Genetic Algorithm Based Best Edge Cut Algorithm

The algorithm proposed in fig. 3 is more expensive in terms of computational cost. To overcome this drawback, genetic algorithm based Best Edge cut algorithm is proposed. Genetic Algorithms (GA) is direct, parallel, stochastic method for global search and optimization. In GA, the search space is composed of candidate solutions to the problem. After the initial population is generated randomly, selection and variation function are executed in a loop until some termination criterion is reached. Each run of the loop is called a generation.

The selection operation is intended to improve the average quality of the population by giving individuals or subtrees of higher quality a higher probability to be copied into the next generation. The quality of the subtree is measured by a fitness function.

In crossover the subtrees chosen by selection, recombine with each other and new subtrees will be created. The aim is to get subtrees that inherit the best possible combination of the characteristics of their parents. After a crossover is performed mutation takes place. This is to prevent falling of all solutions in a population into a local optimum of solved problems. Mutation randomly changes the new subtrees.

```
Genetic Algorithm Based Best Edge Cut Algorithm
Input: User Query
Output : Best Edge Cut
    1. Generate Tree T corresponding to
       user query ( hierarchical result )
    2. Traversing the tree in post order ,
       here n is the current node
           a) If n is not leaf node
              GeneticEdgeSet(T,n)
    3. Select best edge set from trees
       generated from genetic algorithm
```

```
GeneticEdgeSet(T,n)   // (return tree
    set)
Here N – number of subtrees in a sequence
R- Number of rounds
    1. Set S= Initial Generation()
    2. For i=1 to R
         S=Selection ( S)
         S= Recombination (S)
         S= Mutation ( S)
         S= SelectBest ( S)
         End For
    3. Return S
```

In Initial Generation(), generate 'N' subtrees from T

➢ Selection (N subtrees)

Input : N Subtrees
Output : 2N subtrees

- Copy N (input) subtrees to output. Remaining N subtrees are generated as
    1. Select two subtrees from input randomly, copy best (with highest fitness value) to output
    2. Repeat 'step1' N times

➢ Recombination:

Input : 2N Subtrees
Output : 2N subtrees

- Copy first N input trees to output, remaining N trees are generated as
    1. Randomly select two trees X, Y from input and generate a new tree Z by the following
        1. Combine X and Y
        2. Sort the nodes according to the weight
        3. Select the first x nodes where x is the number of nodes in X
    2. Repeat 'step1' N times

➢ Mutation

Input : 2N Subtrees
Output : 2N subtrees

- Copy first N input trees to output, remaining N trees are generated as
    1. Randomly select a tree X from input
    2. Find the minimum weighted node in X and interchange it with any highest weighted node from the remaining subtrees which should not be in X and add the new one to output
    3. Repeat 'step1-2' N times

➢ Select Best

Input : 2N Subtrees
Output :2N subtrees

1. Calculate fitness values of each subtree
2. Arrange the subtrees in the descending order according to their fitness value.
3. Sort the nodes according to the weight and select the first subtree to be displayed

➢ Fitness Value Of Tree with Root Node n

FV = p-expand (n)+ p-showresults(n) + p-ignore (n)+p-backtrack(n)

4. RESULT ANALYSIS

For evaluating the proposed application, expansion time performance and average navigation cost are considered. The empirical studies are made in a PC with XP as operating system. MySQL is used as backend and Java is used to implement all algorithms. The proposed application achieves improvement in navigation cost when compared with the Opt-EdgeCut Algorithm.

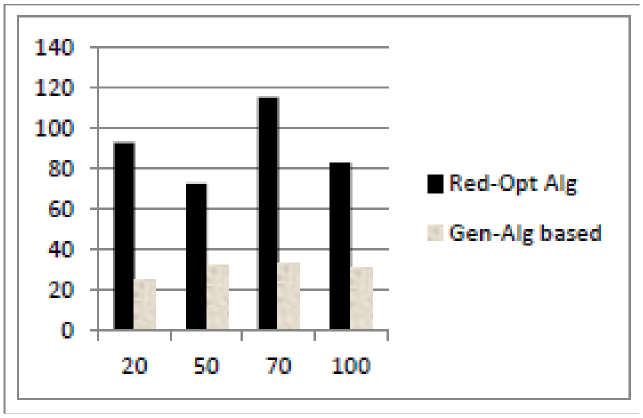


Fig. 4. Overall navigation cost comparison

As can be seen in fig. 4, queries have been presented for two types of algorithms. The X axis takes the number of rounds the algorithms are executed while the Y axis represents the execution time. Genetic Algorithm based Best Edge Cut Algorithm, leads to considerably smaller navigation cost for a set of real queries on the MEDLINE database. These experiments were executed on a reduced navigation tree (20 nodes), constructed from the original query navigation tree for each query, since Opt-EdgeCut is prohibitively expensive for most navigation trees. Finally, show that the execution time of Genetic Algorithm based Best Edge Cut Algorithm is small enough to facilitate interactive time use navigation.

5. Conclusion

This paper focuses on developing a framework that facilitates rich user experience while navigating query results. This framework is required as the large medical databases available over Internet return huge number of records as results. Navigating the results to obtain required information causes wastage of user's time. The proposed framework can effectively reduce the navigation cost and provide easy access to required results. Thus it can solve information overload problem effectively. The proposed system 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. A new cost model is proposed to decide which concepts to display at each step, which reduces the navigation cost as it can eliminate unnecessary navigation steps. A prototype application is built with web based interface, which facilitates users to search for required information and also navigate the results. The experimental results revealed that the user navigation cost is reduced substantially and rich user experience is achieved.

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] J.A. Mitchell, A.R. Aronson and J.G. Mork: *Gene Indexing: Characterization and Analysis of NLM's GeneRIFs*. In Proceedings of the AMIA Symposium, 8th–12th November, Washington, DC, pp. 460–464

[5] Vivísimo, Inc. –Clusty. (2008) [Online].Available: http://clusty.com/

[6] A. Kashyap, V. Hristidis, M. Petropoulos, and S. Tavoulari: *BioNav: Effective Navigation on Query Results of Biomedical Databases*. (Short Paper), ICDE 2009, to appear. Available at http : // www. cs. Fiu .edu /vagelis /publications/BioNavICDE09.pdf

[7] HON (2010): Health On the Net Foundation: Medical information. Available online at: http://www.hon.ch/cgi-bin/HONselect?cat+A [viewed: 15 August 2012]

[8] Medical Subject Headings (MeSH), http: //www.nlm.nih.gov/ mesh/, 2010.

[9] Abhijith Kashyap, Vagelis Hristidis, Michalis Petropoulos, and Sotiria Tavoulari (2011), "Effective Navigation of Query Results Based on Concept Hierarchies".IEEE Transactions On Knowledge And Data Engineering, VOL. 23, NO. 4.

[10] S. Kundu and J. Misra, "A Linear Tree Partitioning Algorithm," SIAM J. Computing, vol. 6, no. 1, pp. 151-154, 1977.

[11] D. Maglott, J. Ostell, K.D. Pruitt and T. Tatusova: Entrez Gene:Gene-Centered Information at NCBI. Nucleic Acids Res. 2005 January 1; 33(Database Issue): D54–D58

[12] (2008) OMIM - Online Mendelian Inheritance in Man. [Online]. Available: http://www.ncbi.nlm.nih.gov/Omim/