Enhancing Internet Search Engines to Achieve Concept-based Retrieval

Fenghua Lu[1], Thomas Johnsten[2], Vijay Raghavan[1] and Dennis Traylor[3]

[1]Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, LA 70504, USA

[2]Department of Computer Science
University of Southwestern Louisiana
Lafayette, LA 70504, USA

[3]Office of Scientific and Technical Information
Department of Energy
Oak Ridge, TN 37831, USA

Abstract: Most engines used for searching information resources via the Internet employ the Boolean Retrieval Model. Two main drawbacks of this model are that users have difficulty to precisely formulate their concept (or, topic) of interest using Boolean logic and the resulting output is not ranked. We propose to address both these problems by employing a Concept-based Retrieval Model, where a concept is defined by a set of production rules and the rule-base is represented as a rule-base tree. Features of a prototype developed at USL, referred to as the Concept-Set Structuring System (CS^3), which includes a graphical interface for defining and refining rule-base trees and for converting them into equivalent sets of conjunctions, called Minimal Term Sets (MTSs), are described. By submitting MTSs generated for a concept to an existing search engine and by reordering the returned results according to the importance of MTSs they satisfy, the CS^3 prototype enhances the capabilities of the underlying search engine. Results that demonstrate the use of the prototype, coupled with DOE Information-Bridge, will be presented.

I. INTRODUCTION

The Internet has become a popular medium for the exchange of information and ideas among various groups of individuals. One group in particular that has become increasingly dependent upon the Internet for the timely exchange of research ideas and results is the scientific community. For instance, it is not uncommon for a scientist to regularly retrieve, through the Internet, both pre-print and printed scientific reports and papers. The Internet's increased role in the dissemination of scientific, as well as non-scientific, information has heightened the need for the design and implementation of efficient and effective Internet search engines.

The design of most commercially developed Internet search engines is based on the Boolean Retrieval Model [1,2]. A system that adheres to this model supports the formulation of search requests by combining individual search terms with the Boolean operators AND, OR and NOT.

The occurrence of a Boolean operator within a search request is typically processed through the application of the corresponding AND, OR or NOT set operator. For example, a Boolean retrieval system would process the search request, “data AND mining”, by applying the intersection operator to the sets *A* and *B*, where *A* and *B* are comprised of documents from the system's database that contain the index terms “data” and “mining”, respectively. In other words, a document is retrieved if and only if it belongs to the intersection of sets *A* and *B*. A characteristic of Boolean retrieval systems is that the rank order of retrieved documents is arbitrary since a document either satisfies, or does not satisfy, a user's search request.

A Boolean retrieval system, although simple to design and implement, has a number of well documented shortcomings [2]. These shortcomings include the inability to allow a user to assign weights of importance to terms within a search request or a document, and the inability to rank a list of retrieved documents based on the documents' estimated degrees of usefulness to the user. Unfortunately, these shortcomings are amplified in the context of the Internet since the volume of searchable information is extremely large. The consequence, in general, of using a Boolean retrieval system to search the Internet in order to satisfy a specific information need is that the results are either too narrow or too broad.

In response to these noted shortcomings, various extensions have been suggested to the Boolean Retrieval Model [2]. Of special interest to the current work is the utilization of a user or expert defined knowledge base. The application of expert system technology was first explored in the context of an information retrieval system referred to as the RUBRIC [3]. The novelty of RUBRIC is its ability to support a user-defined rule-base that is used in formulating requests at the users' conceptual level. A rule-base is a mechanism for representing a set of concepts and the relationships among the concepts. The rule-base allows concept-based retrieval whereby a query expressed at the conceptual level is translated into a series of Boolean expressions that can subsequently be processed by one or more underlying search engines. This enhancement is aimed at alleviating the previously stated shortcomings of the Boolean Retrieval Model. In this paper, we describe the design and implementation of a prototype system for concept-based retrieval that is aimed at improving the performance of search engines accessible over the Internet. Specifically, we have developed a prototype Concept-Set Structuring System, referred to as CS^3, that is able to interface with existing Boolean retrieval search engines. We present its features by using it with the U.S. Department of Energy's “Information-Bridge” retrieval system [4]. To that end, the remainder of this paper is organized as follows. Section two presents an overview of concept-based retrieval and its framework in the RUBRIC retrieval system. Section three describes the design and implementation of the CS^3 prototype. Section four presents the results of a sample concept for which searches were conducted using CS^3 and the "Information-Bridge" system. Section five summarizes the work presented in this paper and outlines some of our future research plans.

II. CONCEPT-BASED RETRIEVAL

The key feature of concept-based retrieval is its support for search requests that are formulated by means of concepts structured as a rule-base tree. The significance of this feature is that concepts of interest are formulated using a *top-down* refinement strategy. In a top-down strategy, the first step is to express a given request as a single concept. The stated concept is intended to represent the search request at a very abstract level. The next step is to refine the initial concept by decomposing it into a set of component parts that are related through either the AND or OR logical operator. The individual components may take the form of a new concept defined at a different abstraction level, a text expression, or a single index term. In each case, a weight value is assigned to the individual *concept-component* pairs that are formed during the decomposition process. The assigned weight value represents the user's belief in the degree to which a given component characterizes the related concept. The process of decomposing concepts into component parts is repeated until a level of abstraction is reached in which every terminal node in the constructed rule-base tree represents an index term or a text expression.

Figure-1 shows the rule-base tree that has been constructed for the user defined concept "Human-Health-Science" where the leaf nodes are index terms and are enclosed within double quotations, the internal nodes are concepts, and the weight values are whole numbers displayed along the edges connecting concepts and components. The "Human-Health-Science" concept is first decomposed into two component parts, "Human" and "Health-Science", which are related to the initial concept by means of the AND operator. In this particular case, both the "Human" and "Health-Science" components represent concepts, but are defined at an abstraction level below the "Human-Health-Science" concept. In turn, the two component concepts, like the initial concept, have also been decomposed into component parts. As shown in Figure-1, the components of the "Human" concept are given as the index terms "Man", "Woman" and "Human"; and, the components of the concept “Health-Science” are given as the concepts “Health-Effects”, “Biological-Effects”, “Risk-Assessment” and “Molecular and Genomic Science”. The decomposed components of the “Human” concept represent index terms, and thus are not decomposed further. However, the components of the “Health-Science” concept represent concepts themselves, and thus are decomposed into component parts. The decomposition, or refinement, process continues until every terminal node represents an index term. In the current example, this condition is reached following the decomposition of the concepts “Health-Effects”, “Biological-Effects”, “Risk-Assessment” and “Molecular and Genomic Science”.

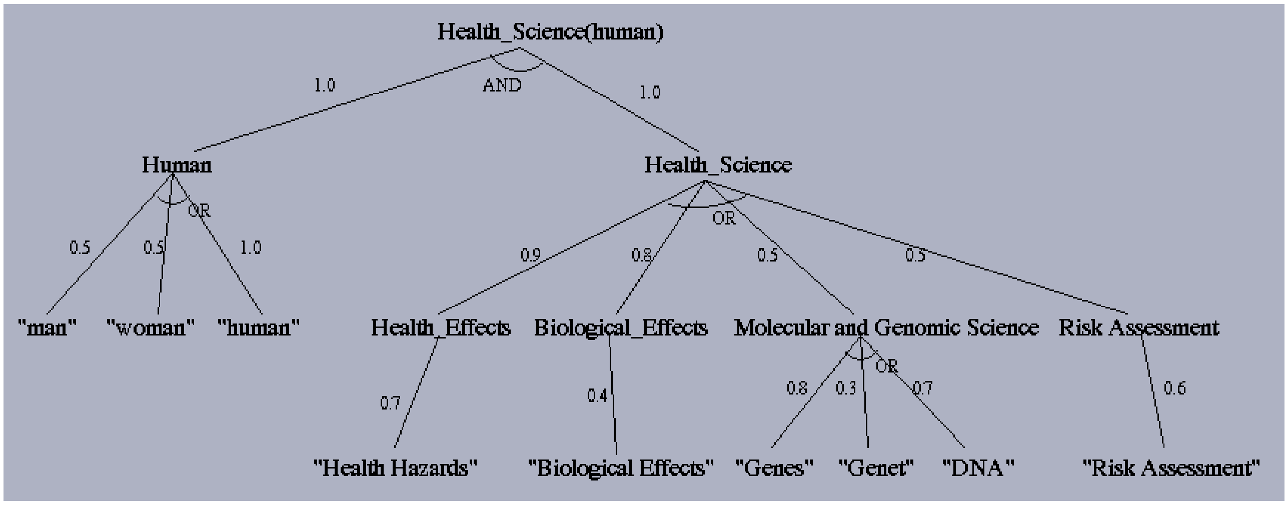


Figure-1 “Human-health-Science” rule-based tree.

The evaluation of a user-defined concept requires an analysis of the given concept's rule-base tree. In the case of RUBRIC the evaluation process is based upon a “run-time”, bottom-up analysis of a rule-base tree. The term “run-time” is used here to denote the fact that the analysis of a rule-base tree occurs only during its application against a document database. In other words, there is no *static* analysis of a rule-base tree, for efficiency reasons, prior to its application against a document database.

The actual run-time evaluation of a rule-base tree begins with the matching of each document D in the given database against the index terms that occur in the given tree. Given a document D, if an index term that occurs in D is present in the rule-base tree, then the term is assigned a weight value of one; otherwise, it is assigned a weight value of zero. The assigned weight values are subsequently propagated upward in the rule-base tree. In general, the propagation of such values is governed through the application of the following two rules. In the case of an AND relationship the assigned weight value is determined by the expression $min\ \{component\_wt_{ik} * component\_concept\_wt_{ik}\}$; and, in the case of the OR relationship the assigned weight value is determined by the expression $max\ \{component\_wt_{ik} * component\_concept\_wt_{ik}\}$. The two quantities $component\_wt_{ik}$ and $component\_concept\_wt_{ik}$ represent the weight value associated with component i of concept k and the weight value associated with the concept$_k$-component$_i$ pair, respectively. The upward propagation of weight values continues until the concept within the given hierarchy that represents the user's current information needs is itself assigned a weight value.

To illustrate the run-time evaluation of a rule-base tree consider the tree in Figure-1 and a document D whose index terms include, “human”, “Genes”, “DNA”, and “Risk Assessment”. Figure-2 shows, in parenthesis, the weight value assigned to each component in the rule-base tree as a result of the upward propagation of the term weights. The process begins with the assignment of weight ‘1’ to the leaf nodes corresponding to the index terms included in D. The method of weight propagation defined in the previous paragraph is applied to determine the importance of document D to the intermediate nodes. The weight value assigned to the root concept, or to the user's concept of interest, represents the current document's Retrieval Status Value (RSV). For example, the RSV of document D in the above example is 0.8. In general, a RSV is an estimate of a document's anticipated usefulness to the user as determined by the retrieval system [1]. The use of RSVs allows RUBRIC to rank documents in a non-increasing order of anticipated usefulness to the user.

In the next section, we describe how concept-based retrieval has been implemented in our

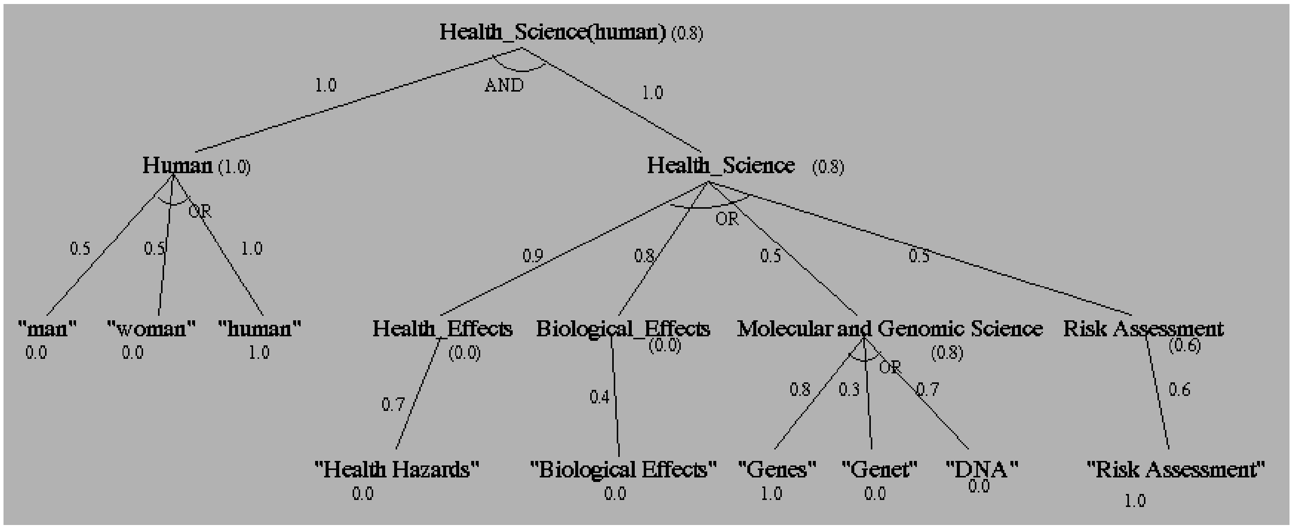


Figure-2 Initial propagation of weight values in Fig.1

prototype system, CS^3, and illustrate how the developed system is able to interface with existing Boolean retrieval search engines.

III. THE CONCEPT-SET STRUCTURING SYSTEM PROTOTYPE

The goal of the Concept-Set Structuring System (CS^3) prototype is to improve the retrieval performance of search engines accessible over the Internet. In order to achieve this goal, the CS^3 prototype has been designed to support both the development and evaluation of concepts that are structured as a rule-base tree.

A. Development of a Rule-Base Tree

The CS^3 prototype provides a number of important features that are designed to facilitate the development and management of concepts. These features include support for the creation and storage of new concepts as well as support for the modification of existing concepts. Figure-3 shows a snapshot of the user-interface that has been implemented as part of the CS^3 prototype. The majority of the control buttons and input fields shown in Figure-3 are directly related to the process of creating and managing concepts. For instance, the construction of a new concept begins with the specification of the root concept's name in the “new topic name” input field. The components of a given concept are inserted into the hierarchy by first selecting the related concept in the “display window”, and then entering the names of the components and their corresponding weight values into the “new topic name” and the “weight value” input fields, respectively. The insertion of an index term (i.e. a leaf node) is distinguished from the insertion of a new concept by the selection of the “term” option from the “new topic type” list-box. The relationship between components and a related concept is specified through the selection of either the “OR” or “AND” option that is displayed in the “new topic type” list-box. A constructed rule-base tree can be permanently stored for future use by assigning it a name through the “rule tree name” input field. The ability to save constructed rule-base trees allows individual users to create both private and public libraries of stored concepts.

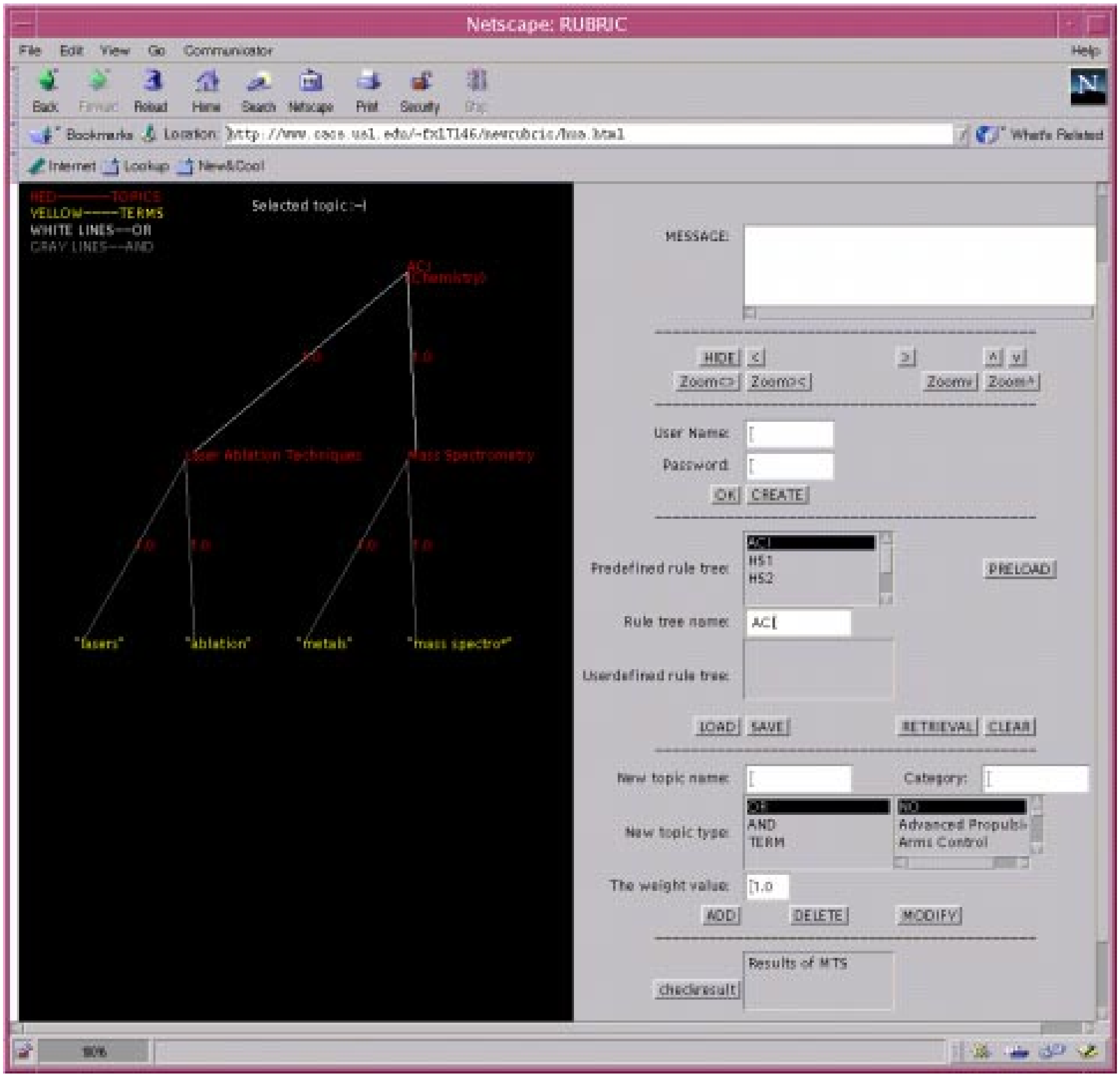


Figure-3 CS^3 User Interface

B. Evaluation of a Rule-Base Tree

The CS^3 prototype evaluates a given concept through the construction of a Minimal Term Set (MTS) [5]. The construction of a MTS, like the run-time evaluation procedure utilized by RUBRIC, requires a bottom-up evaluation of a rule-base tree. However, the actual construction of a MTS occurs through the use of a static analysis step prior to the application of a given concept against a document database. The benefit of such a step is that the propagation of weight values is performed only once instead of for each document in the database. The construction of a MTS involves the upward propagation of the index terms specified in the given rule-base tree. This action results in the replacement of concepts at various levels with a set of weighted logical expressions that are comprised of propagated index terms. The constructed expressions take the form of a logical conjunction (disjunction) of terms whenever the relationship between a given set of terms and a related concept is defined with respect to the AND (OR) operator. The specific weight value that is assigned to an expression is based upon the same propagation rules defined previously in the context of RUBRIC. The assumption is typically made that all index terms have a standard component weight value of one. The upward propagation of the index terms represents the first step in the construction of a MTS. Figure-4 shows the result of the initial upward propagation of the index terms given in Figure-1. Notice that the lowest level concepts in Figure-1 ("Human", "Health-Effects", "Biological-Effects", "Risk-Assessment" and "Molecular and Genomic Science") have all been replaced by sets of weighted logical expressions that are composed of the index terms given in the rule-base tree.

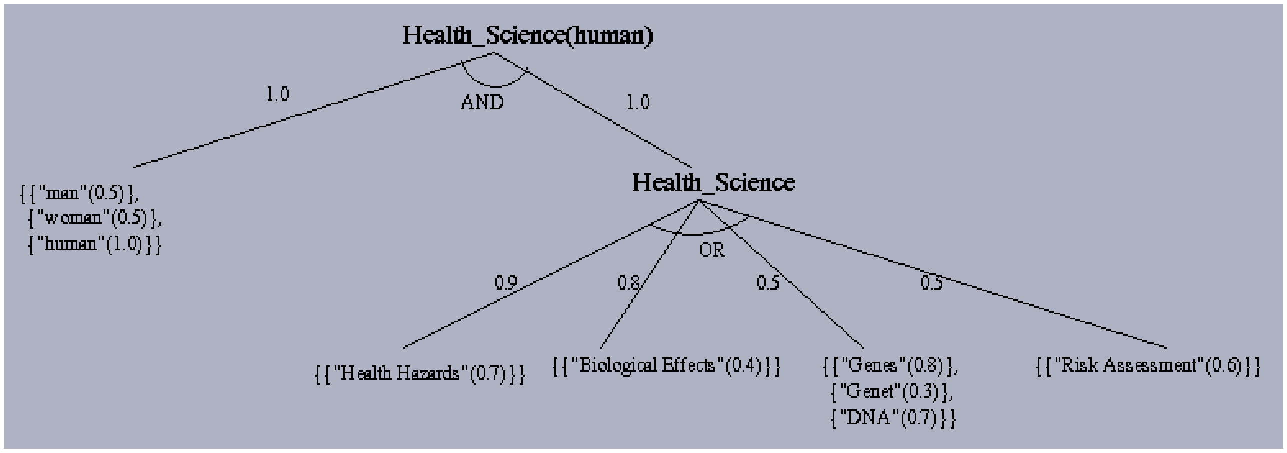


Figure-4 Initial propagation of index terms in Fig.1

The next step is to propagate upward the previously constructed logical expressions. Specifically, the constructed expressions corresponding to components combined by the AND operator are propagated upward as a single search expression, while the expressions of components combined by the OR operator are propagated upward as multiple search expressions. In the latter case, each operand represents a distinct search expression and has its own assigned weight value. For example, the result of the upward propagation of the initial expressions constructed in Figure-4 is shown in Figure-5. In general, the propagation of logical expressions continues until the concept within the given hierarchy that represents the user's current information needs is itself represented as a set of weighted expressions. The final constructed set of expressions represents the MTS of the given concept. The MTS corresponding to the concept “Human-Health-Science” in Figure-1 is shown in Figure-6.

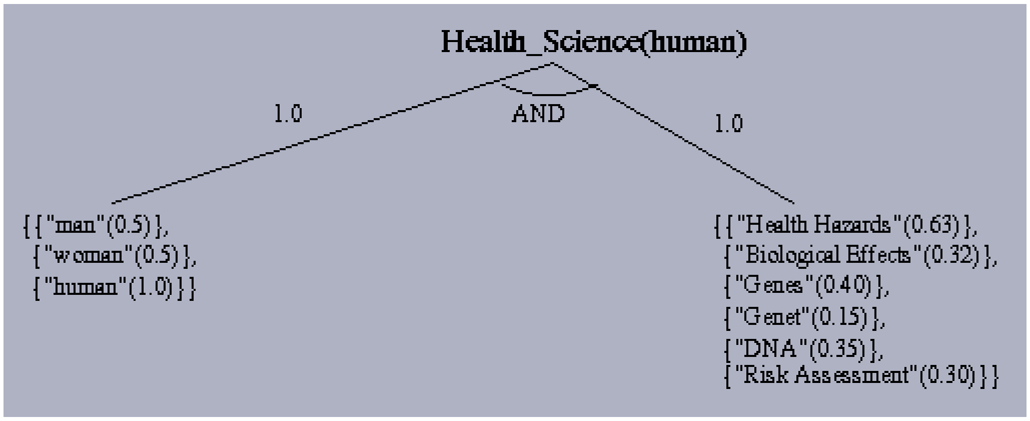
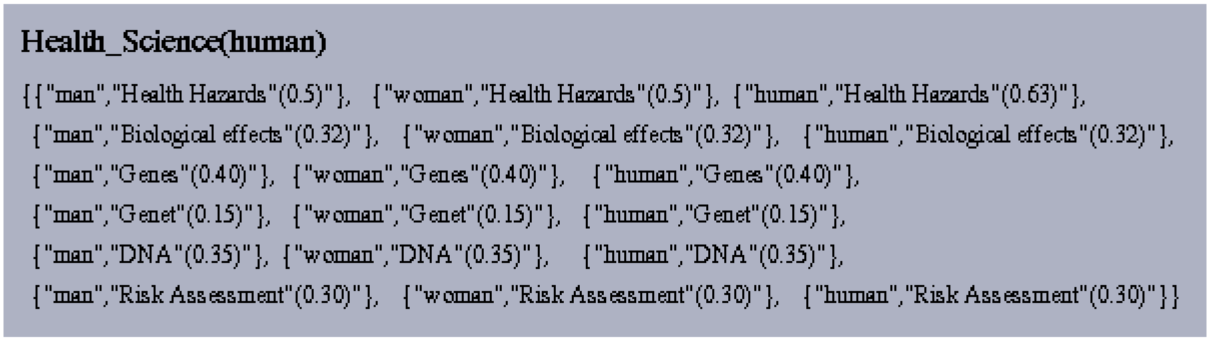


Figure-5 Propagation of search expressions constructed in Fig.4


Health_Science(human)

{{"man","Health Hazards"(0.5)"}, {"woman","Health Hazards"(0.5)"}, {"human","Health Hazards"(0.63)"},
{"man","Biological effects"(0.32)"}, {"woman","Biological effects"(0.32)"}, {"human","Biological effects"(0.32)"},
{"man","Genes"(0.40)"}, {"woman","Genes"(0.40)"}, {"human","Genes"(0.40)"},
{"man","Genet"(0.15)"}, {"woman","Genet"(0.15)"}, {"human","Genet"(0.15)"},
{"man","DNA"(0.35)"}, {"woman","DNA"(0.35)"}, {"human","DNA"(0.35)"},
{"man","Risk Assessment"(0.30)"}, {"woman","Risk Assessment"(0.30)"}, {"human","Risk Assessment"(0.30)"}}

Figure-6. Minimal Term Set for the concept “Human-Health-Science”

Each MTS that is constructed by CS^3 is displayed in the list-box shown to the right of the “check result” button. The displaying of MTSs in a list-box allows for the submission of search requests with respect to all or a subset of the constructed MTS expressions. A search request, based upon a user-selected subset of MTS expressions, is initiated through the selection of the “retrieval” control button.

C. Initiation of a Retrieval Search

The selection of the “retrieval” control button causes the CS^3 prototype to initiate a document retrieval search through the execution of an existing Boolean search engine. In general, each selected MTS expression is passed to the targeted search engine for evaluation; and, the results produced by the search engine are sent back to the CS^3 prototype. The processing of a MTS is equivalent to the evaluation of a series of search requests specified in the context of the Boolean Retrieval Model. Each expression contained within a MTS represents a single Boolean search request. The most significant difference with respect to the processing of a Boolean search request is the assignment of a RSV to the documents belonging to the system's database. Currently, the CS^3 prototype calculates a document's RSV as follows: given a document D and a set S of MTS expressions satisfied by D, the RSV of D is equal to the sum of all the weights of S plus the maximum weight in S. For example, suppose that a document D satisfies a set of MTS expressions having the following assigned weight values, 0.2, 0.5 and 0.75. In this instance, the RSV of document D is equal to 2.2. This particular RSV measure is characterized by the fact that it gives higher preference to documents that satisfy fewer MTS expressions with each MTS having a relatively large weight value as compared to documents that satisfy several MTS expressions having relatively small assigned weight values.

After receiving all of the results from the evaluated MTS expressions, the CS^3 prototype displays the returned documents in rank order based upon each document's assigned RSV. In the event that two or more returned documents have identical RSVs, the documents are ranked in reverse chronological order.

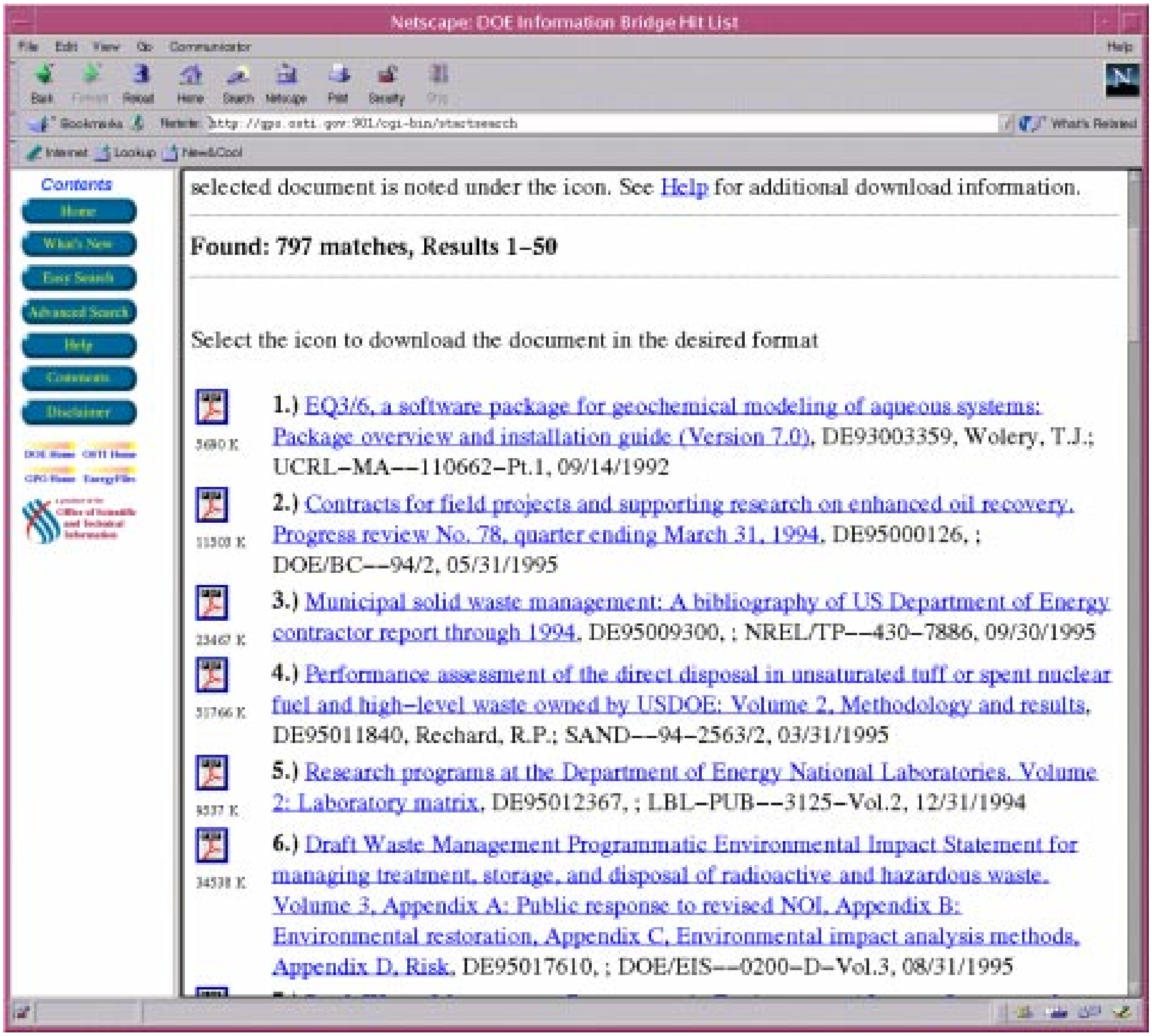
D. Interface with Boolean Search Engines

The interface between the CS^3 prototype and an existing search engine is controlled through the execution of a cgi-script that has been implemented in Java, due to Java’s extensive support of various network services. The script is designed to determine the parameters required by the selected search engine, receive and process the output results produced by the selected search engine, and send the final processed results back to the web browser.

The CS^3 prototype is currently able to interface with two existing Internet search engines, the U.S. Department of Energy's “Information-Bridge” system and the U.S. Department of Transportation's “National Transportation Library” system [4,6]. In the former case, the cgi-script interfaces directly with the “Advanced Search” input form [6]. This particular form has four general categories of input fields, “Field”, “Pick List”, “Word or Phrase” and “Operator”. The cgi-script’s current implementation utilizes only the “Field” and the “Word or Phrase” input fields. Specifically, we have designed the script to select the “OCR Text or Biblio” option associated with the “Field” list-selection box, and to insert a selected MTS expression into the “Word or Phrase” input field. As a result, the subsequent submission of the “Advanced Search” form results in the retrieval of those documents in the database that satisfy the MTS expression that has been inserted into the “Word or Phrase” input field. Following the processing of all the user-selected MTS expressions, the cgi-script calculates each document’s RSV and produces a rank order of the documents based upon their calculated RSV.

IV. COMPARISON OF INFORMATION-BRIDGE AND CS^3

In this section we compare the results produced by the CS^3 prototype with the results produced by the Department of Energy's Information-Bridge retrieval system. The comparison was made based upon a request that was formulated with respect to a subset of the subject categories and sub-categories listed on the Department of Energy’s “Environmental Science Network” search form [7]. The selected subject categories included the general category “Hydrogeology” along with the related sub-categories, “Dnapl Dynamics” and “Fluid-flow and Colloidal Dynamics”. In response to the selected categories, the following Boolean search request, (“Hydrogeology” OR “Dnapl” OR (“Colloid*” AND “Environmental Transport”)), was generated by the Information-Bridge retrieval system. The application of this particular Boolean request resulted in the retrieval of several hundred documents. Figure-7 shows the first six documents that were retrieved by the system. Of course, the retrieved documents have no associated RSV as a result of the direct use of the Boolean Retrieval Model. The lack of RSVs makes it impossible to specify a top n-ranking of the retrieved documents based upon their anticipated usefulness.



http://gps.osti.gov:901/cgi-bin/startsearch

selected document is noted under the icon. See Help for additional download information.

Found: 797 matches, Results 1–50

Select the icon to download the document in the desired format

5690 K **1.)** EQ3/6, a software package for geochemical modeling of aqueous systems: Package overview and installation guide (Version 7.0), DE93003359, Wolery, T.J.; UCRL–MA––110662–Pt.1, 09/14/1992

11503 K **2.)** Contracts for field projects and supporting research on enhanced oil recovery. Progress review No. 78, quarter ending March 31, 1994, DE95000126, ; DOE/BC––94/2, 05/31/1995

23467 K **3.)** Municipal solid waste management: A bibliography of US Department of Energy contractor report through 1994, DE95009300, ; NREL/TP––430–7886, 09/30/1995

31766 K **4.)** Performance assessment of the direct disposal in unsaturated tuff or spent nuclear fuel and high–level waste owned by USDOE: Volume 2, Methodology and results, DE95011840, Rechard, R.P.; SAND––94–2563/2, 03/31/1995

8537 K **5.)** Research programs at the Department of Energy National Laboratories. Volume 2: Laboratory matrix, DE95012367, ; LBL–PUB––3125–Vol.2, 12/31/1994

34538 K **6.)** Draft Waste Management Programmatic Environmental Impact Statement for managing treatment, storage, and disposal of radioactive and hazardous waste. Volume 3, Appendix A: Public response to revised NOI, Appendix B: Environmental restoration, Appendix C, Environmental impact analysis methods, Appendix D, Risk, DE95017610, ; DOE/EIS––0200–D–Vol.3, 08/31/1995

Figure-7. Results produced by the Information-Bridge System.

A rule-base tree was generated using the CS^3 prototype with respect to the previously selected subject category and sub-categories. A constructed rule-base tree that includes the given category and sub-categories is shown in Figure-8. Each of the *concept-component* edges in Figure-8 has been assigned a weight value that reflects the user's belief in the degree to which a component characterizes a related concept. The evaluation of the concept "Hydrogeology" produced a corresponding MTS that was subsequently evaluated through the execution of the Information-Bridge retrieval system. The produced results consisted of the same documents that were previously retrieved via the direct use of the Information-Bridge system. However, in the latter case the retrieved documents were ranked by the CS^3 prototype in non-increasing order of their assigned RSV, and for a given RSV the documents were listed in reverse chronological order. Figure-9 shows the top six documents that were retrieved by the CS^3 prototype.

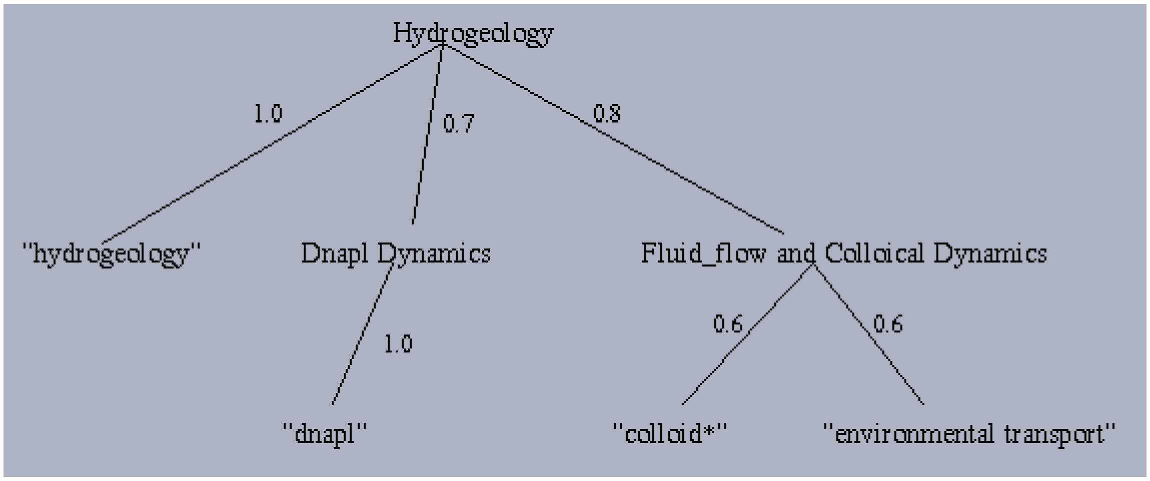
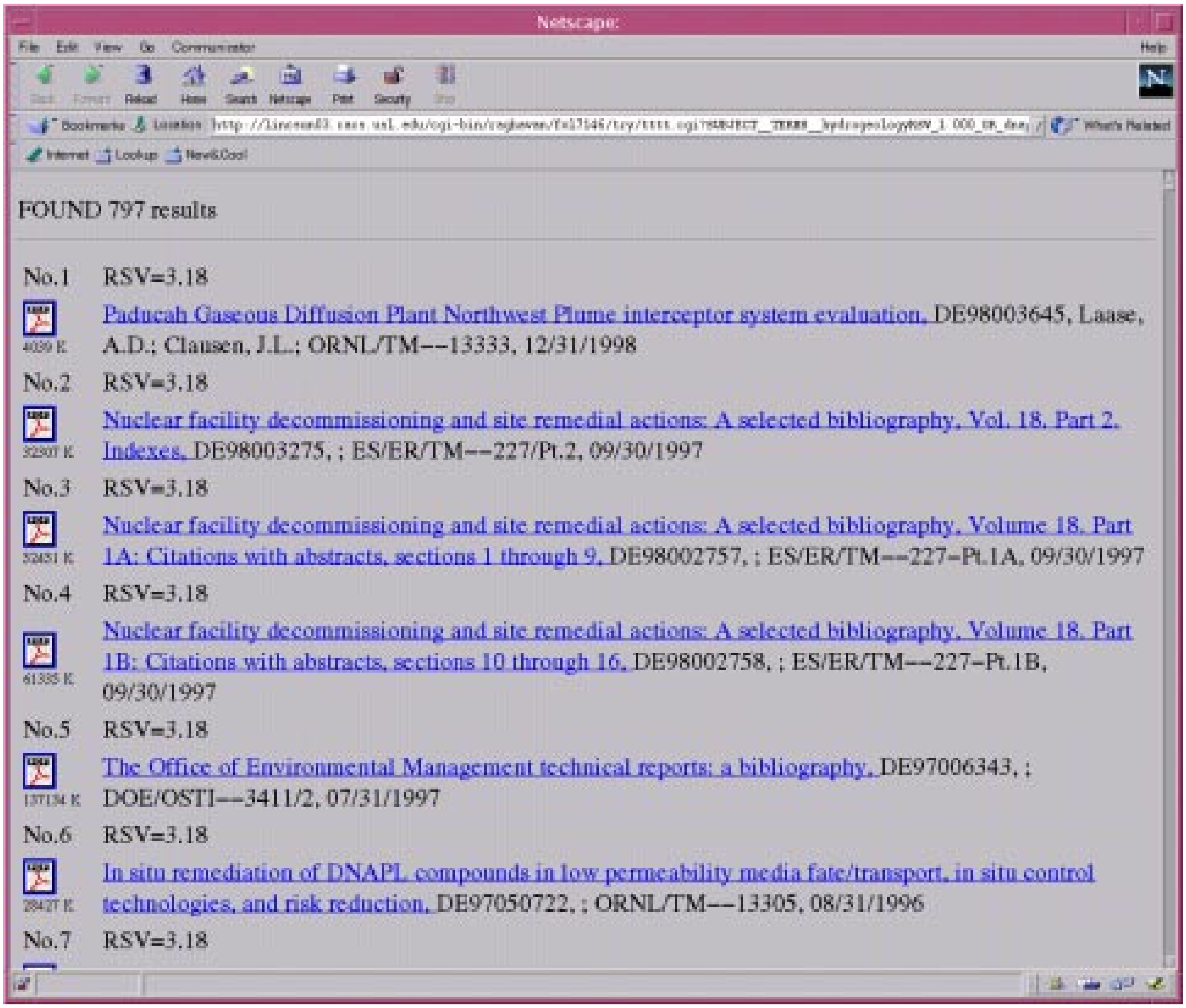


Figure-8. “Hydrogeology” rule-base tree

V. CONCLUSION AND FUTURE WORK

This paper has reported on the development the Concept-Set Structuring Subsystem (CS^3) which is a prototype system that allows for the formulation of search requests specified in terms of concepts that are structured as a rule-base tree. The utilization of the concept-based approach alleviates certain shortcomings of the Boolean Retrieval Model. In particular, the CS^3 prototype addresses the Boolean Model's inability to allow a user to assign weights of importance to document or query terms and the inability to rank a list of retrieved documents based on the documents' estimated degrees of usefulness to the user. The current design of the CS^3 prototype provides an environment that supports both the creation and evaluation of user defined concepts. A key feature of the prototype is its ability to interface with existing Boolean search engines. This particular feature allows for the potential use of the CS^3 prototype in conjunction with nearly Boolean search engine accessible via the Internet.

We have several research projects planned with respect to concept-based retrieval and the CS^3 prototype. Our immediate plans include, conducting a large scale experiment to evaluate the retrieval effectiveness of the concept-based approach as implemented by the CS^3 prototype, investigating the use of alternative methods to compute RSVs within the context of concept-based retrieval, and introducing the process of relevance feedback within the context of concept-based retrieval.



FOUND 797 results

No.1 RSV=3.18

Paducah Gaseous Diffusion Plant Northwest Plume interceptor system evaluation, DE98003645, Laase, A.D.; Clausen, J.L.; ORNL/TM--13333, 12/31/1998

No.2 RSV=3.18

Nuclear facility decommissioning and site remedial actions: A selected bibliography, Vol. 18, Part 2, Indexes, DE98003275, ; ES/ER/TM--227/Pt.2, 09/30/1997

No.3 RSV=3.18

Nuclear facility decommissioning and site remedial actions: A selected bibliography, Volume 18, Part 1A: Citations with abstracts, sections 1 through 9, DE98002757, ; ES/ER/TM--227-Pt.1A, 09/30/1997

No.4 RSV=3.18

Nuclear facility decommissioning and site remedial actions: A selected bibliography, Volume 18, Part 1B: Citations with abstracts, sections 10 through 16, DE98002758, ; ES/ER/TM--227-Pt.1B, 09/30/1997

No.5 RSV=3.18

The Office of Environmental Management technical reports: a bibliography, DE97006343, ; DOE/OSTI--3411/2, 07/31/1997

No.6 RSV=3.18

In situ remediation of DNAPL compounds in low permeability media fate/transport, in situ control technologies, and risk reduction, DE97050722, ; ORNL/TM--13305, 08/31/1996

No.7 RSV=3.18

Figure-9 Results produced by the CS3 prototype

Acknowledgments:

This work is supported in part by a grant from the U.S. Department of Energy (under Grant No. DE-FG02-97ER1220). We acknowledge Rick Fanning and Bob Donohue for their development of the cgi-script that allows the CS^3 prototype to interface with the Department of Energy's Information-Bridge system.

Author Biography:

Fenghua Lu is a computer science M.S. student at the Center for Advanced Computer Studies (CACS) at the University of Southwestern Louisiana (USL). Tom Johnsten is a visiting assistant professor in the computer science department at USL. Vijay Raghavan is a professor of computer science at CACS. The research interests of Fenghua, Tom and Vijay are in database management, data mining, information retrieval and Internet computing. Dennis Traylor is an information scientist at the Department of Energy's Office of Scientific and Technical Information.

REFERENCES

[1] G. Salton and M. McGill, *An Introduction to Modern Information Retrieval*, New York, NY: McGraw-Hill, 1983.

[2] K. Jones and P. Willett, "Introduction to Chapter Five," in *Readings in Information Retrieval* (K. Jones and P. Willett, eds.), San Francisco, CA:Morgan Kaufmann, pp.257-263, 1997.

[3] B. McCune, R. Tong, J. Dean and D. Shapiro, "RUBRIC: A System for Rule-Based Information Retrieval," *IEEE Transactions on Software Engineering*, vol. 11:2, pp. 939-944, 1985.

[4] "U.S. Department of Energy: Information -Bridge System," http://www.doe.gov/bridge/home.html.

[5] A. Alsaffar, J. Deogun, V. Raghavan and H. Sever, “Concept Based Retrieval By Minimal Term Sets,” *International Symposium on Methodologies for Intelligent Systems*, Warsaw Poland, June , 1999.

[6] “U.S. Department of Transportation: National Transportation Library System,” http://www.bts.gov/NTL/.

[7] “U.S. Department of Energy: Environmental Science Network,” http://apollo.osti.gov:2001/.