

Towards Developing a Fuzzy Relational Medical Query System

Joseph Lee, Kristopher Zarns, Eunjin Kim, Arthi Anandhan, Jay Schell
Department of Computer Science
University of North Dakota
Box 9015, Grand Forks, ND 58202-9015
{jlee, kzarns, ejkim, arthi, schell}@cs.und.edu

ABSTRACT

In classical Boolean logic, data is represented as being either entirely true or entirely false. In most real world phenomena however, it is rare that an entity will be absolutely true or absolutely false. When designing a system, Boolean logic lacks the tools necessary to deal with this kind of uncertainty. Using fuzzy logic we were able to develop a system that is able to deal with uncertainty in a medical query environment. The system can retrieve information from a series of fuzzy relational databases to present probable solutions to a given query by the user. A brief introduction to fuzzy logic and problems involving information retrieval are also presented within this paper.

1. INTRODUCTION

Information Retrieval is the act of retrieving desired information from a data repository. In an ideal environment, a system designed to retrieve information should be able to obtain the desired information accurately to satisfy the user's particular query. In addition, the system should retrieve the information in a timely manner, and submit the results in a form that is easily understood. Modern day approaches to information retrieval have to deal with an ever growing, incredibly large depository of documents. Other hindrances in information retrieval include trying to obtain relevant outcomes from data that is imprecise, ambiguous and inaccurate. Moreover, the problem of translation from natural language to machine language, and vice versa, also exists.

Fuzzy logic is a useful tool when dealing with uncertainty within data. Unlike Boolean logic, fuzzy logic has the power to deal with data that contains partial truth and ambiguity. Human beings deal with uncertainty in their everyday language. Statements such as "He is around six feet." or "Cut the rope around three feet long." demonstrate ambiguity present in natural language. Since fuzzy logic is designed specifically to aid in data that is uncertain, it is an ideal tool for decision making involving real world applications.

There are many methods used to attempt an adequate form of information retrieval; however, the focus of this paper is to emphasize on retrieval techniques implementing fuzzy logic. Using fuzzy information retrieval we were able to develop a system which could retrieve medical information from a small relational database and display the results of the given query back to the user. The system implements a technique that allows the user to enter a specific query and then returns a list of results in a ranked order.

2. INFORMATION RETRIEVAL

Information retrieval is the process of obtaining relevant information from storage in response to a given query. Each piece of information or document is designated by a set of key words or symbols. Information retrieval deals with the task of processing the original inquiry by matching the relevant documents and delivering the results to the user. This process should deliver documents with maximum relevancy at minimal cost to the user [1].

The amount of information distributed around the world has reached staggering amounts. Sorting through large database depositories to retrieve relevant documents for a given query has become an increasingly difficult task. Factors that can hinder retrieval include problems such as incomplete information, unorganized delivery, ambiguity in natural language, and documents that contain multiple topics and subtopics. Classical methods mainly employ a crisp or Boolean system for information retrieval. Typically crisp systems only allow exact matching of queries. Problems arise from the fact that most crisp systems can miss items that do not exactly match the items being sought in the

user's original query. An alternative to crisp systems are probabilistic systems. These systems use probability based retrieval methods that rank document relevancy based on given heuristics such as term counting.

When a user enters a typical query into a search engine, the results of the search will not always contain documents that exactly match what the user is trying to obtain. Often, the documents with the most relevancies to the user's search are not located at the top of the list of retrieved documents. There are many causes for this.

A first problem a successful retrieval has to conquer is noise contained within documents. Most internet documents do not just contain information about a specific topic of interest; they also contain other information that might not be particularly interesting to the user. This can easily be attested to when visiting any standard news or information site. The site is not filled with information about the current topic exclusively but also other information such as banner ads, advertisements for other web sites, polling information, links to related topics, etc.

Another problem involving information retrieval based on a specific term is that many terms have multiple meanings. Given a certain context, the meaning of the word could change entirely [2]. Another similar problem in information retrieval is word mismatch. Word mismatch implies that if a user is searching for a term that has multiple titles, the user may miss relevant information if the term is under a different title [3]. If the user was looking for the term *reptiles*, any documents containing the word *lizard*, but not *reptile*, would not be retrieved.

It is common for documents to contain more than one topic. Documents of this nature might be skipped or missed due to their seemingly irrelevant nature. As Takaki et al. [4] state, it is important to have a mechanism for retrieving the subtopics within a document to ensure you do not accidentally pass up any documents that could be relevant to a given query. Cai et al. [5] also state the same as Takaki et al. [4] but also make reference to noise (which was mentioned earlier) and the varying length of the document.

Natural language is also a barrier in retrieving information. Zadeh [6] notes that common search engine technologies do not interpret queries well. The interpretation of natural language into machine language can create issues involved with inadequate query processing that might not satisfy the requirements of the user. Most common search methods simply search for the queried terms within the documents rather than the meaning of the entire queried phrase.

Fuzzy systems allow the user with more flexible means to retrieve items that exactly fit given queries and facilitate the retrieval of items that are closely related to the object being sought in the original query. This delivers results that are more complete and obtains information otherwise missed by crisp or probabilistic system. Fuzzy systems allow users to obtain results from databases where the information provided presents no exact matches. Our team has developed a system based on the concepts used in fuzzy systems to extract information from a small database repository.

3. FUZZY LOGIC

Fuzzy logic and set theory aid in uncertainty within data. Imagine a system that was assigned the task of accepting all objects that weighed around 120 pounds. If this system was designed to be crisp (Boolean), difficulties would arise concerning which objects the machine would actually accept. If an object were to weigh 115 or 125 pounds, the machine might reject the object. Given a human perspective however, these values might be close enough. Likewise, a similar question could be asked about objects weighing 118 or 122 pounds. Would the machine accept these objects? There would have to be an interval of acceptable values (such as 119 to 121 pounds)—anything within these values would be accepted, otherwise rejected. If an object weighed 118.999 the machine would reject it. Here lies the problem in crisp sets. To a human being, the value presented above might be satisfactory, even though a crisp system would reject it. Substituting fuzzy logic for Boolean logic could help to alleviate this problem.

Fuzzy logic is very similar to Boolean logic in how it is implemented. This means that the logical connectives that are employed in Boolean expressions can also be used in fuzzy expressions. The difference between these two forms of logic is that Boolean variables can have values of only 0 or 1. Fuzzy variables can be any value on the interval from 0 to 1 inclusively. Since the value of 0 implies false, and the value of 1 implies true, this suggests that fuzzy logic can have partial truths. The statements would not be entirely true nor entirely false, which is more akin to real world phenomena.

In fuzzy logic, the value given to a variable is known as its membership value. An expression with a membership value of 0 would imply that the statement is totally false similar to its Boolean counterpart. Likewise an expression with a membership value of 1 would imply that the statement is absolutely true. If a variable was given the value of 0.8, this might imply that it is close to being true, or a variable with a membership value of 0.2 might suggest that the statement is most likely false, but still considerable. A membership value of 0.5 would mean that there is total uncertainty about the variable—it is equally true and false.

When using logical connectives such as AND (\wedge), OR (\vee), and NOT (\neg) in Boolean expressions, simple rules can be used to evaluate statements. The statement $A \wedge B$ would only be evaluated to true if and only if both A and B were both true. Likewise the statement $A \vee B$ would be evaluated to false if and only if both A and B were false. When evaluating the expression $\neg A$, the value of the expression will be false if A is true and true if A is false. Fuzzy logic works in a very similar manner. The AND statement in fuzzy logic returns the minimum of the values being evaluated. If A was greater than B, the value of

Name	Truth Values for (A→B)
Gödel	1 where $A \leq B$ B where $A > B$
Goguen	1 where $A \leq B$ B/A where $A > B$
Kleene-Dienes	$\max(1 - A, B)$
Lukasiewicz	$\min(1, 1 - A + B)$
Reichenbach	$1 - A + A*B$

Table 1: Fuzzy Implications

B would be returned as the value of the expression. The OR statement in fuzzy logic returns the maximum of the values being evaluated. If A were greater than B, the value of A would be returned as the value of the expression. The NOT statement in fuzzy logic returns the inverse of the value being evaluated. If the expression being evaluated was $\neg A$, and A's value was 0.7, then the value of the expression would be 1-A or 0.3.

Like Boolean logic, fuzzy logic also employs implication. The classical implication can be defined in several unique forms which all produce the same results. Because fuzzy systems can use different many-valued logics as a base, we can choose from a variety of different logic connectives. When adjusting these theories for fuzzy logic, the different methods used for implication will produce differing results. In our project, we only chose a selected few of these implication connectives to implement. The implications that were used, along with their corresponding mathematical functions can be found in Table 1.

4. FUZZY INFORMATION RETRIEVAL

Fuzzy information retrieval is a set of information retrieval techniques that deal with using fuzzy logic to retrieve relevant documents [7]. Fuzzy information retrieval aids in selecting documents where the data appears to be uncertain or ambiguous. Some of the most prominent tools associated with fuzzy information retrieval are fuzzy thesauri and the BK-products [8].

Fuzzy thesauri aid greatly in information retrieval. When constructing a fuzzy thesaurus, each term is ranked by their similarity to other terms [7], [9]. If two terms contain high similarity (their membership values are high in relation to each other) then documents containing the term relating to the query term would also be retrieved. In Bosc et al.'s

Name	Symbol	Meaning	Definition
Circle Product	\circ	contains at least one of	$x(P \circ Q)z \Leftrightarrow x(P \cap Q)z$

Triangle Subproduct	\triangleleft	is included within	$x(P\triangleleft Q)_z \Leftrightarrow x(P\subseteq Q)_z$
Triangle Superproduct	\triangleright	contains every one of	$x(P\triangleright Q)_z \Leftrightarrow x(P\supseteq Q)_z$
Square Product	\square	are exactly the same as	$x(P\square Q)_z \Leftrightarrow x(P\equiv Q)_z$

Table 2: BK-Products

Name	Definition	Harsh Criteria	Mean Criteria
Circle Product	$(P\circ Q)_{ik}$	$\mathbf{V}_j (P_{ij} \wedge Q_{jk})$	$\frac{1}{N_j} \sum_j (P_{ij} \wedge Q_{jk})$
Triangle Subproduct	$(P\triangleleft Q)_{ik}$	$\mathbf{\Lambda}_j (P_{ij} \rightarrow Q_{jk})$	$\frac{1}{N_j} \sum_j (P_{ij} \rightarrow Q_{jk})$
Triangle Superproduct	$(P\triangleright Q)_{ik}$	$\mathbf{\Lambda}_j (P_{ij} \leftarrow Q_{jk})$	$\frac{1}{N_j} \sum_j (P_{ij} \leftarrow Q_{jk})$
Square Product	$(P\square Q)_{ik}$	$\mathbf{\Lambda}_j (P_{ij} \leftrightarrow Q_{jk})$	$\frac{1}{N_j} \sum_j (P_{ij} \leftrightarrow Q_{jk})$

Table 3: Harsh and Mean Criteria

paper [10] the benefits of a fuzzy thesaurus in information retrieval are discussed. Fuzzy thesauri aid greatest in the problem of word mismatch. As mentioned above, word mismatch is where an idea can be denoted by multiple terms (the example given was the relativity of *lizard* and *reptile*).

In John and Mooney's paper [11], a fuzzy module for aiding standard search engine results was developed. The authors implemented a fuzzy thesaurus contained on top of the Lycos search engine to enhance retrieval results. The first work, however, that introduced relational requests using a fuzzy thesaurus, that were run on top of a query retrieval engine was [8], see also [7].

Other essential elements to fuzzy information retrieval are the BK-products. They are outlined for the reader in Table 2. BK-products are fuzzy relational methods developed by Dr. Kohout and Dr. Bandler [12]. These methods are often used to retrieve information in a matrix environment [13] but implementations different from matrices are possible [14], [15].

We will define two matrixes P and Q. The first matrix P is a relational matrix mapping a set X to a set Y. The second matrix Q is a relational matrix mapping a set Y to set a Z. If X is a set of *users* and Y is a set of *terms*, then P would be the relational matrix from *users* to *terms*. Likewise if Y is a set of *terms* and Z is a set of *documents*, then Q would be the relational matrix from *terms* to *documents*.

We can now use the BK-products to define a relationship between the matrices P and Q. BK-products consist of four products, namely Circle product, Triangle Subproduct, Triangle Superproduct, and Square product. The meaning of these four products can also be located within Table 2. The function of the BK-products is to relate one relational matrix to another so that one can have a mapping from set X to set Z. In other words, BK-products will help map *users* to *documents* in the above example. They do this by employing the implication techniques listed in Table 1.

When using BK-products, the resulting matrix values need to be individually evaluated according to some criteria. There are two main forms of evaluation—Harsh and Mean criteria. These criteria can be found in Table 3. Harsh criteria evaluate the matrix computations by selecting the minimum value computed. Mean criteria add up the resulting matrix computations and then take the average of those values.

When the results are computed from any fuzzy relation, there must be some function to eliminate data whose membership functions are considered too low. Using an α -cut we can specify a value α , that for every data member whose membership value falls below the given α , we can regard that data member as not applicable (i.e. set their membership value to 0). There are two types of α -cuts—normal α -cuts and strong α -cuts. If we are using a normal α -cut of 0.5 and a data member has the value of 0.5, then that data member will continue to be considered after the cut. If we are using a strong α -cut value of 0.5, then any value *less than or equal to* 0.5 *will not* be considered.

5. THE FUZZY RELATIONAL INFERENCE ENGINE

Our system is known as the Fuzzy Relation Inference Engine (FRIE). Originally intended to be part of a larger Fuzzy Relational Medical Knowledge Base System (FRKBMS), the FRIE receives queries from a user about a specific medical inquiry and return the results in ranked order that is output in an easy to read format. The FRIE also allows for a large degree of customization by the user on how the query results are determined.

The FRIE contains five domains. These domains are *Patients*, *Signs & Symptoms*, *Syndromes*, *General Diseases*, and *Lab Tests*. The user is able to enter a query inquiring

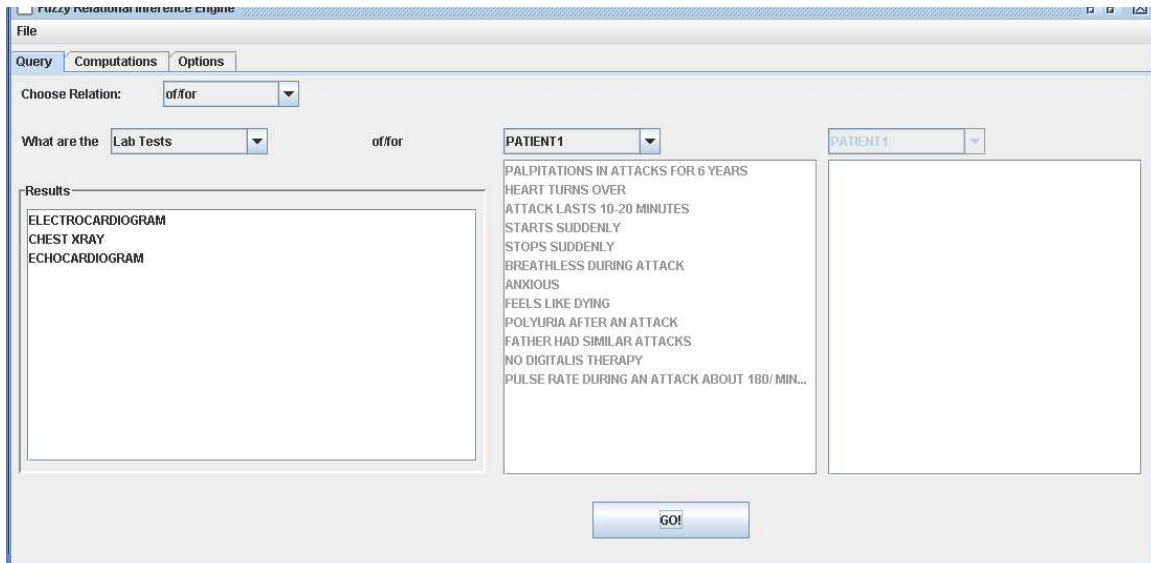


Figure 1: The Query Tab

about relationships from one set of these domains to another. An example query that the user might ask would be “What are *Lab Tests of/recommended for Patient1?*” The system would then retrieve the relevant information for the query given specifications set by the user.

The FRIE is divided into three separate tabs for organization and ease of use. The first of these tabs is the *Query* tab. The *Query* tab allows the user to select a predefined relation given for the query. Currently the system employs two distinct queries—the *of/recommended for* and the *are similar to* queries.

The *of/recommended for* query retrieves information pertaining to what the system believes is the cause or solution to a given set of parameters. If the user asks “What are the *lab tests of/recommended for* a specific set of *signs & symptoms*”, the system will return a set of *lab tests* recommended for the given set of *signs & symptoms* in order of relevance as shown by Figure 1. Users can also present queries such as “What are the *lab tests of/recommended for* a specific *syndrome*” or “What are the *lab tests of/recommended for* a specific *syndrome*”. The other query relation “*are similar to*” gives a more definitive query result by implementing a fuzzy thesaurus.

The second tab is the *Computations* Tab which is shown in Figure 2. The *Computations* Tab allows the user to view the computational path through the matrices that the query took to achieve the result. Located inside this tab are four separate sub-windows. Each sub-window keeps track of a specific domain within the system (*Signs & Symptoms*, *Syndromes*, *General Diseases*, and *Lab Tests*). Membership values are also kept track of for each particular entity located within the sub-windows. This screen is useful for analyzing queries by viewing intermediate steps and is also useful for debugging purposes.

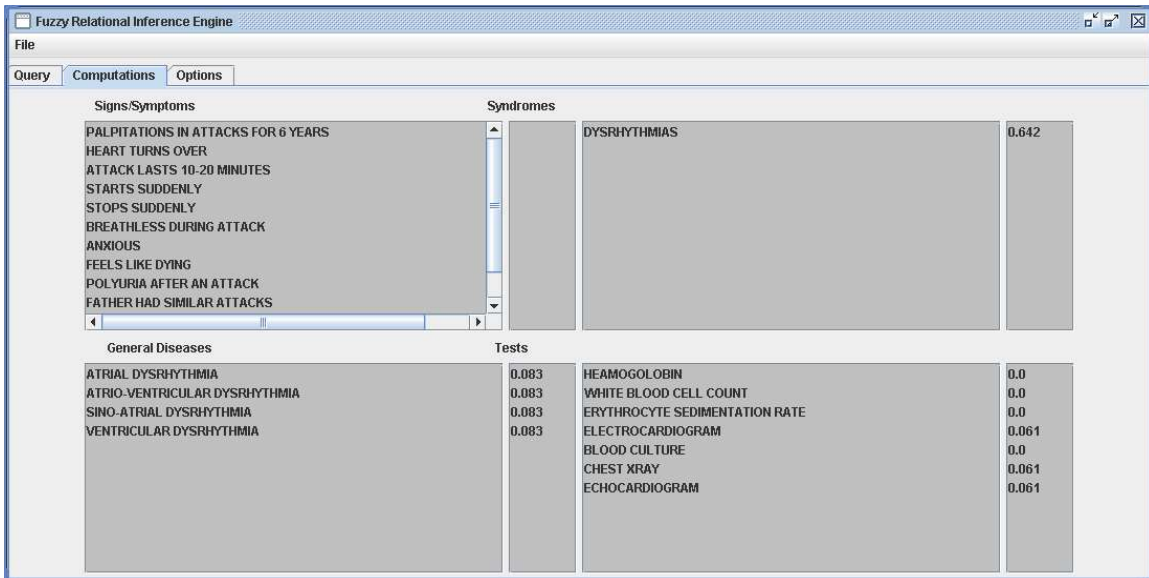


Figure 2: The Computations Tab

The final tab is the *Options* Tab which is shown in Figure 3. The Options Tab allows the user to customize how their query results are computed. The user is allowed to choose an α value and whether they want it to be a strong cut or a normal cut. Increasing the α value can help to eliminate query results if their membership values are too small. Likewise, decreasing the α value can aid in obtaining a more thorough result to a query.

The user can also choose which type of fuzzy implication they would like to use in retrieving their query results. The implications are currently limited to Gödel, Goguen, Lukasiewicz, Kleene-Dienes, and Reichenbach implications. Other options present within the *Options* Tab are whether the user would like to use a Mean or Harsh criteria, and what type of BK-product the user would like to employ when computing the results for a specified query.

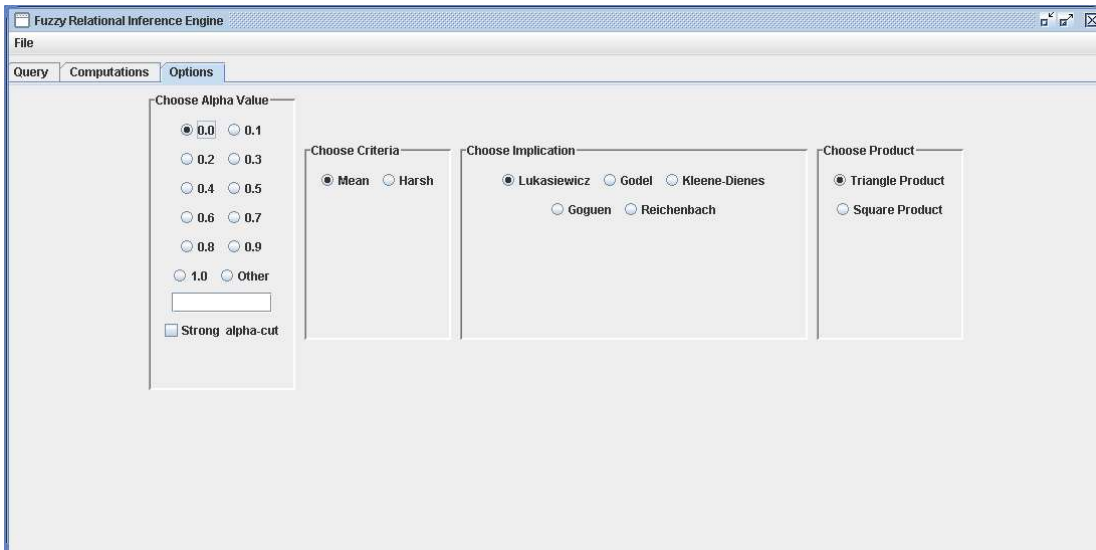


Figure 3: The Options Tab

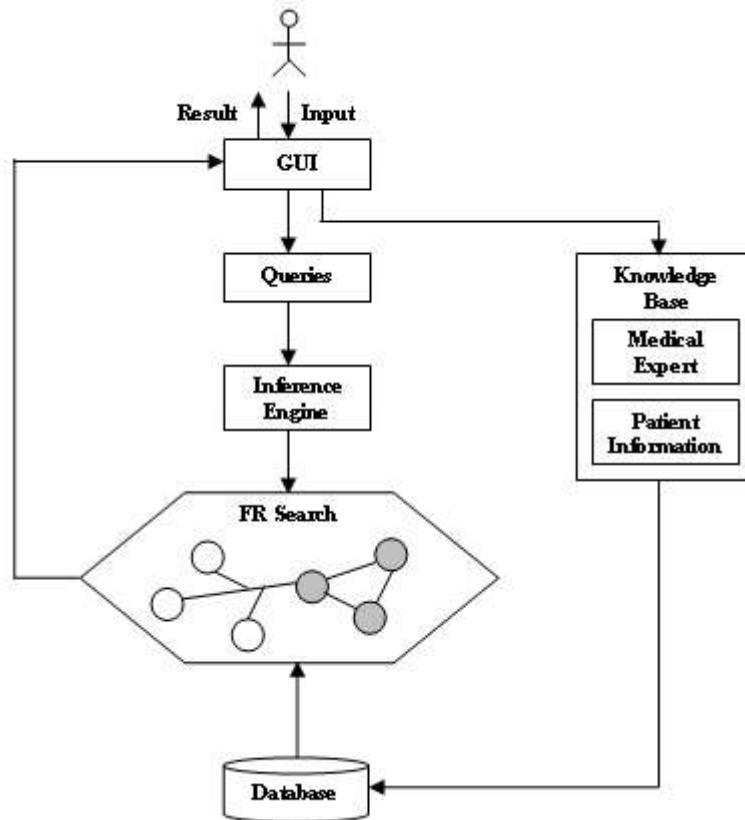


Figure 4: The Fuzzy Relational Knowledge Base Medical System

6. CONCLUSION

We were able to successfully develop a medical knowledge-base query system. Implementing the tools of fuzzy logic and information retrieval, the system is proficient at retrieving relevant query results in a timely manner. Even though the information retrieved using the system seems accurate, we would not recommend this system to be used someone not trained in medicine. The FRIE is a very efficient tool at suggesting solutions to medical queries, but the system lacks the experience of a fully trained medical physician. The FRIE would best be used to confirm or assert medical assumptions that are already predetermined by a medical expert, or provide suggestions and additional information to a medical expert that this expert has carefully verify as applicable to a particular individual patient.

7. FUTURE WORK

Currently, the FRIE has only limited functionality. There is much future work that can be done to make the FRIE a more complete system. Additional modules could be added to eventually turn the system into the FRKBMS. The diagram for a complete FRKBMS can be seen in Figure 4. Other improvements can be made in the following areas:

- Natural Language Query Translation - The system could be modified to accept queries developed by the user via natural language. This would give the system the ability to compute the results for a wider assortment of queries.
- Learning Capabilities – The system could be modified to allow the user to input new *signs & symptoms, syndromes, general diseases, or lab tests*. This would allow for greater customization and a more complete assessment for determining query results.
- Different Focal Topics – The System could be adapted to retrieve from different sources, rather than just medical information. For example, the system’s topic could be changed to the relationships between different types of reptiles.

8. ACKNOWLEDGEMENTS

We’re grateful to Professor L. J. Kohout and Dr. Isabel Stabile for allowing us to use their medical data used within this project.

9. REFERENCES

- [1] Klir, G., Yuan, K., *Fuzzy Sets and Fuzzy Logic: Theory and Application*, Prentice-Hall, 1995.
- [2] Terra, E. and Clarke, C., Scoring Missing Terms in Information Retrieval Tasks, *ACM Conference on Information and Knowledge Management (CIKM '04)*, pp. 50-58, 2004.
- [3] Bot, R. and Wu, Y., Improving Document Representations Using Relevance Feedback: The RFA Algorithm, *ACM Conference on Information and Knowledge Management (CIKM '04)*, pp. 270-278, 2004.

- [4] Takaki, T., Fujii, A., and Ishikawa, T., Associative Document Retrieval by Query Subtopic Analysis and its Applications to Invalidity Patent Search, *ACM Conference on Information and Knowledge Management (CIKM '04)*, pp. 399-405, November 2004.
- [5] Cai, D., Yu, S., Wen, J., and Ma, W., Block-based Web Search, *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '04)*, pp. 456-463, July 2004.
- [6] Zadeh, L., From Search Engines to Question-Answering Systems—The Problems of World Knowledge, Relevance and Deduction, June 2005.
- [7] Kohout, L.J. and Keravnou, E. and Bandler, W. Automatic documentary information retrieval by means of fuzzy relational products. *Fuzzy Sets in Decision Analysis*, pp.383-404, 1984, Eds. Gaines, B.R. and Zadeh, L.A. and Zimmermann, H.-J. North-Holland, Amsterdam.
- [8] Kohout, L.J. and Keravnou, E. Th. and Bandler, W., Design of an information retrieval system using triangle and square products of fuzzy relations for thesaurus Construction. *Proc. {IFAC} International Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis*, Marseille, France, 19-21 July 1983, Pergamon Press, Oxford.
- [9] Kohout, L.J. and Bandler, W., The use of fuzzy information retrieval techniques in construction of multi-centre knowledge-based systems, *Uncertainty in Knowledge-Based Systems* (Lecture Notes in Computer Science vol. 286), pp. 257 – 264, 1987, Ed. Bouchon, B. and Yager, R.R., Springer Verlag.
- [10] Bosc, P., Kraft, D., Petry, F., Fuzzy Sets in Database and Information Systems: Status and Opportunities, *Science Direct*, pp. 418-426, June 2005.
- [11] John, R., and Mooney, G., Fuzzy User Modeling for Information Retrieval on the World Wide Web, *Knowledge and Information Systems*, pp. 81-95, 2001.
- [12] Kohout, L.J. and Bandler, W., Use of fuzzy relations in knowledge representation, acquisition and processing. *Fuzzy Logic for the Management of Uncertainty*, pp 415-435, 1992, Eds. Zadeh, L.A. and Kacprzyk, J., John Wiley, New York.
- [13] Yue, L., Information Retrieval System Using Fuzzy Logic, *MS thesis, Dept. of Computer Science, Florida State University*, 1998.
- [14] Kohout, L.J. and Bandler, W., Relational-product architectures for information processing, *Information Science*, Vol. 37, pp.25-37, 1985.
- [15] Kohout, L.J. and Harris, M. Computer representation of fuzzy relations by means of threaded trees. *Journal of Fuzzy Logic and Intelligent Systems*, Vol. 3, No.1, pp. 41-64, 1993.

