

# A FUZZY SEMANTIC INFORMATION RETRIEVAL SYSTEM FOR TRANSACTIONAL APPLICATIONS

*A.O. Ajayi\**, *H.A. Soriyan*, and *G.A. Aderounmu*

*Dept. of Computer Science and Engineering, Obafemi Awolowo University, Nigeria*

*\*Email: [anuajayi@yahoo.com](mailto:anuajayi@yahoo.com), [Anuoluwapo.Ajayi@loria.fr](mailto:Anuoluwapo.Ajayi@loria.fr)*

*Email: [hasoriyan@yahoo.com](mailto:hasoriyan@yahoo.com), [gaderoun@oauife.edu.ng](mailto:gaderoun@oauife.edu.ng)*

## ABSTRACT

*In this paper, we present an information retrieval system based on the concept of fuzzy logic to relate vague and uncertain objects with un-sharp boundaries. The simple but comprehensive user interface of the system permits the entering of uncertain specifications in query forms. The system was modelled and simulated in a Matlab environment; its implementation was carried out using Borland C++ Builder. The result of the performance measure of the system using precision and recall rates is encouraging. Similarly, the smaller amount of more precise information retrieved by the system will positively impact the response time perceived by the users.*

**Keywords:** Fuzzy set, Query, Information, Fuzzy Inference Engine, Response time

## 1 INTRODUCTION

Information Retrieval (IR) is the science of searching for information in documents, searching for metadata which describe documents, or searching within databases, whether relational stand-alone databases or a hyper-textually-networked database such as the World Wide Web. The core functionality of an IR system is the retrieval of data from a database whose abstraction matches the description of an ideal object, inferred from a query (Bellman et al., 1992). A complex algorithm is used to search through the information, retrieve, and deliver the results to the user.

Traditional information retrieval models as employed in current search engines typically express a query as a set of keywords in which Boolean expressions involving words are used as terms to find information (Huang & Tsai, 2005). These keyword indexing systems and Boolean logic queries are sometimes equipped with statistical methods (Buckley & Fuhr, 1990), for instance, using frequency of occurrence of a keyword to strengthen the relationship between keyword and object. This model, called the keyword-based information retrieval model, uses keyword lists to describe the contents of information objects. The keyword list is a description that does not say anything about semantic relationships between keywords. According to Baeza-Yates and Ribiero (1999) the simplicity of these models usually prevents the formulation of more elaborate querying tasks. In addition, they give unsatisfactory query results when the semantic content of the query forms can not be easily represented.

In this study, a fuzzy semantic information retrieval model is developed for query tasks that can not be translated into clear forms. For instance, a potential buyer may be interested in an elementary Java textbook that introduces graphics programming with a pinch of data structure together with a bit of design patterns. Formulating a query using conventional retrieval models to describe the buyer's information needs specified above is difficult due to the ambiguity introduced in the specification of intensity of the data structure content and the concentration of design patterns to be included in said textbook. In a related manner, the use of advanced features of a typical search engine by users has always been limited because most users do not know about them or understand their use (Jansen et al., 2000). This study also presents a simple search interface that does not require users to use the advanced feature explicitly but is powerful enough to describe all the user's information needs.

The IR system retrieves documents based on a given query, and since documents and in most cases the queries themselves are often vague and uncertain, the use of fuzzy logic to relate these classes of objects with un-sharp boundaries (degree of membership) will certainly not be out of place, especially in transactional applications where the needs of the potential buyers are diverse and imprecise. This study uses the flexibility and power of fuzzy if-then rules to develop an information retrieval system with an interface that enables users to easily define their complex information needs and obtain results that will closely and precisely represent such needs.

In a related manner, the use of advanced features of a typical search engine by users has always been limited because most users do not know about them or understand how to use them (Jansen et al., 2000). This study presents a simple search interface that does not require users to use advanced features explicitly but is powerful enough to describe all the user's information needs.

This paper is organised as follows. Section 2 presents related work. In Section 3, a description of the fuzzy semantic information retrieval system is given. In Section 4, the simulation of the IR model using Simulink from the Matlab software is described. In Section 5, the implementation of the fuzzy semantic information retrieval model using Borland C++ Builder software is discussed. Performance evaluation of the IR model with results is presented in Section 6. Finally, Section 7 concludes the study.

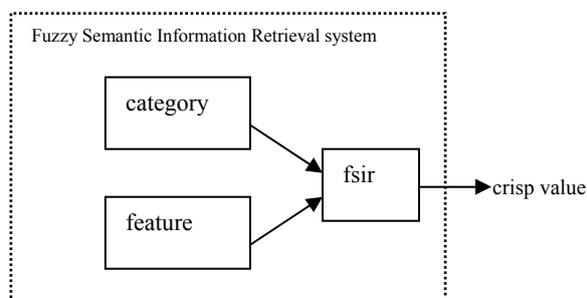
## 2 RELATED WORK

Several approaches have been proposed to help users specify their information needs more effectively. For instance, Belkin et al. (2003) proposed the use of an additional space for users to type a more wordy description of their information needs while Kelly et al. (2005) proposed the use of clarification forms to extract additional information about search context from users. These approaches are effective in best-match retrieval systems where longer queries generally lead to more relevant search results (Belkin et al., 2003). The downside of these methods is the increase in size of the query. Relevance Feedback (RF) (Oddy, 1977) and interactive query expansion (Efthimiadis, 1996) are other useful techniques to improve the quality of information provided by users of IR systems regarding their information needs. For the RF approach, the user presents the system with examples of relevant information that are later used to formulate an improved query. However, according to Kaski et al. (2005), getting users to use RF in the Web domain is difficult due to the complexity in conveying the meaning and the benefit of RF to users. Query suggestions offered based on query logs have the potential to improve retrieval performance with a limited burden on users. However, the approach is not suitable for commercial sites where re-execution of a similar query is rare.

Most commercial search engines provide advanced query interface to allow the specification of advanced queries using Boolean operators (AND, OR, and NOT) to combine terms. However, according to Jansen (2000) and Silverstein et al. (1999), only a small percentage of users are able to use the function perfectly because of the complexity involved in the formulation of such syntax. Separate studies conducted in Chi et al. (2001) and Teevan et al. (2005) revealed that gathering more information about users can improve the effectiveness of searches. However, this comes at the expense of storing more information about users than what is typically available from interaction logs, and also, there is difficulty in associating interactions with user characteristics.

## 3 FUZZY SEMANTIC INFORMATION RETRIEVAL MODEL

A bottom-up approach was employed in the design of the fuzzy semantic information retrieval system. For simplicity of presentation, an information retrieval system for books is considered here. The system consists of three modules, each implemented as a fuzzy inference system. The modules comprise *category*, which ranks related group of books (programming, artificial Intelligence, databases, etc.) into one partition; *feature*, which maps books using their features (price, format, publisher, etc.) into another partition; and *fsir*, which combines the partitions from the other two modules (*category* and *feature*) into a partition that is used to identify book(s) in the database. Figure 1 depicts the architecture of the model. In this model, the ambiguity in the database or user's query is represented using a fuzzy logic concept.



**Figure 1.** Fuzzy Semantic Retrieval System

A fuzzy semantic information retrieval system is described using a fuzzy logic model (Mamdani, 1974) of the form:

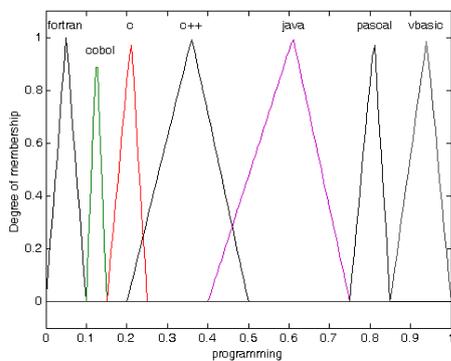
$$R^n : \text{if } X^1 = A^{1n} \text{ and } \dots X^m = A^{mn} \text{ then } Y = B^n \quad (1)$$

where  $R^n$  is a typical fuzzy rule of degree  $n$ , and the output of the rule depends on the degree of activation of its antecedent. The Mamdani inference scheme aggregates the outputs into a single fuzzy set for the variable  $Y$ , where de-fuzzification process is applied later to transform the output fuzzy set into a crisp value. The premise of a fuzzy rule specifies the condition that must be true before the firing of the rule. The firing of each component premise clause of a rule depends on the degree of truth associated with it due to the result of the fuzzified crisp input values. The premise space of the variables is partitioned into fuzzy subspace by studying the characteristics (context and content of books and their features) and the relationships between the books in the database. For instance, a Java programming textbook is likely to be related to computer graphics concepts, as one to four chapters of the Java textbook may be dedicated to these concepts. The linguistic variables are associated with a specific range of values by defining fuzzy sets over the Universe of Discourse (UoD) for each input variable.

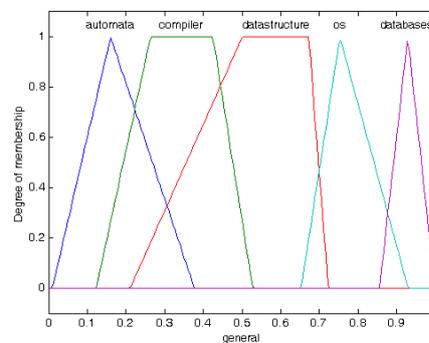
For clarity of presentation, the fuzzy logic modelling of the *category* module is discussed in-depth in this study. The *Category* module has five input variables (**programming**, **general**, **graphics**, **artificial intelligence**, and **internet**), with their linguistic terms expressed as fuzzy sets. For example, the input variable **general** has five linguistic terms, which represent a sub-group of books under it. This sub-group includes automata theory, compiling techniques, data structures and algorithms, operating systems, and databases, formally represented in the system as

$$\text{general} = \{\text{automata}, \text{compiler}, \text{datastructure}, \text{os}, \text{database}\}.$$

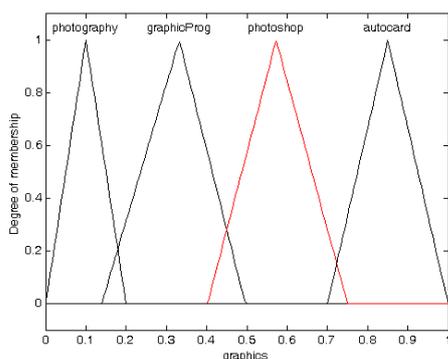
The degree to which an input value belongs to a given fuzzy set is computed by the respective membership function. After a careful analysis of the characteristics of the available books (input data), the triangular and trapezoidal membership functions are selected. The fuzzy set for each of the input variables (**programming**, **general**, **graphics**, **ai**, and **internet**) of the *category* module is shown in Figure 2.



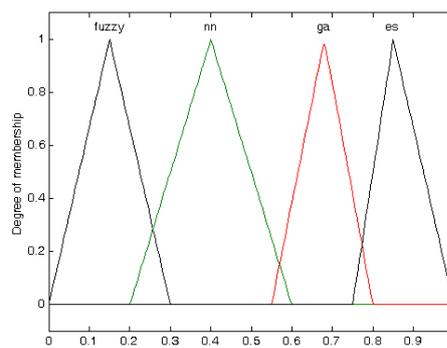
(a) Fuzzy set for programming



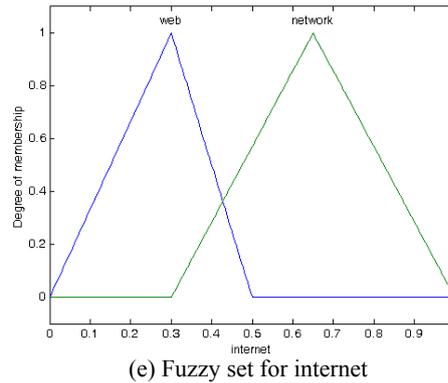
(b) Fuzzy set for general



(c) Fuzzy set for graphics



(d) Fuzzy set for ai



**Figure 2.** Fuzzy sets for the input variables

For the premise parameters identification (identification of premise and consequence) process, the space of each input variable is taken in turn and partitioned into fuzzy subsets while keeping the range of the other variables unpartitioned. Therefore, for the *category* module, when the ‘programming’ variable is partitioned, the variables ‘general’, ‘graphics’, ‘ai’, and ‘internet’ are not partitioned. In addition, when the ‘general’ variable is partitioned, the variables ‘programming’, ‘graphics’, ‘ai’, and ‘internet’ are not partitioned. At the end of the identification process for the consequence and premise parameters, a set of rules, which describes the behaviour of the fuzzy inference system, is produced. Looking at the membership functions depicted in Figure 2, the input variable ‘programming’ has seven sets of premises, the variable ‘general’ has five sets of premises, the variables ‘graphics’ and ‘ai’ have four sets of premises respectively, while the variable ‘internet’ has two sets of premises for each fuzzy subset. Hence, there are  $7*5*4*4*2=1120$  rules for each input variable. As there are five variables, the total number of rules will amount to  $1120*5=5600$ . However, using the rule of thumb, or heuristic, concerning the relationship among the variables, it is possible to reduce the number of rules significantly (Zhang et al., 1997). It should be noted that removing a fuzzy subset from the clause of a rule reduces the number of rules by 25. After eliminating irrelevant rules, the total number of rules left in the *category* module is 540. Similar procedures were carried out for the *feature* and *fsir* modules respectively. The *feature* module has 360 rules, while the *fsir* module has 720 rules.

Performing a fuzzy inference process involves the following steps:

- (i) Fuzzification: takes the crisp numerical values of the inputs and determines the degree to which they belong to each of the appropriate fuzzy sets via membership functions.
- (ii) Weighting: applies specific fuzzy logic operators (AND, OR, and NOT) on the membership values of the premise parts to get a single number between 0 and 1 that forms the fuzzy strength of each rule.
- (iii) Generation: creates the consequent relative to each rule.
- (iv) Defuzzification: aggregates the consequents to produce the output. From the various defuzzification methods, the weighted average is used in this study because it is reliable in average performances.

The following example illustrates how fuzzy logic is used to define a partition for books. We assume that a set of books has the following properties: 70-100% in content of C++ programming, with 20-40% content of automata theory, 60-100% of compilation techniques, and 20-60% content of data structures and algorithms. Then the values 0.35 and 0.36 (see Figure 2 (a-b), for instance) are entered for the input variables **programming** and **general** respectively for the *category* module. The corresponding truth values when obtained from Figure 2(a-b) give the values indicated in Table 1. These values, when plugged into rules that will be fired (3, 4, and 10 given below), give a crisp value that represents the ranking of books in this set. The membership functions (singletons) 2.5, 3.5, and 9.5 are assigned to ‘fuzzy-set is c3’, ‘fuzzy-set is c4’, and ‘fuzzy-set is c10’ respectively.

*Rule 3: If (programming is C++) and (general is compiler) then (fuzzy-set is c3)*

*Rule 4: If (programming is C++) and (general is automata) then (fuzzy-set is c4)*

*Rule 10: If (programming is C++) and (general is compiler) then (fuzzy-set is c10)*

For rule 3, the weighting for the output  $\mu(3)$  is given as  $\mu(3) = \min(1,1) = 1$ , and the output value ‘t(3)’ is 2.5. The other two rules give:

$$\mu(4) = \min(1, 0.3) = 0.3, t(4) = 3.5$$

$$\mu(10) = \min(1, 0.4) = 0.4, t(10) = 9.5$$

The weighted average computation yields:

$$\frac{\mu(3) * t(3) + \mu(4) * t(4) + \mu(10) * t(10)}{\mu(3) + \mu(4) + \mu(10)} = \frac{1 * 2.5 + 0.3 * 3.5 + 0.4 * 9.5}{1 + 0.3 + 0.4} = 4.32$$

Thus, the value 4.32 represents the partition of the set of books with the properties given in the example above.

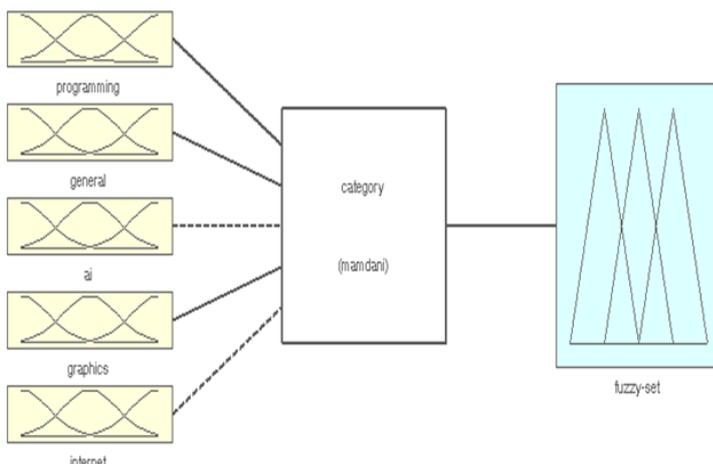
In most fuzzy IR systems (Anvari & Rose, 1987; Buckles & Petry, 1982; Medina et al., 1994), a numeric indexing form  $F$  exists where  $F: D \times T \rightarrow [0,1]$  such that  $F$  maps a given record  $r_j$  and a given keyword  $k_j$  to a numeric weight between 0 and 1.  $F(r_j, k_i) = 0$  implies that the record  $r_j$  is not at all about the concept represented by keyword  $k_i$ , and  $F(r_j, k_i) = 1$  implies that the record  $r_j$  is perfectly represented by the concept indicated by  $k_i$ . On the contrary, based on the assumption that the clustering process can be performed, our proposed model partitions the sample into sets such that each one contains exactly those values that represent one and only one real world object. Thus, the partitions are used as the set of values that should be returned.

**Table 1.** Truth values for the input variables

Programming (0.35)	General (0.36)
0	0.3
0	1
0	0.4
1	0
0	0
0	-
0	-

#### 4 MODELLING WITH MATLAB

In schools and industry, simulation tools based on MATLAB and Simulink are popular for science and engineering applications. MATLAB has many instructions and tools for designing applications and developing algorithms, while Simulink provides excellent graphical user interface and block libraries that allow rapid and easy building, simulating, and testing of system models. Furthermore, since MATLAB contains the Fuzzy Logic Toolbox, it turns into a powerful intelligent systems simulation and analysis tool. The fuzzy inference engine of the *category* module is depicted in Figure 3. The fuzzy semantic IR system was simulated using Simulink. Figure 4 depicts an interaction with the model when the following book features are entered: book category (100%), programming in C++ (0.65), price (\$85), pages (500), format-paperback (0.5), and published by O'Reilly (14.5).



**Figure 3.** Fuzzy inference engine of the *category* module

The simulated model responds with the crisp value of 18.53, which describes the user’s information needs. Figure 5 depicts the objects in the database (in the blue rectangle) that relate closely to this value (18.53) given by the fuzzy semantic information retrieval system modelled in Simulink and depicted in Figure 4. In this study, the relationship between the user’s query and the objects in the database is defined using Equation (2):

$$r = |q - o| \leq 1 \quad (2)$$

where q is the query, o represents the object, and r is the query result.

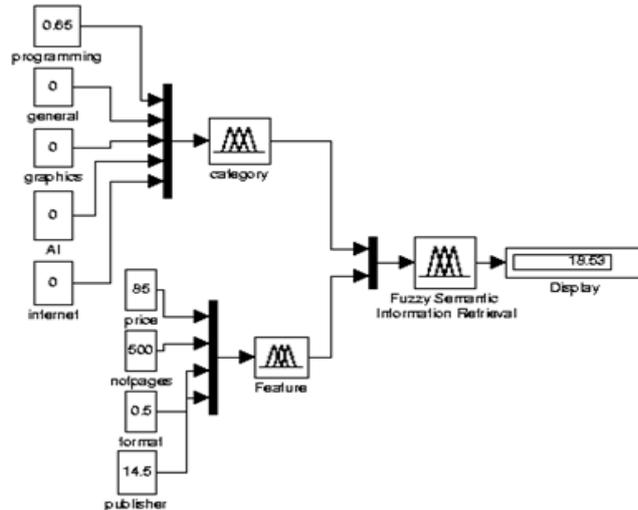


Figure 4. Simulink model of the information retrieval system

Beginning J2ME	1 Paperback	200	40 Programming	12.5
Computer Programming 2 Visual Basic	1 Paperback	360	39 Programming	13.9
Designing Visual Language: Strategies for Professional Commun.	1 Paperback	455	30 Programming	14.9
The C Programming Language	2 Paperback	274	45 Programming	14.5
Head First Design Patterns	1 Paperback	676	45 Programming	16
Introduction to C++	2 Paperback	500	50 Programming	18.5
Designing problems in C++	2 Paperback	480	46 Programming	18.4
Object orientation in C++	3 Paperback	680	70 Programming	18.5
Concept & Models of Programming	3 Hardcover	560	50 Programming	20.2
Beginning Programming with Perl	4 Paperback	450	45 Programming	21
Learning Perl	3 Paperback	320	75 Programming	21.4

Figure 5. Objects related to query given in Figure 4

## 5 MODEL IMPLEMENTATION

A fuzzy semantic product search system (ABC Bookstore) was developed using Borland C++ Builder running in a Windows environment on a PC. The search interface of the system is depicted in Figure 6. To conduct a query, a user specifies his/her information needs using one of two methods: either by dragging the scroll bar in which the Textbox above the scroll bar shows the corresponding value or by typing the feature value into the text box with the scroll bar moving accordingly. To prevent an erroneous input, the system disallows entering a value smaller than the minimum or larger than the maximum value. For instance, the minimum and the maximum prices considered in the system, based on information available on books used for the study, are \$20 and \$200 respectively. Hence, a user is not able to enter a price for a book that is less than 20 or greater than 200. The system also allows for indecisive selection by users. For instance, if the user is not too particular about the book category, he or she can simply check the “don’t know” option, and the system thereafter executes the query using default settings. To search the system, the user selects a coarser-grain option of book categories, for instance programming, then, using a combo box, selects a finer-grain option for specific programming books, e.g., JAVA, BASIC, Pascal, etc., and finally, specifies the feature (price, format, publisher, etc.) that represents

his/her information need. The search interface (see Figure 6) is divided into two parts (category and feature selection), which allows for easy entry of query specifications. The experimental database consists of five book categories and four book features. The categories include Programming, General, Design and Graphics, Artificial Intelligence, and Internet/Network, while the features considered are price, number of pages, publisher, and format.

The respective default settings for each category and features are shown in Figure 6. After finishing the query specifications, the user clicks the “Search” button, which triggers the system to search the database and to present a search result that closely relates to the user’s information need. For instance, assume a user wants a list of paperback format books consisting of 50% Java and 50% graphics in content, published by O’Reilly, and having price and pages not more than \$50 and 1000, respectively. The search interface showing how the query is entered is depicted in Figure 7.

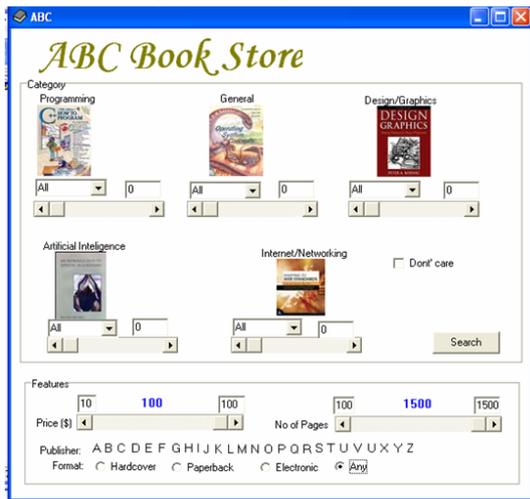


Figure 6. A fuzzy semantic information retrieval user interface

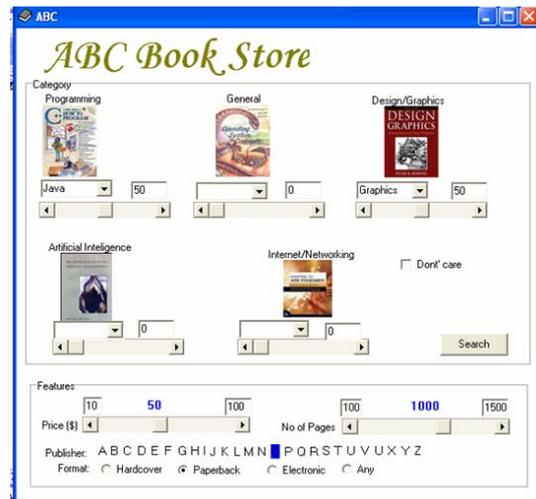


Figure 7. A sample search



Figure 8. A sample search result

Figure 8 shows the search result (targets) returned by the search engine. In this example, the system returns the objects that are closely related to the specified query. Equation (2) is used to determine the closeness between the objects and user’s query.

## 6 PERFORMANCE EVALUATION

The algorithm developed in Section 2 was applied to a university library database containing about 2000 computer textbooks. The database structure has fields that include bookID, author, title, isbn, publisher, category, price, edition number, and publication date. The proposed fuzzy IR system reads a book’s properties and partitions it into sets using book attributes as well as the relationship between book categories (i.e., is Java

programming related to database). Queries were manually built with terms from the title, publisher, category, and price fields. Twenty (20) queries were used in the evaluation, and some examples of the queries used are given in Table 2. In order to evaluate the performance of our proposed fuzzy semantic information retrieval model, a comparison of results was accomplished with respect to the fuzzy IR algorithm developed by Kraft et al. (1994) and the boolean IR system, using the standard recall and precision evaluations by computing the precision and recall at various cut-off points, where the precision is determined at various recall levels. The equations describing the two measures are given in Equations (3) and (4).

$$P = \frac{FR}{TF} \quad (3)$$

$$R = \frac{NR}{TR} \quad (4)$$

where P is the precision rate, FR represents targets found to be relevant, and TF represents total targets found. Similarly, R is the precision rate; NR is the number of targets found to be relevant, and TR is the total number of targets. Stated in another way, the recall rate is the ratio of the numbers of relevant targets discovered to the number of total relevant targets in the database repository. Figure 9 shows the precision evaluation when taking 10 recall levels from 0 to 100%; that is, given a ranked result of the search, we used human experts to judge the relevance of the first ranked document to the query. If it was truly relevant, it was associated with a 100% precision level. The same procedure was repeated for the second ranked document, the third ranked document, and so forth. The values in Figure 9 were obtained using the average of 20 searches of roughly similar queries. The proposed fuzzy semantic information retrieval system outperforms the other two IR systems, with 82% on average of relevant documents being retrieved versus 75% and 41% of relevant documents obtained in fuzzy IR and boolean IR systems respectively. Our model outperformed both boolean and fuzzy IR systems.

Although our system and the fuzzy IR system are both built using fuzzy logic, they use different indexing and query processing strategies, which leads to their performance difference. In traditional fuzzy IR, indexed terms map every record and the query to a numeric weight between 0 and 1. As a result, queries are always evaluated against all objects in the database. On the contrary, in our system, objects are partitioned into sets, and queries are evaluated against the most suitable partition(s) from the available list of partitions. In this way, our system benefits from grouping objects with similar interests. By doing so, a query is only evaluated against the correct set/partition, which usually produces good results.

Moreover, most fuzzy IR systems use Hamming distance or Euclidean distance as the distance measure between query and object, whereas in our model, the formula for the matching degree is simpler (given in Equation (2)), which slightly reduces the computational costs.

**Table 2.** Queries used in the evaluation

Query	Meaning
60% of Java and 30% of Graphics	Extract all textbooks having at most 60 percent of Java programming content and at most 30 percent of graphics content
Data structures and price between \$45 and \$80	Extract from the data structure category, all books having price in the range \$45 to \$80
20 percent of database	Extract all computer textbooks with at most 20 percent of their contents containing "database" concepts

Similarly, we evaluated the fuzzy semantic information retrieval model against the boolean IR system with respect to information size returned. Our belief is that the smaller and more precise the information returned, the better for users as well as for the resources (bandwidth, processor, RAM) used to process and convey the information. This will go a long way to improving the user perceived quality of service, particularly the response time, where bandwidth limitation do often results in slow traffic.

We compared the two models with relation to search result size and size of information (images and texts) returned when the same query is invoked using the two IR systems. This process was repeated a number of times using different queries, and the average value of five trials was recorded. The approximate system response time was calculated for the two IR systems using Equation (5) for different transmission speeds of 28.8kbps, 56kbps, 96kbps, and 128kbps.

$$RT = \frac{PgSize}{B} \quad (5)$$

where PgSize is the size of the information returned (texts and images) measured in Kbytes, and B is the transfer rate.

The average page sizes returned by the two models were taken and plotted against bit rates. The response time was recorded for both as shown in Figure 10. The approximate response time experienced in the fuzzy semantic information retrieval model is shorter (half of the time recorded for the boolean IR system) for the different transmission speeds. This is a result of the extraction of less and more relevant information. This attribute will go a long way to reducing loads on computing device and the network.

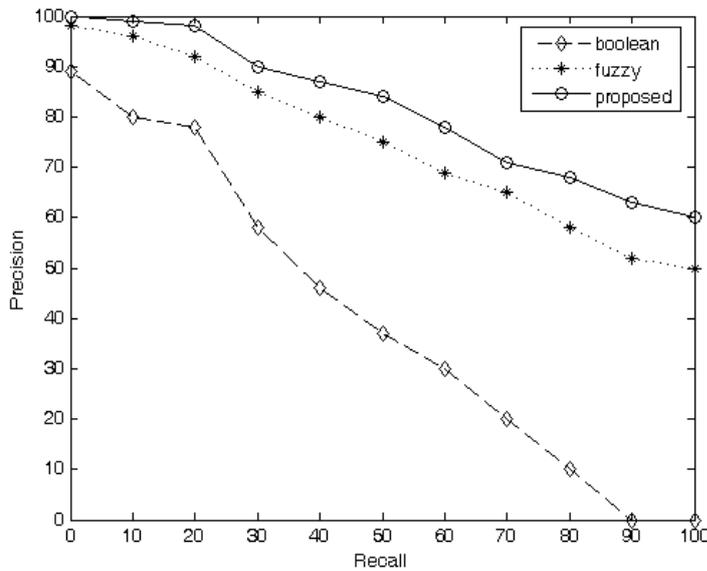


Figure 9. Precision and recall rates of the model

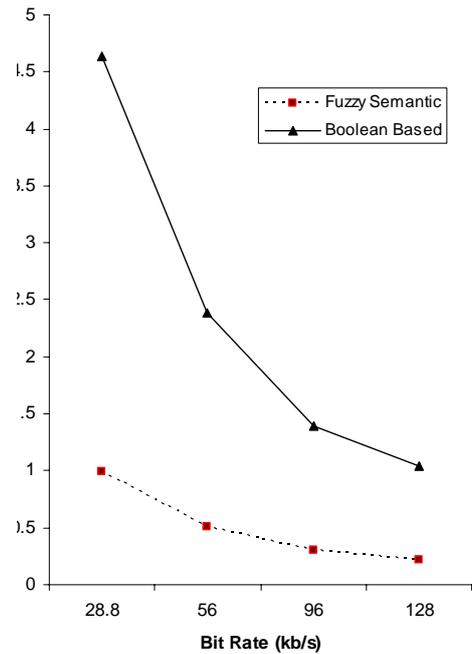


Figure 10. System response time

## 7 CONCLUSION AND FUTURE WORK

In this article we have developed a fuzzy semantic information retrieval model that can be used to query transactional databases. The most apparent aspect is the use of a fuzzy inference system to develop the three sub-modules of the system, namely *category*, *feature*, and *fsir*. The experimental results obtained from our simulation model have demonstrated that fuzzy semantic information retrieval has an acceptable performance, especially for recall rate. Similarly, the fuzzy semantic information retrieval model retrieves less and more precise information when benchmarked with the logic based boolean model. This feature will definitely reduce the response time as perceived by users of transactional applications conducting a product search, particularly where computing and network (bandwidth) resources are limited. The user-friendly and comprehensive interface design of the system also lets users enter their uncertain specifications in query forms.

Although the application we have developed is a monolithic one, in the future, we plan to make it web-based and deployed on a server, so that network users can experiment with it and evaluate its performance in real life applications.

## 8 REFERENCES

Anvari, M. & Rose, G. (1987) Fuzzy relational databases. In Bezdek, J. (Ed.), *The Analysis of Fuzzy Information, Vol. II*, Boca Raton, FL: CRC Press.

- Belkin, N.J. et al. (2003) Query length in interactive information retrieval. *Proc. ACM SIGIR*, 205-212.
- Bellman, R., Kalaba, R., & Zadeh, L. (1992) Abstraction and pattern classification. *Fuzzy Models for Pattern Recognition*, IEEE Press.
- Buckles, B. & Petry, F. (1982) A fuzzy model for relational databases. *Internat. J. Fuzzy Sets and Systems* 7, 213-226.
- Buckley, C. & Fuhr, N. (1990) Probabilistic document indexing from relevance feedback data. *Proc. 13th Int. Conf. ACM SIGIR on Research and Development in Information Retrieval*, Bruxelles.
- Chi, E. H., Pirulli, P. L., Chen, K., & Pitkow, J. E. (2001). Using information scent to model user information needs and actions and the Web. In *Proc. ACM SIGCHI*, 490-497.
- Efthimiadis, E.N. (1996) Query expansion. *Annual Review of Information Science and Technology* 31, 121-187.
- Huang, Y-P. & Tsai, T. (2005) A fuzzy Semantic Approach to Retrieving Bird Information Using Handheld Devices. *IEEE Intelligent Systems* 20(1), 16-23.
- Jansen, B.J. (2000). An investigation into the use of simple queries on Web IR systems. *Inf. Res. Vol. 6 (1)*.
- Jansen, B. J., Spink, A., & Saracevic, T. (2000) Real life, real users, and real needs: A study and analysis of user queries on the web. *Information Processing and Management* 36(2), 207-227.
- Kaski, S., Myllymäki, P., & Kojo, I. (2005) User models from implicit feedback for proactive information retrieval. *Workshop at UM Conference; Machine Learning for User Modeling: Challenges*.
- Kelly, D., Dollu, V.D. & Fu, X. (2005). The loquacious user: a document-independent source of terms for query expansion. *Proc. ACM SIGIR*, 457-464.
- Kraft, D., Bordogna, G., & Pasi, G. (1994) An extended fuzzy linguistic approach to generalized Boolean retrieval. *J. Information Sci.* 2(3), 119-134.
- Mamdani, E.A. (1974) Application of Fuzzy Algorithms for Control of Simple Dynamic Plant. *Proceedings of the Institution of Electrical Engineers* 121, 1585-1588.
- Medina, J., Pons, O. & Vila, M. (1994) Gefred: a generalized model to implement fuzzy relational databases. *Inform. Sci.* 47, 234-254.
- Oddy, R. (1977) Information retrieval through man-machine dialogue. *J. Doc.* 33(1), 1-14.
- Rubens, N.O. (2006). The application of fuzzy logic to the construction of the ranking function of information retrieval systems. *Computer Modelling and New Technologies Vol. 10(1)*, 20-27.
- Silverstein, C., Marais, H., Henzinger, M., & Moricz, M. (1999) Analysis of a very large web search engine query log. *SIGIR Forum* 33 (1), 6-12.
- Teevan, J., Dumais, S.T., & Horvitz, E. (2005) Personalizing search via automated analysis of interests and activities. *Proc. ACM SIGIR*, 449-456.
- Zhang, J.H., Jia, L.M., & Zhang, X.D. (1997) On a novel predictive control. *Proceedings of the American Control Conference*, Albuquerque, 1251-1255.

(Article history: Received 6 October 2008, Accepted 21 July 2009, Available online 30 September 2009)