c15

A Software Tool to Teach the Performance of Fuzzy IR Systems based on Weighted Queries

Enrique Herrera-Viedma[1], Sergio Alonso[1], Francisco J. Cabrerizo[1], Antonio G. Lopez-Herrera[2], Carlos Porcel[3]
[1] Dept. of Computer Science and A.I., Faculty of Library Science, University of Granada, Granada, Spain
viedma, salonso, fjcabrerizo@decsai.ugr.es

[2] Dept. of Computer Science, Faculty of Computer Science, University of Jaen, Jaen, Spain
aglopez@ujaen.es

[3] Dept. of Computing and Numerical Analysis, Faculty of Computer Science, University of Cordoba, Cordoba, Spain
carlos.porcel@uco.es

This paper describes a software tool that allows us to teach students the principles and concepts of Fuzzy Information Retrieval Systems based on weighted queries. This tool is used in the course Information Retrieval Systems Based on Artificial Intelligence at the Faculty of Library and Information Science at the University of Granada. With this teaching tool students learn the management of the fuzzy weighted query languages which could be used in any conventional Web search engine to improve the representation of user information needs.

Keywords: teaching, education, fuzzy weighted queries, fuzzy connectives, fuzzy information retrieval.

1. INTRODUCTION

Due to the growth of e-business, the Web has become a critical part of many real-world systems. Thus, it is ingreasingly important that information technology professionals and students be proficient and knowledgeable in various Web technologies like [1] Web mining, query processing, Information Retrieval (IR) models, search engines, meta-search engines, recommender systems, information filtering, Web quality evaluation, etc., which are also evolving at a rapid rate, making it critical to keep up-to-date with them [6].

At the Faculty of Library and Information Science at the University of Granada there are different degree courses that address these evolving needs. In particular, there exists a degree course called "*Information Retrieval Systems based on Artificial Intelligence*" which deals with the study and analysis of artificial intelligence tools applied in the design of Information Retrieval Systems (IRSs). The key goals of this course are to learn the foundations of fuzzy tools and genetic algorithms and its application in the design of IRSs. As it is known, both are important soft computing tools [2,26] and are being satisfactorily applied in the development of the Web access technologies [3,7,8,9,13,14,19].

Fuzzy IRSs (FIRSs) are those IRSs that use the potential of the fuzzy tools to improve the retrieval activities [9,14]. In our degree course we focus on fuzzy IR models that use fuzzy weighted queries to improve the representation of user information needs and fuzzy connectives to process such queries. We use hands-on-keyboard classroom exercises for teaching and practising the use of fuzzy weighted queries and fuzzy connectives. However, in our teaching experience we observed that this is not enough to show learners the searching skills of FIRSs.

IR instruction is an obvious application for computer-supported learning systems. The advantage of using computer-supported learning systems is that the learner gets a realistic feeling of the particular IRS used and he/she learns typical operations of IRSs [11]. To do that, it is possible to use real world search engines like Google, Altavista, Lycos or to build ad-hoc training IRSs [5,6,10,11,18]. There are very few training IRSs [11] and, particularly, a fuzzy IR training system does not exist. As it is pointed out in [11], existing training IR systems present several shortcomings, e.g., they do not give feedback about the performance or success of user queries, it is not possible to observe how a user query is evaluated, and it is not possible to compare the performance of different types of user queries and different evaluation procedures of user queries.

In this paper we introduce a software tool, which is just being used as first time. This software gives students a chance to acquire the complex skills that provide those FIRSs based on weighted queries. This is a Web-based computer-supported learning application whose goal is to provide a environment for demonstrating the performance of fuzzy queries and their evaluation using different fuzzy connectives. It offers students the opportunity to see and compare the achieved results of different weighted queries. User can choose different semantics (threshold, relative importance, ideal importance, [12,17] to formulate weighted queries, different fuzzy connectives to evaluate these queries (maximum, minimum, OWA operators, Induced OWA operators) [23,24], and different expression domains (numerical or linguistic one) [12,16] to assess weights associated with queries. Furthermore, several standard test collection (ADI, CISI, CRANSFIELD, etc) can be used. Finally this tool presents visualization tools to show better evaluation processes of queries.

The paper is structured as follows: in Section 2 we review the components of the fuzzy IR models that we want to teach; Section 3 describes the structure and performance of our software tool and shows some example of its use. In the last section, we discuss some lessons learned from our experience and suggest some possible uses and improvements of our computerized system.

2. CHARACTERISTICS OF FUZZY IR MODELS TO TEACH

The set of fuzzy IR models that we have implemented in our software application presents the following components:

1. Database.

We assume a database built like in an usual IRS [1,20] and therefore where the IRS-user interaction is unnecessary because it is built automatically. Then, the database stores the finite set of documents $D=\{d_1,\dots,d_m\}$, the finite set of index terms $T=\{t_1,\dots,t_l\}$, and the representation R_{d_j} of each document d_j characterized by a numeric indexing function $F: D\times T \rightarrow [0,1]$ such that $R_{d_j}=\sum_{i=1}^{l} F(d_j,t_i)/t_i$. Using the numeric values in [0,1] F can weight index terms according to their significance in describing the content of a document in order to improve the retrieval of documents. Test standard collections have been indexed using a $tf\cdot idf$ scheme.

2. Query system

The implemented fuzzy IR models provide a query system with fuzzy weighted Boolean query languages to express user information needs. With these languages each query is expressed as a combination of the weighted index terms which are connected by the logical operators AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). The weights can be numerical values in [0,1] or linguistic values taken from a label set S which is defined using the concept of fuzzy linguistic variable [25].

By assigning weights in queries, users specify restrictions on the documents that the IRS has to satisfy in the retrieval activity. In the literature we find that a user can impose four kinds of restrictions on documents to be retrieved which are associated to four different semantic interpretations [12,17]:

a. *Importance semantics.* This semantics defines query weights as measures of the relative importance of each term for the query with respect to the others in the query. By associating relative importance weights to terms in a query, the user is asking to see all documents whose content represents the concept that is more associated with the most important terms than with the less important ones. In practice, this means that the user requires that the computation of the relevance degree of a document be dominated by the more heavily weighted terms.

b. *Threshold semantics.* This semantics defines query weights as satisfaction requirements for each term of query to be considered when matching document representations to the query. By associating threshold weights with terms in a query, the user is asking to see all the documents sufficiently about the topics represented by such terms. In practice, this means that the user will reward a document whose index term weights F exceed the established thresholds with a high relevance degree, but allowing some small partial credit for a document whose F values are lower than the thresholds.

c. *Perfection semantics.* This semantics defines query weights as descriptions of ideal or perfect documents desired by the user. By associating weights with terms in a query, the user is asking to see all the documents whose content satisfies or is more or less close to his ideal information needs as represented in the weighted query. In practice, this means that the user will reward a document whose index term weights are equal to or at least near to term weights for a query with the highest relevance degrees.

d. *Quantitative semantics.* This semantics defines query weights to express criteria that affect the quantity of the documents to be retrieved, i.e., constraints to be satisfied by the number of documents to be retrieved.

Formally, in [4,22] a fuzzy weighted Boolean query with one semantics was defined as any legitimate Boolean expression whose atomic components are pairs $<t_i, c_i>$ belonging to the set, $T \times H$; where $c_i \in [0,1]$ or S is a value that qualifies the importance that the term t_i must have in the desired documents. Accordingly, the set Q of the legitimate queries is defined by the following syntactic rules:

i) $\forall q = <t_i, c_i> \in T \times H \rightarrow q \in Q$;
ii) $\forall q, p \in Q \rightarrow q \wedge p \in Q$;
iii) $\forall q, p \in Q \rightarrow q \vee p \in Q$;
iv) $\forall q \in Q \rightarrow \neg q \in Q$;
v) all legitimate queries $q \in Q$ are only those obtained by applying rules i-iv, inclusive.

3. Evaluation procedure

To evaluate these weighted Boolean queries we use a constructive bottom-up process based on the *criterion of separability* (one of the most important properties of the wish list) [22]. This process includes two steps:

- Firstly, the documents are evaluated according to their relevance only to atoms of the query. In this step, a partial relevance degree is assigned to each document with respect to each atom in the query.

- Secondly, the documents are evaluated according to their relevance to Boolean combinations of atomic components (their partial relevance degree), and so on, working in a bottom-up fashion until the whole query is processed. In this step, a total relevance degree is assigned to each document with respect to the whole query.

We represent the evaluation procedure using an evaluation function $E: Q \times D \rightarrow H$. Depending on the kind of query, E obtains the relevance degree RSV_j of any $d_j \in D$ according to the following rules:

1.- $E(<t_i, c_i>, d_j) = g((F(d_j, t_i), c_i) = RSV_j$ where g is a matching function that depends on both expression domain and semantic interpretation associated to c_i.

2.- $E(q \wedge p, d_j) = E(q, d_j) \quad FUZZCONN_{AND} \quad E(p, d_j)$, where $FUZZCONN_{AND}$ is a fuzzy connective that models a combination behaviour of values similar to a t-norm.

3.- $E(q \vee p, d_j) = E(q, d_j) \quad FUZZCONN_{OR} \quad E(p, d_j)$, where $FUZZCONN_{OR}$ is a fuzzy connective that models a combination behaviour of values similar to a t-conorm.

4.- $E(\neg(q), d_j) = Neg(E(\neg(q), d_j))$, where Neg is a complement operator of fuzzy sets.

We should point out that the fuzzy connectives that apply in the evaluation procedure are mainly the family of connectives of type OWA or IOWA [23,24] whose behaviour can be controlled through an *orness measure* [23].

3. A SOFTWARE TOOL TO TEACH FUZZY IR SYSTEMS BASED ON WEIGHTED QUERIES

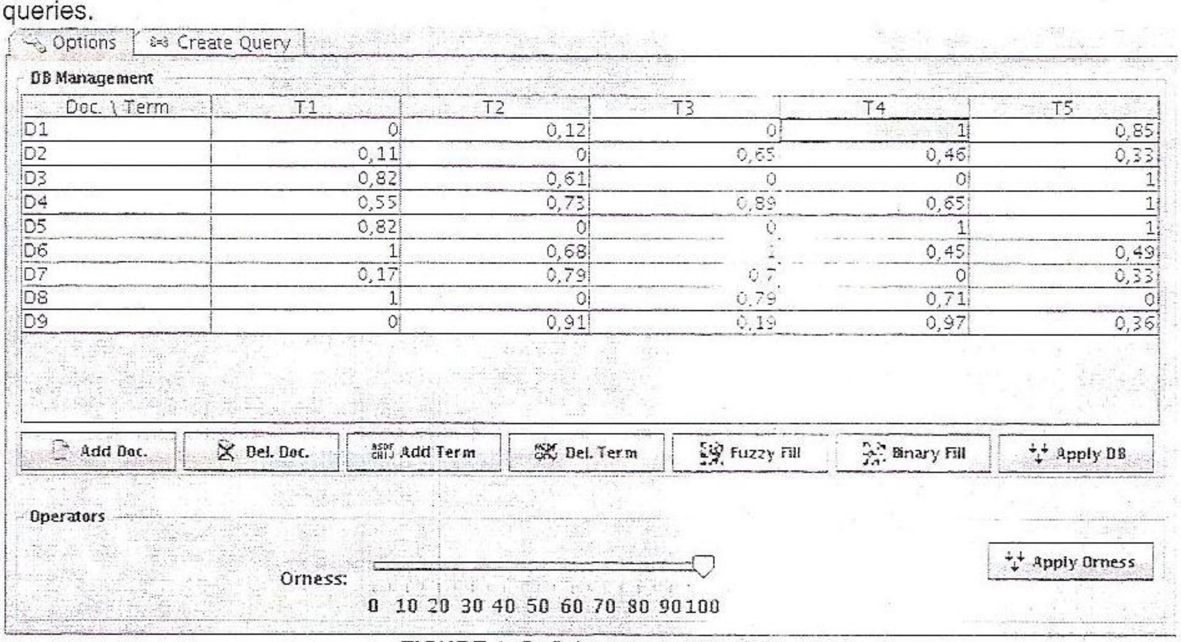
We have developed this software tool at the Faculty of Information and Library Science at the University of Granada as a useful fuzzy weighted query analysis tool (see http://sci2s.ugr.es/pruebaApplet/). The goal of this software application is to provide an environment for demonstrating students the performance of fuzzy weighted queries and in such a way to aid in their learning.

This software tool is a Web-based application that is implemented in Java language. It is composed of three main modules: i) definition module of test collection, ii) formulation module of weighted queries, and ii) a visual execution module of queries.

3.1 Definition Module of Test Collection

An experimental test collection consists of a database, a collection of queries and relevance assessments indicating which documents are relevant in respect to a given query [21]. Usually, the performance of a system is measured by means of the precision and recall achieved across the whole set of queries. As in [11,15] our goal is to encourage the analysis of the individuals queries, and therefore, we only need an instructional test collection. However, the tool provides some standard test collection (ADI, CISI, CRANFIELD).

We have decided to give the capability students to build their own test collections (see Figure 1), i.e., toy test collections, to analyze the performance of the different fuzzy weighted queries. In the definition of test collection they can establish particular queries and which documents of the database match the relevance requirements of these queries.



Doc. \ Term	T1	T2	T3	T4	T5
D1	0	0,12	0	1	0,85
D2	0,11	0	0,65	0,46	0,33
D3	0,82	0,61	0	0	1
D4	0,55	0,73	0,89	0,65	1
D5	0,82	0	0	1	1
D6	1	0,68	1	0,45	0,49
D7	0,17	0,79	0,7	0	0,33
D8	1	0	0,79	0,71	0
D9	0	0,91	0,19	0,97	0,36

FIGURE 1: Defining a test collection.

3.2 Formulation Module of Weighted Queries

We have designed a formulation module of weighted queries that allows students to define their weighted queries (see Figure 2). To define a weighted query they have to choose: search terms, Boolean connectives (And, Or, and Not), query structure, expression domain of weights (numerical or linguistic), semantic interpretation, and values of weights.

3.3 Visual Execution Module of Queries

We have implemented an execution module that allows measuring and visualizing the performance of any query executed. Before to execute a weighted query a student has to choose fuzzy connectives that will be associated with the Boolean connectives in the evaluation procedure. This is done choosing a level of orness [23].

This module generates performance feedback for the students by means of visual tools. This feedback can be given in both ways by showing internal aspects of evaluations of weighted queries, e.g. evaluation trees, (see Figures 3 and 4) or analysis of search results using traditional precision/recall curves. Furthermore, this module allows the comparison of different different weighted queries in the evaluation process.

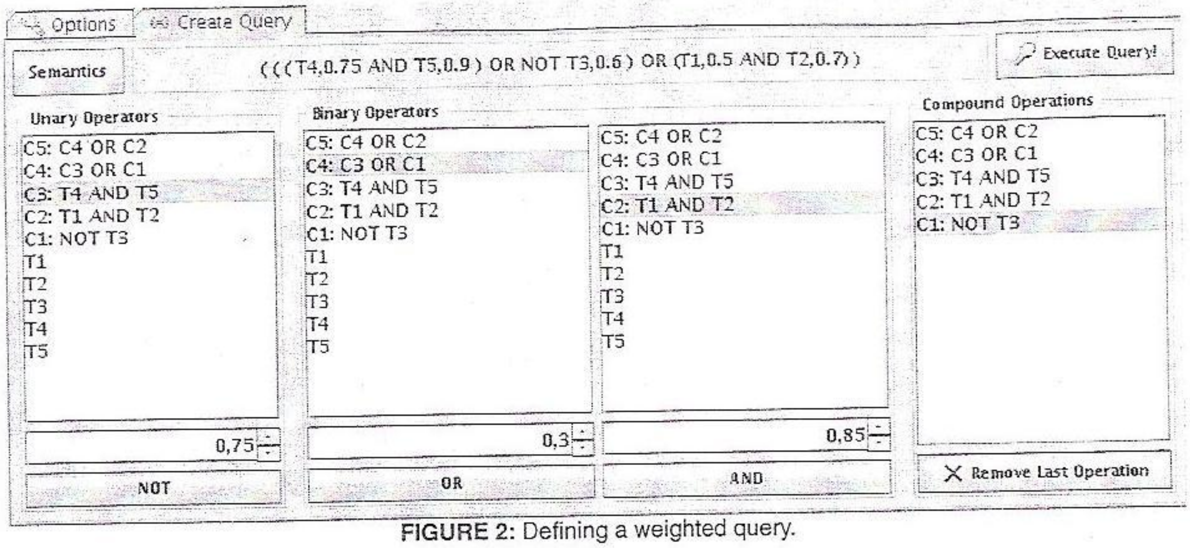


FIGURE 2: Defining a weighted query.

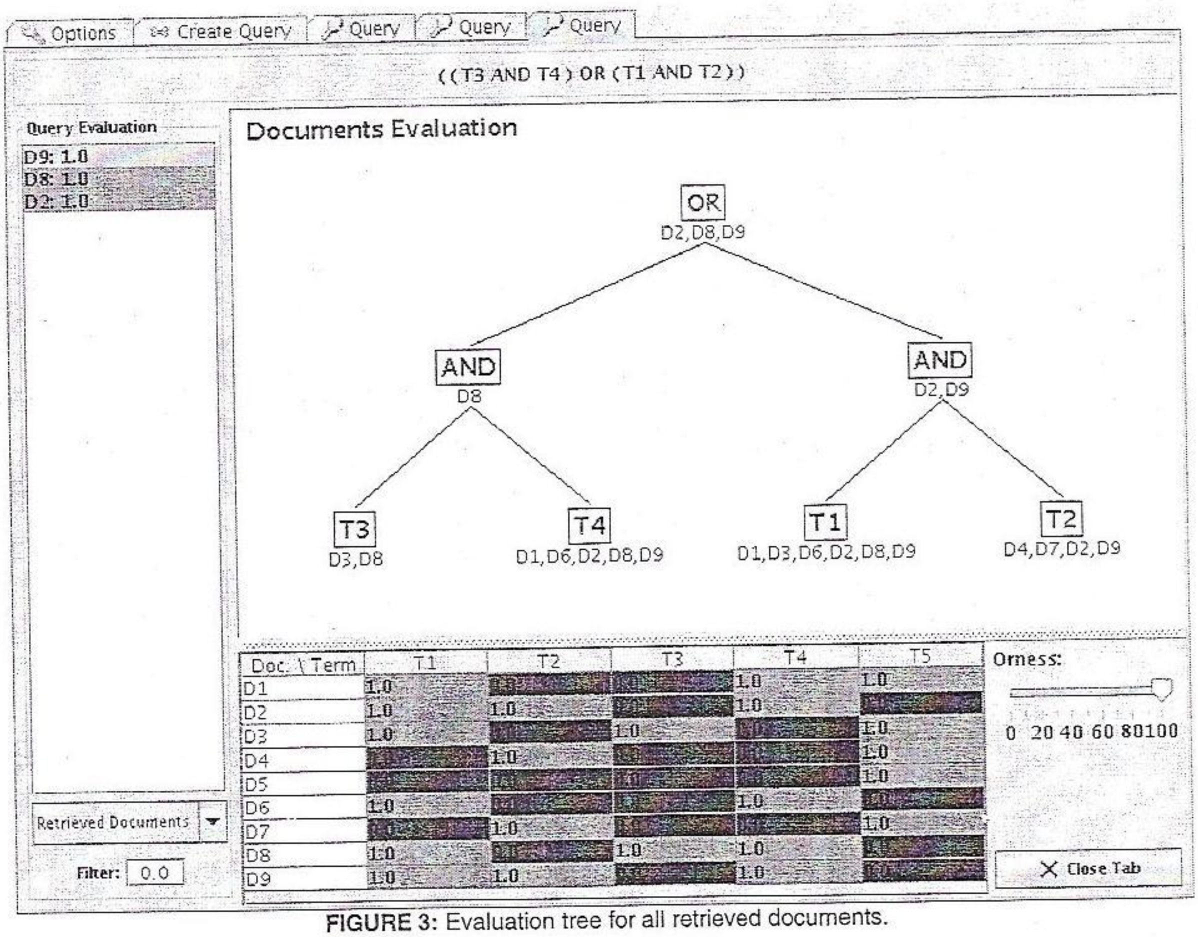


FIGURE 3: Evaluation tree for all retrieved documents.

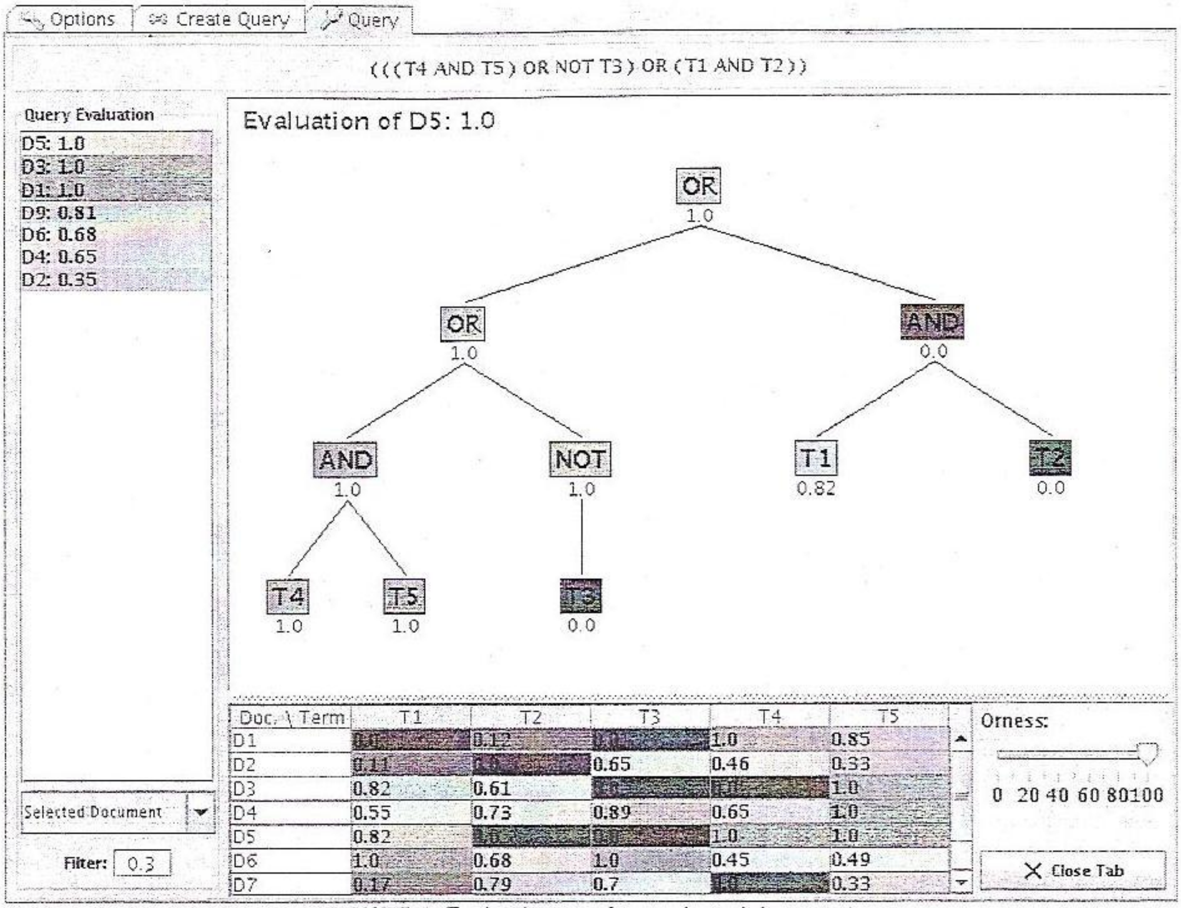


FIGURE 4: Evaluation tree for a selected document.

4. CONCLUSIONS

In this contribution we have presented a software tool to teach the use of fuzzy weighted queries in IR activity. Our experience reveals that the use of this tool enhances students' learning on fuzzy IR systems.

Currently, we are working to develop a better set of tools for building fuzzy search engines, that integrate spidering, indexing, searching, and storage, to be applied in real situations. In such a way, we want to stimulate students' creativity and innovation and to improve their learning. Additionally, we are designing a survey with which students can express their experiences and suggestions to improve this software tool.

REFERENCES

[1] Baeza-Yates R and Ribeiro-Neto B. (1999) *Modern Information Retrieval.* Addison-Wesley.
[2] Bonissone P P. (1997) Soft computing: the convergence of emerging reasoning technologies. Soft Computing, 1(1), pp. 6-18.
[3] Bordogna G and Pasi G. (1993) Special issue: Management of imprecision and uncertainty. Journal of the American Society for Information Science, 49(3), pp. 193-194.
[4] Bordogna G and Pasi G. (1993) A Fuzzy Linguistic Approach Generalizing Boolean Information Retrieval: A Model and Its Evaluation. Journal of the American Society for Information Science, 44, pp. 70-82.
[5] Caruso E. (1981) Computer aids to learning online retrieval. Annual Review of Information Science and Technology, pp. 317-336.
[6] Chau M Huang Z and Chen H. (2003) Teaching key topics in computer science and information systems through a Web search engine project. ACM Journal of Educational Resources in Computing, 3(3), pp. 1-14.
[7] Cordon O and Herrera-Viedma E. (2003) Special issue on soft computing applications to intelligent information retrieval on the internet. International Journal of Approximate Reasoning, 34(2-3).
[8] Crestani F and Pasi G. (2003) Handing vagueness, subjectivity, and imprecision in information access: an introductionto to the special issue. Information Processing & Management, 39, pp. 161-165.

[9] Crestani F and Pasi G (Eds). (2000) *Soft Computing in Information Retrieval: techniques and applications.* Studies in Fuzziness and Soft Computing Series, vol. 50, Physica-Verlag.
[10] Griffith J C and Norton N P. (1981) A computer assisted instruction programs for end users on an automated information retrieval systems. *Proc. of the Second National Online Meeting,* New York 239-248.
[11] Halttunen K and Sormunen E. (2000) Learning information retrieval through an educational game. Is Gaming sufficient for learning?. Education for Information, 18(4), pp. 289-311.
[12] Herrera-Viedma E. (2001) Modeling the retrieval process for an information retrieval system using an ordinal fuzzy linguistic approach. Journal of the American Society for Information Science and Technology, 52(6), pp. 460-475.
[13] Herrera-Viedma E and Pasi G. (2006) Soft Approaches to Information Retrieval and Information Access on the Web: An Introduction to the Special Issue. Special Issue on Soft Approaches to Information Retrieval and Information Access on the Web. Journal of the American Society for Information Science and Technology, 57(4), pp. 511-514.
[14] Herrera-Viedma E, Pasi G and Crestani F. (2006) *Soft Computing in Web Information Retrieval: Models and Applications.* Studies in Fuzziness and Soft Computing Series, vol. 197, Physica-Verlag.
[15] Hull D. (1996) Stemming algorithms: a case study for detailed evaluation. Journal of the American Society for Information Science, 52(6), pp. 70-84.
[16] Kraft D H and Buell D A. (1983) Fuzzy sets and generalized Boolean retrieval systems. International Journal of Man-Machine Studies, 19, pp. 45-56.
[17] Kraft D H, Bordogna G and Pasi G. (1994) An Extended Fuzzy Linguistic Approach to Generalize Boolean Information Retrieval. Information Sciences, 2, pp. 119-134.
[18] Markey K and Atherton P. (1978) *ONTAP online training and practice manual for ERIC data base searchers.* Syracuse, New York: ERIC Clearinghouse on Information Sources, Syracuse University.
[19] Nikravesh M, Loia V and Azvine B. (2002) Special issue on Fuzzy logic and the Internet (FLINT): Internet, world wide web and search engines. Soft Computing, 6(5), pp. 287-299.
[20] Salton G and McGill M H. (1984) {\em Introduction to modern information retrieval}. New York: McGraw-Hill.
[21] Sparck Jones K and van Rijsbergen C J. (1976) Information retrieval test collections. Journal of Documentation, 32, pp. 59-75.
[22] Waller W G and Kraft D H. (1979) A Mathematical Model of a Weighted Boolean Retrieval System. Information Processing & Management, 15, pp. 235-245.
[23] Yager R R. (1988) On ordered weighted averaging aggregation operators in multicriteria decision making. IEEE Transactions on Systems, Man, and Cybernetic, 18, pp. 183-190.
[24] Yager R R and Filev D P. (1999) Induced ordered weighted averaging operators. IEEE Transaction on Systems, Man and Cybernetics, 29, pp. 141-150.
[25] Zadeh L A. (1975) The concept of a linguistic variable and its applications to approximate reasoning. Part I. Information Sciences, 8, pp. 199-249; Part II. Information Sciences, 8, pp. 301-357; Part III. Information Sciences, 9, pp. 43-80.
[26] Zadeh L A. (1997) What is soft computing?. Soft Computing, 1(1), pp. 1.

BCS- Information Retrieval Specialist Group

Proceedings of TLIR 2007

The First International Workshop on Teaching and Learning in Information Retrieval

Editors:
Andrew MacFarlane
Juan Manuel FernAndez Luna
Iadh Ounis
Juan F. Huete

TABLE OF CONTENTS