Fuzzy Logic and Multiobjective Evolutionary Algorithms as Soft Computing Tools for Persistent Query Learning in Text Retrieval Environments

Oscar Cordón
Dept. of Computer Science and A.I.
University of Granada
18071 - Granada (Spain)
E-mail: ocordon@decsai.ugr.es

Félix de Moya
Faculty of Information Sciences
University of Granada
18071 - Granada (Spain)
E-mail: felix@ugr.es

Carmen Zarco
PULEVA S.A.
Camino de Purchil, 66
18004 - Granada (Spain)
E-mail: czarco@puleva.es

***Abstract*— Persistent queries are a specific kind of queries used in information retrieval systems to represent a user's long-term standing information need. These queries can present many different structures, being the "bag of words" that most commonly used. They can be sometimes formulated by the user, although this task is usually difficult for him and the persistent query is then automatically derived from a set of sample documents he provides.**

In this work we aim at getting persistent queries with a more representative structure for text retrieval issues. To do so, we make use of soft computing tools: fuzzy logic is considered for representation and inference purposes by dealing with the extended Boolean query structure, and multiobjective evolutionary algorithms are applied to build the persistent fuzzy query. Experimental results will show how both an expressive fuzzy logic-based query structure and a proper learning process to derive it are needed in order to get a good retrieval efficacy, when comparing our process to single-objective evolutionary methods to derive both classic Boolean and extended Boolean queries.

I. Introduction

Persistent queries (PQs) are useful tools for information retrieval system (IRS) [2] users having a relatively specific information need remaining fixed during a certain time period [18], [13]. By the definition of these kinds of queries, the information filtering process can be put into effect by delivering interesting information to a user, thus getting him permanently updated on his interest topics [14].

Although different structures can be used to represent a PQ, it is usually difficult for a user to formulate the query regardless its structure [18], [12], [13]. This way, explicit PQs automatically learned from a training set of documents by means of user's relevance feedback are normally considered in information routing systems.

Soft computing tools have demonstrated to be useful in the personalization of IRSs, providing them with flexibility and some kind of "intelligence". The latter is viewed as the capability of automatically adaptating to a context or service based on implicit behavior and learning instead of explicit solicitation from users [11], [19], [8].

One of the ways to add flexibility to an IRS is to make it tolerant to uncertainty and imprecision —both inherent to the user-system interaction— what can be achieved allowing a more natural expression of users' needs [19]. To do so, some flexible query languages based on the application of fuzzy set theory have been proposed which make possible simple and approximate expressions of subjective information needs [4]. In this contribution, we will deal with extended Boolean queries, considering them to improve the representative power of classic Boolean ones when used as PQ structures.

On the other hand, the IRS self-adaptativeness can be tackled by the machine learning perspective of soft computing, put into effect by evolutionary algorithms [1], neural networks and Bayesian networks, among others. These techniques can be hybridized with the representative power of flexible query languages to get "intelligent" IRSs [8]. In particular, evolutionary algorithms has obtained promising results in IR [9]. We will consider the use of multiobjective evolutionary algorithms [6] to automatically derive several fuzzy PQs representing the user's information needs in a single run.

So, the aim of this contribution is to propose the use of a new, more flexible query structure —the extended Boolean query— to appropriately represent PQs for text retrieval and to introduce an evolutionary learning process to explicitly derive PQs of this composition. The latter will be based on a multiobjective technique able to automatically generate several PQs with a different trade-off between precision and recall in a single run.

The proposal will be validated in a simulated text retrieval environment considering seven different information needs extracted from the classic Cranfield collection. Its efficacy will be compared with Boolean and extended Boolean PQs extracted by means of single-objective evolutionary algorithms.

To do so, this contribution is structured as follows. Section 2 is devoted to introduce the PQ framework basics and the use of soft computing tools to construct them. Then, Section 3 briefly describes the single-objective evolutionary algorithms considered to derive Boolean and extended Boolean PQs. The multiobjective GA-P proposal to construct fuzzy PQs is reviewed in Section 4. Section 5 presents the experiments developed to test it and the analysis of results, while the

conclusions are pointed out in Section 6. Finally, an Appendix is reported with the basics of the extended Boolean retrieval model.

II. Construction of Flexible Persistent Queries

A. Information Filtering and Persistent Queries

Information filtering refers to a information seeking process where the user is assumed to be searching information addressing a specific long-term interest [18], [14].

In an information filtering system, the user's permanent information need is represented in the form of a "profile". The most common profile structure is the "bag of words", which is based on a set of keywords representing the user's interest. Many systems assume a implicit definition of the profile by the user, although this comes with the classic human-computer interaction "vocabulary problem", involving the difficulty for the user to select the right words to communicate with the system. This is specially important in this case as the profile can neither be too broad —as in that case the information filtering system would retrieve so many non relevant documents— nor too specific —as much valuable information can be lost—.

Due to this reason, machine learning techniques have been applied to construct "implicit profiles" [12], [18]. In this case, the profile is automatically learned by the system from a training set of documents provided by the user.

Belkin and Croft suggested that IR techniques can be successfully applied to information filtering [3]. This way, the profile can be represented as a query formulated using any IR retrieval model, the so called PQ [13]. Besides, IR query formulation techniques such as relevance feeeback or inductive query by example can be applied in information filtering.

B. Flexible Persistent Queries

As different query structures from different IR retrieval models can be used to represent a PQ, the obtaining of effective retrieval results depends on the user's ability to express his information needs in the form of a query both in information filtering and in IR. It has been shown that the user often does not have a clear picture of what he is looking for and can only represent his information need in vague and imprecise terms, resulting in a situation known as fuzzy-querying [17].

Flexible query languages can help to solve this problem due to their capability of personalization. A flexible query language is a language that makes possible a simple and approximate expression of subjective information needs [19] (see the description of the fuzzy IR model in the Appendix).

This way, the modeling of user profiles in the form of flexible PQs can help us to improve both the retrieval efficacy and the comprehensibility of the obtained PQs.

C. Inductive Query by Example of Persistent Queries

Inductive Query by Example (IQBE) [5] was proposed as "a process in which searchers provide sample documents and the algorithms induce the key concepts in order to find other relevant documents". It works by taking a set of relevant (and optionally, non relevant documents) provided by a user and applying an off-line machine learning process to automatically generate a query describing the user's needs from that set. The obtained query can then be run in other IRSs to obtain more relevant documents.

Hence, IQBE techniques can be directly applied to construct PQs for information filtering, as they work in the same way than explicit profile learning methods. In this contribution, we consider the application of existing evolutionary IQBE techniques to the derivation of flexible PQs.

III. Evolutionary Methods to Construct Persistent Boolean and Fuzzy Queries

As the structure of Boolean and extended Boolean queries is easily represented in the form of a expression tree, the IQBE approaches to automatically derive them are usually based on a specific evolutionary algorithm, genetic programming (GP) [1]. The next two subsections are devoted to briefly review the two most known algorithms for each of the query types considered: Smith and Smith's proposal for Boolean IRSs [20] and Kraft et al.'s [16] for fuzzy IRSs.

A. The Smith and Smith's Inductive Query by Example Algorithm for Boolean Information Retrieval Systems

Coding Scheme: The Boolean queries are encoded in expression trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators AND, OR or NOT, according to the grammar:

<QUERY>::= <TERM> | (<QUERY><OPERATOR><QUERY>)
<OPERATOR>::= AND | OR | NOT
<TERM>::= t_1 | ...| t_n

Selection Scheme: Each generation is based on selecting two parents, with the best fitted one having a greater chance to be chosen, and generating two offspring from them. Both offspring are added to the current population[1].

Genetic Operators: The usual GP crossover is considered, which is based on randomly selecting one edge in each parent and exchanging both subtrees from these edges between the both parents. No mutation operator is used[2].

Generation of the Initial Population: All the individuals in the first population are randomly generated. A pool is created with all the terms included in the set of relevant documents provided by the user, having those present in more documents a higher probability of being selected.

Fitness function:: The following function, combining the common precision and recall measures, is maximized:

$$F = \alpha \cdot P + \beta \cdot R \quad ; \quad P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \quad ; \quad R = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d} \tag{1}$$

[1]Our implementation differs in this point as we consider a classical generational scheme with elitism. The intermediate population is created by means of tournament selection [1], which involves the random selection of a number t of individuals from the current population and the choice of the best adapted of them to take one place in the new population.

[2]We do use a mutation operator which changes a randomly selected term or operator by a random one, or a randomly selected subtree by a randomly generated one.

with $r_d \in \{0,1\}$ being the relevance of document d for the user and $f_d \in \{0,1\}$ being the retrieval of document d in the processing of the current query. α and β are real-valued coefficients weighting the relative importance of precision and recall.

B. The Kraft et al.'s Inductive Query by Example Algorithm for Fuzzy Information Retrieval Systems

Coding Scheme: The fuzzy queries are encoded in expression trees, whose terminal nodes are query terms with their respective weights and whose inner nodes are the Boolean operators AND, OR or NOT.

Selection Scheme: It is based on the classical generational scheme with elitism, where the intermediate population is created from the current one by means of classic roulette wheel selection[3].

Genetic Operators: The usual GP crossover is used. The following three possibilities are randomly selected —with the showed probability— for the GP mutation:

a) Random selection of an edge and random generation of a new subtree that substitutes the old one located in that edge (p=0.4).
b) Random change of a query term for another one, not present in the encoded query, but belonging to any relevant document (p=0.1).
c) Random change of the weight of a query term (p=0.5).

Generation of the Initial Population: A first individual is obtained by generating a random tree representing a query with a maximum predefined length and composed of randomly selected terms existing in the initial relevant documents provided by the user, and with all the term weights set to 1. The remaining individuals are generated in the same way but with a random size and random weights in [0,1].

Fitness function: The fitness function showed in Section III-A is considered.

IV. A Multiobjective GA-P Algorithm to Construct Persistent Extended Boolean Queries

Our multiobjective proposal to learn persistent extended Boolean queries is based on a specific variant of the GP technique, the GA-P paradigm [15]. This evolutionary algorithm that simultaneously evolves hybrid individuals encoding both an expression and its associated parameters (a query tree and its numeric weigths in our case) has demonstrated a larger performance than the basic GP used by Kraft et al. [7].

The IQBE algorithm deals with the fuzzy query learning as a multiobjective problem, thus being able to automatically generate several queries with a different trade-off between precision and recall in a single run. To do so, classic Fonseca and Fleming's Pareto-based MOGA scheme [6] is considered.

The different components of the multiobjective evolutionary algorithm (MOGA-P) are reviewed as follows. A wider description can be found in [10].

[3]Once again, our implementation considers the tournament selection to improve the algorithm's performance.

Coding Scheme: The expressional part (GP part) encodes the query composition —terms and logical operators— and the coefficient string (GA part) represents the term weights with a real coding scheme, as shown in Fig. 1.

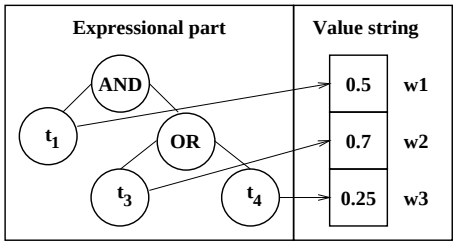

Fig. 1. Individual encoding the query $0.5\ t_1\ AND\ (0.7\ t_3\ OR\ 0.25\ t_4)$

Fitness Function: The precision and recall criteria —computed as showed in Section III-A— are jointly maximized.

Pareto-based Multiobjective Selection and Niching Scheme: The selection scheme involves the following steps:

1) Each individual is assigned a rank equal to the number of individuals dominating it plus one (non-dominated individuals receive rank 1).
2) The population is increasingly sorted by that rank.
3) Each individual is assigned a fitness value according to its ranking in the population: $f(C_i) = \frac{1}{rank(C_i)}$.
4) The fitness assignment of each group of individuals with the same rank is averaged among them.

Then, a niching scheme is applied in the objective space to obtain a well-distributed set of queries with a different trade-off between precision and recall (see [10] for details). Finally, the intermediate population is obtained by tournament selection.

Genetic Operators: The BLX-α crossover operator [1] is applied twice on the GA part to obtain two offsprings. Michalewicz's non-uniform mutation operator [1] is considered to perform mutation on that part.

The usual GP crossover is considered for the GP part. Two different mutation operators are applied: random generation of a new subtree, and random change of a query term by another not present in the encoded query.

V. Experiments and Analysis of Results

The document collection considered to design our experimental setup has been the classic *Cranfield* collection, composed of 1398 documents about Aeronautics [2]. It has been automatically indexed by first extracting the non-stop words, applying a stemming algorithm, thus obtaining a total number of 3857 different indexing terms, and then using a binary indexing for the Boolean PQ experiments and the normalized IDF scheme (see Appendix A) to generate the term weights in the document representations for the fuzzy PQ ones.

Among the 225 queries associated to the Cranfield collection, we have selected those presenting 20 or more relevant documents (queries 1, 2, 23, 73, 157, 220 and 225). The number of relevant documents associated to each of these

seven queries are 29, 25, 33, 21, 40, 20 and 25, respectively. The relevance judgements associated to each of these selected queries have been considered to play the role of seven different user's information needs.

For each one of these queries, the documentary base has been randomly divided into two different, non overlapped, document sets, training and test, each of them composed of a fifty percent of the (previously known) relevant and irrelevant documents for the query.

The two single-objective GP algorithms have been run five different times on each of the seven training sets representing the corresponding PQ learning scenario, considering five different combinations for the fitness function coefficients weighting the relative importance of precision and recall ($(\alpha, \beta) = \{(1.2, 0.8), (1.1, 0.9), (1, 1), (0.9, 1.1), (0.8, 1.2)\}$). This way, five different PQs of each kind (Boolean and fuzzy), with a different precision-recall trade-off have been derived for each information need. On the other hand, MOGA-P has been run[4] a single time on each training document set and five PQs well distributed on the Pareto front has been selected from each of the seven Pareto sets obtained.

All these PQs (the five Boolean queries, the five fuzzy queries obtained from Kraft et al.'s algorithm, and the other five fuzzy queries derived from MOGA-P) has been run on the corresponding test set once preprocessed[5] in order to evaluate their capability to retrieve relevant information for the user.

Tables I to III shows the retrieval efficacy of the five PQs derived for each scenario. In those tables, P and R stand for the precision and recall values, and rr/rt for the absolute number of relevant and retrieved documents, respectively.

The experiments developed allow us to develop an interesting analysis. At first sight, we can remark the good performance of our proposal. On the one hand, the PQs constructed by it show a higher diversity with respect to those of the other two approaches. As expected, the multiobjective scheme allows us to cover the Pareto front in a better way than selecting different weight combinations for the single-objective fitness function. On the other hand, the extended Boolean queries generated by MOGA-P got the best retrieval efficacy in the training sets in every case, i.e., they were able to more properly model the information need represented by the document set provided by the user. Finally, their performance on the test collections, that is, the capability of the derived PQs to retrieve new relevant documents for the user's information need, is significant. It can be seen how the MOGA-P fuzzy PQs have a better trade-off between precision and recall in the test document sets than the other two groups of PQs: while their recall is usually a little bit lower (in all cases but in query 157), their precision is pretty higher, what makes the access to the new relevant information easier for the user.

TABLE I
RETRIEVAL EFFICACY OF SMITH AND SMITH'S BOOLEAN PQS

		Training set			Test set		
$\#q$		P	R	rr/rt	P	R	rr/rt
1	1	1.000	0.571	8/8	0.000	0.000	0/1
	2	1.000	0.429	6/6	0.000	0.000	0/6
	3	1.000	0.500	7/7	0.059	0.400	6/101
	4	0.074	1.000	14/189	0.046	0.600	9/195
	5	0.066	1.000	14/211	0.048	0.667	10/210
2	1	1.000	0.250	3/3	0.000	0.000	0/6
	2	1.000	0.417	5/5	0.111	0.077	1/9
	3	1.000	0.500	6/6	0.200	0.077	1/5
	4	0.046	1.000	12/260	0.047	0.923	12/255
	5	0.032	1.000	12/378	0.033	0.923	12/361
23	1	1.000	0.375	6/6	0.000	0.000	0/4
	2	1.000	0.250	4/4	0.000	0.000	0/0
	3	1.000	0.312	5/5	0.111	0.059	1/9
	4	0.047	1.000	16/341	0.031	0.647	11/350
	5	0.058	1.000	16/275	0.039	0.588	10/254
73	1	1.000	0.300	3/3	0.000	0.000	0/1
	2	1.000	0.400	4/4	0.000	0.000	0/4
	3	1.000	0.500	5/5	0.000	0.000	0/2
	4	1.000	0.300	3/3	0.000	0.000	0/1
	5	0.069	1.000	10/145	0.038	0.545	6/159
157	1	1.000	0.300	6/6	0.000	0.000	0/2
	2	1.000	0.300	6/6	0.025	0.750	15/601
	3	1.000	0.250	5/5	0.000	0.000	0/5
	4	0.044	1.000	20/456	0.029	0.650	13/446
	5	0.041	1.000	20/484	0.027	0.650	13/488
220	1	1.000	0.500	5/5	0.014	1.000	10/699
	2	1.000	0.500	5/5	0.250	0.100	1/4
	3	1.000	0.300	3/3	0.000	0.000	0/0
	4	1.000	0.500	5/5	0.167	0.200	2/12
	5	0.093	1.000	10/107	0.042	0.500	5/118
225	1	1.000	0.583	7/7	0.000	0.000	0/4
	2	1.000	0.750	9/9	0.000	0.000	0/2
	3	1.000	0.667	8/8	0.000	0.000	0/2
	4	0.068	1.000	12/176	0.026	0.385	5/194
	5	0.074	1.000	12/162	0.021	0.231	3/140

It can be seen how Boolean PQs do not show a good retrieval performance on the test collections. They are either not able to retrieve relevant documents at all or output every relevant document together with a large number of irrelevant ones, thus having small precision values and making difficult the retrieval task for the user. However, it is very important to realize that *it is not enough to work with a more flexible and representative structure for the PQ (as the extended Boolean query) to achieve good results but it has to be constructed in a proper way*. Note how the Kraft et al.'s fuzzy PQs presenting higher recall in the test documents usually retrieve every document in the base, thus showing a very low precision (this happens in all queries but in number 220, where the latter process constructs a PQ with full recall and good precision), while the Boolean queries succesfully discriminate a larger number of documents, getting higher precision values. Hence, it is clear that the multiobjective evolutionary algorithm is playing a key role in the good retrieval performance of the derived fuzzy PQs.

Finally, we would like to focus the analysis on the test

[4]The common parameter values considered by the three evolutionary algorithms are a population size of 800 individuals, 50000 evaluations per run, a maximum of 20 nodes for the query tree sizes, a tournament size t of 10% of the population size, 0.8 and 0.2 for the crossover and mutation probabilities (in both the GA and the GP parts in the MOGA-P). The retrieval threshold σ has been set to 0.1 in the fuzzy IRS.

[5]As the index terms of the training and test documentary bases can be different, there is a need to translate training queries into test ones, removing those terms without a correspondence in the test set.

TABLE II

RETRIEVAL EFFICACY OF KRAFT ET AL.'S FUZZY PQS

		Training set			Test set		
$\#q$		P	R	rr/rt	P	R	rr/rt
1	1	1.000	0.500	7/7	0.000	0.000	0/1
	2	1.000	0.714	10/10	0.750	0.200	3/4
	3	1.000	0.643	9/9	0.667	0.133	2/3
	4	0.020	1.000	14/698	0.021	1.000	15/700
	5	0.020	1.000	14/698	0.021	1.000	15/700
2	1	1.000	0.750	9/9	0.750	0.231	3/4
	2	1.000	0.750	9/9	0.500	0.077	1/2
	3	1.000	0.667	8/8	0.667	0.308	4/6
	4	0.017	1.000	12/698	0.019	1.000	13/700
	5	0.017	1.000	12/698	0.019	1.000	13/700
23	1	1.000	0.500	8/8	0.500	0.059	1/2
	2	1.000	0.438	7/7	0.000	0.000	0/12
	3	1.000	0.500	8/8	0.000	0.000	0/10
	4	0.023	1.000	16/698	0.024	1.000	17/700
	5	0.023	1.000	16/698	0.024	1.219	17/700
73	1	1.000	0.600	6/6	1.000	0.455	5/5
	2	1.000	0.700	7/7	1.000	0.091	1/1
	3	1.000	0.900	9/9	0.556	0.455	5/9
	4	1.000	0.500	5/5	0.000	0.000	0/0
	5	0.435	1.000	10/23	0.059	0.156	1/17
157	1	1.000	0.500	10/10	0.083	0.050	1/12
	2	1.000	0.500	10/10	0.000	0.000	0/13
	3	0.923	0.600	12/13	0.077	0.200	4/52
	4	0.029	1.000	20/699	0.029	1.000	20/699
	5	0.029	1.000	20/699	0.029	1.000	20/699
220	1	1.000	0.800	8/8	0.200	0.100	1/5
	2	1.000	0.600	6/6	0.500	0.300	3/6
	3	1.000	0.900	9/9	0.250	0.200	2/8
	4	1.000	0.800	8/8	0.014	1.000	10/699
	5	0.106	1.000	10/94	0.120	1.000	10/83
225	1	1.000	0.667	8/8	0.000	0.000	0/2
	2	1.000	0.667	8/8	0.250	0.077	1/4
	3	1.000	0.667	8/8	0.000	0.000	0/0
	4	1.000	0.500	6/6	0.019	1.000	13/700
	5	0.207	1.000	12/58	0.036	0.154	2/55

TABLE III

RETRIEVAL EFFICACY OF MOGA-P FUZZY PQS

		Training set			Test set		
$\#q$		P	R	rr/rt	P	R	rr/rt
1	1	1.000	0.643	9/9	0.000	0.000	0/3
	2	0.786	0.786	11/14	0.143	0.067	1/7
	3	0.591	0.929	13/22	0.154	0.133	2/13
	4	0.318	1.000	14/44	0.111	0.267	4/36
	5	0.304	1.000	14/46	0.188	0.400	6/32
2	1	1.000	0.667	8/8	0.143	0.154	2/14
	2	1.000	0.667	8/8	0.143	0.154	2/14
	3	0.579	0.917	11/19	0.000	0.000	0/24
	4	0.387	1.000	12/31	0.216	0.615	8/37
	5	0.273	1.000	12/44	0.297	0.846	11/37
23	1	1.000	0.625	10/10	0.111	0.059	1/9
	2	0.786	0.688	11/14	0.455	0.294	5/11
	3	0.591	0.812	13/22	0.344	0.647	11/32
	4	0.390	1.000	16/41	0.208	0.588	10/48
	5	0.232	1.000	16/69	0.031	0.118	2/65
73	1	1.000	0.900	9/9	0.455	0.455	5/11
	2	0.769	1.000	10/13	0.062	0.091	1/16
	3	0.526	1.000	10/19	0.071	0.091	1/14
	4	0.500	1.000	10/20	0.208	0.455	5/24
	5	0.692	0.900	9/13	0.250	0.455	5/20
157	1	1.000	0.500	10/10	0.375	0.150	3/8
	2	0.789	0.750	15/19	0.250	0.150	3/12
	3	0.593	0.800	16/27	0.300	0.300	6/20
	4	0.390	0.800	16/41	0.119	0.250	5/42
	5	0.299	1.000	20/67	0.195	0.800	16/82
220	1	1.000	0.900	9/9	0.111	0.100	1/9
	2	0.714	1.000	10/14	0.600	0.300	3/5
	3	0.588	1.000	10/17	0.167	0.100	1/6
	4	0.588	1.000	10/17	0.167	0.100	1/6
	5	0.833	1.000	10/12	0.200	0.100	1/5
225	1	1.000	0.917	11/11	1.000	0.077	1/1
	2	0.688	0.917	11/16	0.000	0.000	0/8
	3	0.579	0.917	11/19	0.000	0.000	0/15
	4	0.324	1.000	12/37	0.000	0.000	0/33
	5	0.324	1.000	12/37	0.000	0.000	0/33

results of query 225, where the three groups of PQs actually perform very bad and the worst results are obtained by the MOGA-P (just one of the learned PQs succeds at retrieving a relevant document). In our opinion, this is since there is a larger diversity of index terms in the relevant documents for this Cranfield query, and hence it is more difficult for those index terms existing in the training documents to appropriately describe the test relevant documents. This would also explain the bad behavior of the MOGA-P in this case as its larger capability to adapt to the training document set will cause the usual machine learning overfitting problem to happen.

VI. CONCLUDING REMARKS

The use of soft computing tools to design PQs for text retrieval has been analyzed by constructing extended Boolean queries from sets of training documents extracted from the Cranfield collection. The multiobjective GA-P algorithm considered for this task has obtained better results than previous single-objective evolutionary algorithms for Boolean and extended Boolean queries, showing how a right use of soft computing tools allows an appropriate personalization of IRSs.

In our opinion, several future works arise from the present contribution. On the one hand, retrieval measures considering not only the absolute number of relevant and non relevant documents retrieved, but also their ranking in the retrieved document list have to be considered as they will help us to analyze the real performance of the fuzzy PQs. On the other hand, we think on using more advanced multiobjective evolutionary approaches than the basic MOGA technique. Besides other even more expressive PQ structures such as linguistic queries can be considered. Finally, we also plan to consider fuzzy aggregation operators to combine the retrieved document sets obtained from the different PQs derived from the MOGA-P algorithm into a single list, thus enhancing the user's ability to get relevant information as it would come from PQs with a different trade-off between precision and recall.

ACKNOWLEDGMENTS

This work was supported by the Spanish Ministerio de Ciencia y Tecnología under project TIC2003-00877, with part of the project budget coming from FEDER fundings.

APPENDIX

A. Fuzzy Information Retrieval Systems

The fuzzy IR model [4] was proposed to flexibilize Boolean IRSs with the aim of overcoming several of its limitations (such as the ranking of the retrieved document set) without a need of a complete redesign. Its main aspects are as follows:

Indexing: An indexing function $F : D \times T \rightarrow [0,1]$ is defined as a fuzzy relation mapping the degree to which document d belongs to the set of documents "about" the concept(s) represented by term t. By projecting it, a fuzzy set is associated to each document ($d_i = \{< t, \mu_{d_i}(t) > | \, t \in T\}$; $\mu_{d_i}(t) = F(d_i, t)$) and term ($t_j = \{< d, \mu_{t_j}(d) > | \, d \in D\}$; $\mu_{t_j}(d) = F(d, t_j)$). In this paper we will work with Salton's normalized *inverted document frequency* (IDF) [2].

Query subsystem: Fuzzy IRSs deal with a extended Boolean (fuzzy) query structure composed of weighted, positive or negative terms joined by the AND and OR operators. Weights allows the users to define selection conditions as soft constraints on the significance of the index terms in the document representations, while the modeling of the Boolean conjunctions as fuzzy operators allows us to increase the flexibility of the IRS.

Hence, the query subsystem affords a fuzzy set q defined on the document domain specifying the RSV of each document in the data base with respect to the processed query: $q = \{< d, \mu_q(d) > | \, d \in D\} \quad ; \quad \mu_q(d) = RSV_q(d)$.

Thus, documents can be ranked according to the membership degrees of relevance before being presented to the user, as in vector space IRSs. The retrieved document set can be specified providing an upper bound for the number of retrieved documents or defining a threshold σ for the RSV (the σ-cut of the query response fuzzy set q).

Matching mechanism: Several possibilities arise depending on the query weight interpretation considered [4]. In the *importance* interpretation, query weights represent the relative importance of each term in the query. When a single term query is logically connected to another by the AND or OR operators, the relative importance of the single term in the compound query is taken into account by associating a weight to it. To maintain the semantics of the query, this weighting has to take a different form according as the single term queries are ANDed or ORed. Therefore, assuming that A is a fuzzy term with assigned weight w, the following expressions are applied to obtain the fuzzy set associated to the weighted single term queries A_w (*disjunctive queries*) and A^w (*conjunctive ones*):

$$\begin{array}{c} A_w = \{< d, \mu_{A_w}(d) > | \, d \in D\} \\ \mu_{A_w}(d) = Min\ (w, \mu_A(d)) \end{array} \tag{2}$$

$$\begin{array}{c} A^w = \{< d, \mu_{A^w}(d) > | \, d \in D\} \\ \mu_{A^w}(d) = Max\ (1-w, \mu_A(d)) \end{array} \tag{3}$$

If the term is negated in the query, a negation function is applied to obtain the corresponding fuzzy set: $\overline{A} = \{< d, \mu_{\overline{A}}(d) > | \, d \in D\} \quad ; \quad \mu_{\overline{A}}(d) = 1 - \mu_A(d)$.

Finally, the RSV of the compound query is obtained by combining the single weighted term evaluations into a unique fuzzy set as follows:

$$\begin{array}{c} A\ AND\ B = \{< d, \mu_{A\ AND\ B}(d) > | \, d \in D\} \\ \mu_{A\ AND\ B}(d) = Min(\mu_A(d), \mu_B(d)) \end{array} \tag{4}$$

$$\begin{array}{c} A\ OR\ B = \{< d, \mu_{A\ OR\ B}(d) > | \, d \in D\} \\ \mu_{A\ OR\ B}(d) = Max(\mu_A(d), \mu_B(d)) \end{array} \tag{5}$$

REFERENCES

[1] T. Bäck, D.B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*, IOP Publishing–Oxford University Press, 1997.

[2] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Adisson, 1999.

[3] N.J. Belkin and W.B. Croft, "Information filtering and information retrieval: two sides of the same coin?," *Communications of the ACM*, vol. 35:12, pp. 50, 1998.

[4] G. Bordogna, P. Carrara, and G. Pasi, "Fuzzy approaches to extend Boolean information retrieval," in: P. Bosc, J. Kacprzyk (Eds.), *Fuzziness in Database Management Systems*, Springer, pp. 231–274, 1995.

[5] H. Chen et al., "A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, GAs, and SA," *Journal of the American Society for Information Science*, vol. 49:8, pp. 693–705, 1998.

[6] C.A. Coello, D.A. Van Veldhuizen, and G.B. Lamant, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, 2002.

[7] O. Cordón, F. Moya, and C. Zarco, "A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system," *Mathware & Soft Computing*, vol. 7:2-3, pp. 309–322, 2000.

[8] O. Cordón and E. Herrera-Viedma, "Editorial: special issue on soft computing applications to intelligent information retrieval on the internet," *International Journal of Approximate Reasoning*, vol. 34:2-3, pp. 89–95, 2003.

[9] O. Cordón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, and C. Zarco, "A review on the application of evolutionary computation to information retrieval," *International Journal of Approximate Reasoning*, vol. 34:2-3, pp. 241–264, 2003.

[10] O. Cordón, F. Moya, and C. Zarco, "Automatic learning of multiple extended Boolean queries by multiobjective GA-P algorithms," in: V. Loia, M. Nikravesh, and L.A. Zadeh (Eds.), *Fuzzy Logic and the Internet*, Springer, 2004.

[11] F. Crestani and G. Pasi (Eds.), *Soft Computing in Information Retrieval*, Physica-Verlag, 2000.

[12] W. Fan, M.D. Gordon, and P. Pathak, "Effective profiling of consumer information retrieval needs: a unified framework and empirical comparison," *Decision Support Systems*, 2004, to appear.

[13] W. Fan, M.D. Gordon, and P. Pathak, "An integrated two-stage model for intelligent information routing," submitted to *Decision Support Systems*, 2004.

[14] U. Hanani, B. Shapira and P. Shoval, "Information filtering: overview of issues, research and systems," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 203–259, 2001.

[15] L. Howard and D. D'Angelo, "The GA-P: a genetic algorithm and genetic programming hybrid," *IEEE Expert*, vol. 10:3, pp. 11–15, 1995.

[16] D.H. Kraft et al., "Genetic algorithms for query optimization in information retrieval: relevance feedback," in: E. Sanchez, T. Shibata, and L.A. Zadeh (Eds.), *Genetic Algorithms and Fuzzy Logic Systems*, World Scientific, pp. 155–173, 1997.

[17] M. Nikravesh, V. Loia, and B. Azvine, "Fuzzy logic and the internet (FLINT): Internet, world wide web and search engines," *Soft Computing*, vol. 6:5, pp. 287-299, 2002.

[18] D.W. Oard, G. Marchionini, "A conceptual framework for text filtering," *CS-TR-3643, University of Maryland*, College Park, 1996.

[19] G. Pasi, "Intelligent information retrieval: some research trends," in: J. Benítez, O. Cordón, F. Hoffmann, and R. Roy (Eds.), *Advances in Soft Computing. Engineering Design and Manufacturing*, Springer, pp. 157–171, 2003.

[20] M.P. Smith, M. Smith, "The use of GP to build Boolean queries for text retrieval through relevance feedback," *Journal of Information Science*, vol. 23:6, pp. 423–431, 1997.