Analyzing the Performance of a Multiobjective GA-P Algorithm for Learning Fuzzy Queries in a Machine Learning Environment*

Oscar Cordón[1], Enrique Herrera-Viedma[1], María Luque[1], Félix de Moya[2], and Carmen Zarco[3]

[1] Dept. of Computer Science and A.I. University of Granada.
18071 - Granada (Spain).
{ocordon,viedma,mluque}@decsai.ugr.es
[2] Dept. of Library and Information Science. University of Granada.
18071 - Granada (Spain).
felix@ugr.es
[3] PULEVA Food S.A. Camino de Purchil, 66. 18004 - Granada (Spain).
czarco@puleva.es

Abstract. The fuzzy information retrieval model was proposed some years ago to solve several limitations of the Boolean model without a need of a complete redesign of the information retrieval system. However, the complexity of the fuzzy query language makes it difficult to formulate user queries. Among other proposed approaches to solve this problem, we find the Inductive Query by Example (IQBE) framework, where queries are automatically derived from sets of documents provided by the user. In this work we test the applicability of a multiobjective evolutionary IQBE technique for fuzzy queries in a machine learning environment. To do so, the Cranfield documentary collection is divided into two different document sets, labeled training and test, and the algorithm is run on the former to obtain several queries that are then validated on the latter.

1 Introduction

Information retrieval (IR) may be defined as the problem of the selection of documentary information from storage in response to search questions provided by a user [2]. Information retrieval systems (IRSs) deal with documentary bases containing textual, pictorial or vocal information and process user queries trying to allow the user to access to relevant information in an appropriate time interval.

The fuzzy information retrieval (FIR) model [3] was proposed to overcome several limitations of Boolean IRSs [2], the most extended ones, without a need of a complete redesign. However, the extended Boolean (fuzzy) query structure considered in fuzzy IRSs – weighted, positive or negative terms joined by the AND and OR operators – is difficult to be formulated by non expert users.

* Research supported by CICYT TIC2002-03276 and by Project "Mejora de Metaheurísticas mediante Hibridación y sus Aplicaciones" of the University of Granada.

T. Bilgiç et al. (Eds.): IFSA 2003, LNAI 2715, pp. 611–619, 2003.

The paradigm of Inductive Query by Example (IQBE) [4], where queries describing the information contents of a set of documents provided by a user are automatically derived, has proven to be useful to assist the user in the query formulation process. Focusing on the FIR model, the most known approach is that of Kraft et al. [13], based on genetic programming (GP) [12]. Several other approaches have been proposed based on more advanced evolutionary algorithms (EAs) [1], such as genetic algorithm-programming (GA-P) [11] or simulated annealing-programming, to improve Kraft et al.'s [6,7].

In [9], we proposed a new IQBE algorithm that tackled fuzzy query learning as a multiobjective problem. The algorithm was able to automatically generate several queries with a different trade-off between precision and recall in a single run. To do so, a Pareto-based multiobjective EA scheme [5] was incorporated into the single-objective GA-P IQBE technique proposed in [6].

In this contribution, we design a experimental framework to test the said technique in a machine learning environment. To do so, several queries are selected from the Cranfield collection and the document set is divided into two different subsets, training and test, for each of them. The multiobjective GA-P algorithm is then run on the former sets and the obtained queries are validated on the latter ones to get a view of the real applicatibility of the approach.

The paper is structured as follows. Section 2 is devoted to the preliminaries, including the basis of FIRSs and a short review of IQBE techniques. Then, the multiobjective GA-P proposal is reviewed in Section 3. Section 4 presents the experimental setup designed and the experiments developed. Finally, several concluding remarks are pointed out in Section 5.

2 Preliminaries

2.1 Fuzzy Information Retrieval Systems

FIRSs are constituted of the following three main components:

The documentary data base, that stores the documents and their representations (typically based on index terms in the case of textual documents).

Let D be a set of documents and T be a set of unique and significant terms existing in them. 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]: $w_{d,t} = f_{d,t} \cdot log(N/N_t) \quad ; \quad F(d, t) = \frac{w_{d,t}}{Max_d \, w_{d,t}}$, where $f_{d,t}$ is the frequency of term t in document d, N is the total number of documents and N_t is the number of documents where t appears at least once.

The query subsystem, allowing the users to formulate their queries and presenting the retrieved documents to them. Fuzzy queries are expressed using a query language that is based on weighted terms, where the numerical or linguistic weights represent the "subjective importance" of the selection requirements.

In FIRSs, the query subsystem affords a fuzzy set q defined on the document domain specifying the degree of relevance (the so called *retrieval status value* (RSV)) of each document in the data base with respect to the processed query: $q = \{< d, \mu_q(d) > \mid d \in D\} \quad ; \quad \mu_q(d) = RSV_q(d)$.

Thus, documents can be ranked according to the membership degrees of relevance before being presented to the user. 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).

The matching mechanism, that evaluates the degree to which the document representations satisfy the requirements expressed in the query (i.e., the RSV) and retrieves those documents that are judged to be relevant to it.

When using the *importance* interpretation [3], the query weights represent the relative importance of each term in the query. The RSV of each document to a fuzzy query q is then computed as follows [15]. 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*):

$$A_w = \{< d, \mu_{A_w}(d) > \mid d \in D\} \qquad ; \qquad \mu_{A_w}(d) = Min\ (w, \mu_A(d))$$
$$A^w = \{< d, \mu_{A^w}(d) > \mid d \in D\} \qquad ; \qquad \mu_{A^w}(d) = Max\ (1 - w, \mu_A(d))$$

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

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

$$A\ AND\ B = \{< d, \mu_{A\ AND\ B}(d) > \mid d \in D\} \ ; \ \mu_{A\ AND\ B}(d) = Min(\mu_A(d), \mu_B(d))$$
$$A\ OR\ B = \{< d, \mu_{A\ OR\ B}(d) > \mid d \in D\} \ ; \ \mu_{A\ OR\ B}(d) = Max(\mu_A(d), \mu_B(d))$$

2.2 Inductive Query by Example

IQBE was proposed in [4] as "a process in which searchers provide sample documents (examples) and the algorithms induce (or learn) the key concepts in order to find other relevant documents". This way, IQBE is a technique for assisting the users in the query formulation process performed by machine learning methods. It works by taking a set of relevant (and optionally, non relevant documents) provided by a user and applying an off-line 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.

Apart from the IQBE algorithms for the FIR model reviewed in the Introduction, several others have been proposed for the remaining IR models, such as the Boolean [16,8] or the vector space [4] ones.

3 A Multiobjective GA-P Algorithm for Automatically Learning Fuzzy Queries

In [9], we proposed a multiobjective IQBE algorithm to learn fuzzy queries based on the GA-P paradigm whose components are described next.

Coding Scheme: The expressional part (GP part) encodes the query composition – terms and logical operators – and the real-coded coefficient string (GA part) represents the term weights, as shown in Figure 1.

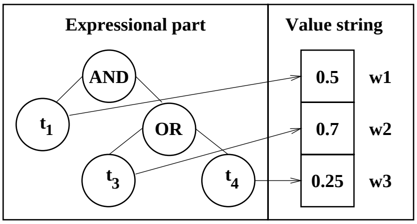

Fig. 1. GA-P individual representing the fuzzy query 0.5 t_1 AND (0.7 t_3 OR 0.25 t_4)

Fitness Function: The multiobjective GA-P (MOGA-P) algorithm is aimed at jointly optimizing the classical precision and recall criteria [2], as follows:

$$Max\ P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \quad ; \quad Max\ R = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d}$$

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.

Pareto-Based Multiobjective Selection and Niching Scheme: The Pareto-based multiobjective EA considered is Fonseca and Fleming's Pareto-based MOGA [5]. Each individual is first assigned a rank equal to the number of individuals dominating it plus one (non-dominated individuals receive rank

1) and the population is sorted in ascending order according to that rank. Then, each individual C_i is assigned a fitness value according to its ranking in the population: $f(C_i) = \frac{1}{rank(C_i)}$. Finally, the fitness assignment of each group of individuals with the same rank is averaged among them.

A niching scheme is then applied in the objective space to obtain a well-distributed set of queries with a different trade-off between precision and recall. To do so, Goldberg and Richardson's sharing function [14] is considered:

$$F(C_i) = \frac{f(C_i)}{\sum_{j=1}^{M} Sh(d(C_i, C_j))} \quad ; \quad Sh(x) = \begin{cases} 1 - (\frac{x}{\sigma_{share}})^{\gamma}, & \text{if } x < \sigma_{share} \\ 0, & \text{otherwise} \end{cases}$$

with M being the population size, σ_{share} being the niche radius and d standing for the Euclidean distance.

Finally, the intermediate population is obtained by Tournament selection [14], i.e., to fill each free place in the new population, t individuals are random selected from the current one and the best adapted of them is chosen.

Genetic Operators: The BLX-α crossover operator [10] is applied twice on the GA part to obtain two offsprings. Michalewicz's non-uniform mutation operator [14] is considered to perform mutation on that part.

The usual GP crossover randomly selecting one edge in each parent and exchanging both subtrees from these edges between the both parents [12] is considered. Each time a mutation is to be made, one of the two following mutation operators (selected at random) is applied: random generation of a new subtree substituting an old one located in a randomly selected edge, and random change of a query term by another one not present in the encoded query, but belonging to any relevant document.

4 Experimental Setup and Experiments Developed

The documentary set used to design our machine learning experimental framework has been the *Cranfield* collection, composed of 1400 documents about Aeronautics. 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 the normalized IDF scheme (see Section 2.1) to generate the term weights in the document representations.

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.

For each one of these queries, the documentary collection has been divided into two different, non overlapped, document sets, training and test, each of them composed of a 50% of both the relevant and non relevant documents. Hence, we represent a retrieval environment where no document retrieved by the learned queries in the test sets has been previously seen by the user.

MOGA-P has been run ten different times on the training document set associated to each query. The parameter values considered are a maximum of 20 nodes for the expression parts, a population size of 800, 50000 evaluations per run, a Tournament size t of 8, 0.8 and 0.2 for the crossover and mutation probabilities in both the GA and the GP parts, a sharing function parameter γ equal to 2, and a niche radius σ_{share} experimentally set to 0.1. The retrieval threshold σ has been set to 0.1 in the FIRS.

Table 1. Statistics of the Pareto sets obtained by the MOGA-P algorithm

$\#q$	$\#p$	$\sigma_{\#p}$	$\#dp$	$\sigma_{\#dp}$	M_2^*	$\sigma_{M_2^*}$	M_3^*	$\sigma_{M_3^*}$
1	110.0	10.5	5.3	0.348	39.901	4.574	0.918	0.044
2	127.4	7.462	4.3	0.202	47.023	3.235	0.895	0.033
23	133.8	5.805	6.9	0.170	52.156	3.004	1.042	0.015
73	93.0	12.435	2.6	0.210	24.893	5.133	0.730	0.041
157	118.9	7.886	7.8	0.310	45.264	3.943	1.066	0.006
220	91.1	6.897	1.9	0.221	18.987	4.395	0.437	0.094
225	98.1	6.243	2.3	0.202	22.931	4.266	0.626	0.083

Table 1 collects several statistics about the ten Pareto sets generated for each query. From left to right, the columns contain the number of non-dominated solutions obtained ($\#p$), the number of different objective vectors (i.e., precision-recall pairs) existing among them ($\#dp$), and the values of two of the usual multiobjective metrics $\mathcal{M}_2^*$ and $\mathcal{M}_3^*$ [17][1], all of them followed by their respective standard deviation values.

In order to test the real applicability of the algorithm in the machine learning environment, the Pareto sets obtained in the ten runs performed for each query were put together, and the dominated solutions were removed from the unified set. Then, five queries well distributed on the Pareto front were selected from each of the seven unified Pareto sets[2] and run on the corresponding test set once preprocessed (for example, the query terms not existing on the test collection are removed from the query). The results obtained are shown in Table 2, standing Sz for the query size, P and R for the precision and recall values and $\#rr/\#rt$ for the number of relevant and retrieved documents, respectively.

In view of the precision and recall values obtained, the performance of our proposal is very significant[3]. The algorithm is always able to find at least a

[1] $\mathcal{M}_2^* \in [0, \#p]$ measures the diversity of the solutions found, while $\mathcal{M}_3^*$ measures the range to which the Pareto front spreads out in the objective values (in our case, the maximum possible value is $\sqrt{2} = 1.4142$). In both cases, the higher the value, the better the quality of the obtained Pareto set.

[2] An example of such a query (#q1-5) in preorder is: `OR AND OR` t_{1158}`(w=0.298) OR` t_{1051}`(w=0.518) OR` t_{2721}`(w=0.957) OR` t_{2950}`(w=0.838)` t_{12}`(w=0.970) OR OR` t_{1320}`(w=0.577)` t_{238}`(w=0.847)` t_{2579}`(w=0.737) OR` t_{1129}`(w=0.701)` t_{12}`(w=0.329).`

[3] The interested reader can refer to [6,7,8,9] to compare the obtained results with those of several other approaches in the same documentary base.

Table 2. Retrieval efficacy of the selected queries on the training and test collections

		Training set				Test set			
$\#q$		Sz	P	R	$\#rr/\#rt$	Sz	P	R	$\#rr/\#rt$
1	1	19	0.304	1.0	14/46	9	0.188	0.4	6/32
	2	19	0.318	1.0	14/44	19	0.111	0.267	4/36
	3	19	0.591	0.929	13/22	5	0.154	0.133	2/13
	4	19	0.786	0.786	11/14	5	0.143	0.067	1/7
	5	19	1.0	0.643	9/9	15	0.0	0.0	0/3
2	1	19	0.273	1.0	12/44	19	0.297	0.846	11/37
	2	19	0.387	1.0	12/31	19	0.216	0.615	8/37
	3	19	0.579	0.917	11/19	17	0.0	0.0	0/24
	4	19	0.786	0.917	11/14	15	0.059	0.077	1/17
	5	19	1.0	0.667	8/8	17	0.143	0.154	2/14
23	1	19	0.232	1.0	16/69	19	0.031	0.118	2/65
	2	19	0.39	1.0	16/41	17	0.208	0.588	10/48
	3	19	0.591	0.812	13/22	19	0.344	0.647	11/32
	4	19	0.786	0.688	11/14	15	0.455	0.294	5/11
	5	19	1.0	0.625	10/10	19	0.111	0.059	1/9
73	1	19	0.692	0.9	9/13	17	0.25	0.455	5/20
	2	19	0.5	1.0	10/20	15	0.208	0.455	5/24
	3	19	0.526	1.0	10/19	15	0.071	0.091	1/14
	4	19	0.769	1.0	10/13	17	0.062	0.091	1/16
	5	19	1.0	0.9	9/9	17	0.455	0.455	5/11
157	1	19	0.299	1.0	20/67	15	0.195	0.8	16/82
	2	19	0.39	0.8	16/41	19	0.119	0.25	5/42
	3	19	0.593	0.8	16/27	17	0.3	0.3	6/20
	4	19	0.789	0.75	15/19	15	0.25	0.15	3/12
	5	19	1.0	0.5	10/10	15	0.375	0.15	3/8
220	1	19	0.833	1.0	10/12	13	0.2	0.1	1/5
	2	19	0.588	1.0	10/17	13	0.167	0.1	1/6
	3	19	0.588	1.0	10/17	15	0.167	0.1	1/6
	4	17	0.714	1.0	10/14	13	0.6	0.3	3/5
	5	19	1.0	0.9	9/9	19	0.111	0.1	1/9
225	1	17	0.324	1.0	12/37	15	0.0	0.0	0/33
	2	17	0.324	1.0	12/37	15	0.0	0.0	0/33
	3	19	0.579	0.917	11/19	11	0.0	0.0	0/15
	4	19	0.688	0.917	11/16	15	0.0	0.0	0/8
	5	19	1.0	0.917	11/11	15	1.0	0.077	1/1

query retrieving all the relevant documents ($R = 1.0$) provided by the user in the training set. As regards the generalization ability of the learned queries, i.e., their capability to retrieve new relevant documents for the user, it can be seen how it is very satisfactory in the most of the cases. For example, recall levels of 0.4, 0.846, 0.647, 0.455, 0.8 and 0.3 are respectively obtained for queries 1, 2, 23, 73, 157, and 220, all of them with appropriate precision values ranging from 0.188 to 0.6. However, we should note that the results obtained in the last query, 225, have not been appropriate as only one of the learned queries has been able to retrieve a relevant document in the test set. In this case, it seems that there is a larger diversity of index terms in the relevant documents for the query, and those index terms existing in the training documents do not describe the test relevant documents.

5 Concluding Remarks

The real applicability of a multiobjective evolutionary IQBE technique for learning fuzzy queries has been tested in a machine learning environment. It has been run on training document sets obtained from the Cranfield collection to derive several queries that have been validated on a different test document set. Very promising results have been achieved for six of the seven Cranfield queries considered in view of the retrieval efficacy obtained.

As future works, we will study other real-like environments based on different training-test partitions of the document collection and will use retrieval measures considering not only the absolute number of relevant and non relevant documents retrieved, but also their relevance order in the retrieved document list.

References

1. Bäck, T.: Evolutionary algorithms in theory and practice. Oxford (1996).
2. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. Addison (1999).
3. Bordogna, G., Carrara, P., Pasi, G.: Fuzzy approaches to extend Boolean information retrieval. In: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in database management systems. Physica-Verlag (1995) 231–274.
4. Chen, H., 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 **49:8** (1998) 693–705.
5. Coello, C.A., Van Veldhuizen, D.A., Lamant, G.B.: Evolutionary algorithms for solving multi-objective problems. Kluwer Academic Publishers (2002).
6. Cordón, O., Moya, F., Zarco, C.: A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system, Mathware & Soft Computing **7:2-3** (2000) 309–322.
7. Cordón, O., Moya, F., Zarco, C.: A new evolutionary algorithm combining simulated annealing and genetic programming for relevance feedback in fuzzy information retrieval systems, Soft Computing **6:5** (2002) 308-319.
8. Cordón, O., Herrera-Viedma, E., Luque, M.: Evolutionary learning of Boolean queries by multiobjective genetic programming. In: Proc. PPSN-VII, Granada, Spain, LNCS 2439. Springer (September, 2002) 710-719.
9. Cordón, O., Moya, F., Zarco, C.: Automatic learning of multiple extended Boolean queries by multiobjective GA-P algorithms. In: V. Loia, M. Nikravesh, L.A. Zadeh (Eds.), Fuzzy Logic and the Internet. Springer (2003), in press.

10. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: L.D. Whitley (Ed.), Foundations of Genetic Algorithms 2. Morgan Kaufman (1993) 187–202.
11. Howard, L., D'Angelo, D.: The GA-P: a genetic algorithm and genetic programming hybrid, IEEE Expert **10:3** (1995) 11–15.
12. Koza, J.: Genetic programming. On the programming of computers by means of natural selection. The MIT Press (1992).
13. Kraft, D.H., et al.: Genetic algorithms for query optimization in information retrieval: relevance feedback. In: E. Sanchez, T. Shibata, L.A. Zadeh, Genetic algorithms and fuzzy logic systems. World Scientific (1997) 155–173.
14. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs. Springer (1996).
15. Sanchez, E.: Importance in knowledge systems, Information Systems **14:6** (1989) 455–464.
16. Smith, M.P., Smith, M.: The use of GP to build Boolean queries for text retrieval through relevance feedback, Journal of Information Science **23:6** (1997) 423–431.
17. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results, Evolutionary Computation **8:2** (2000) 173–195.