Oscar Cordón[1], Enrique Herrera-Viedma[1], María Luque[1], Félix Moya[2], Carmen Zarco[3]
[1] Dept. of Computer Science and A.I. University of Granada. 18071-Granada
[2] Dept. of Librarianship. Faculty of Librarianship and Documentation. 18071-Granada
[3] PULEVA Salud S.A. Camino de Purchil, 66. 18004 - Granada

An Inductive Query by Example Technique for Extended Boolean Queries Based on Simulated-Annealing Programming

Abstract: One of the key problems that non-expert users have to deal with when using an Information Retrieval System is the need to deeply know its query language in order to express their information needs in the form of a valid query allowing them to retrieve relevant information. To solve this problem, Inductive Query by Example Techniques can be considered to automatically derive queries from a set of relevant documents provided by a user. In this paper, a new hybrid evolutionary technique is proposed to automatically learn extended Boolean queries and is compared to Kraft et al.'s approach in several queries of the well known Cranfield collection.

1. Introduction

Information retrieval (IR) may be defined, in general, as the problem of the selection of documentary information from storage in response to search questions provided by an user (Salton and McGill, 1984). Information retrieval systems (IRSs) are a kind of information system that deal with data bases composed of information items —documents that may consist of textual, pictorial or vocal information— and process user queries trying to allow the user to access to relevant information in an appropriate time interval.

Most of the commercial IRSs are based on the Boolean model (van Rijsbergen, 1979), which presents some limitations. Due to this fact, some paradigms have been designed to extend this retrieval model and overcome its problems, such as the vector space (Salton and McGill, 1984) or the fuzzy information retrieval (FIR) models (Bordogna et al., 1995), (Croft, 1994).

However, the increase in the power of the retrieval model also comes with a high complexity augment in the query language, what makes difficult for the user to represent his information needs in the form of a valid query. This is especially significant in the case of fuzzy IRSs, whose query language allows us to formulate weighted Boolean (fuzzy) queries where the query terms are joined by the logical operators AND and OR. If it is difficult for a human user to formulate a classical Boolean query due to the need to know how to properly connect the query terms together using the Boolean operators, it will be even more difficult to both define the query structure and specify the query terms weights to retrieve the desired documents.

Hence, the paradigm of *Inductive Query by Example* (IQBE) (Chen, 1998), where a query describing the information contents of a set of documents provided by a user is automatically derived, can be useful to solve this problem and assist the user in the query formulation process. Focusing on the FIR model, the most known existing approach is that of Kraft et al.'s (1997), which is based on genetic programming (GP) (Koza, 1992).

In this paper, a new IQBE technique for FIRSs based on a hybrid simulated annealing-genetic programming evolutionary algorithm will be introduced with the aim of improving the performance of Kraft et al.'s proposal in terms of retrieval accuracy. To do so, the paper is structured as follows. Section 2 is devoted to the preliminaries, briefly presenting the basis of FIRSs and of IQBE techniques. Then, Kraft et al.'s proposal is reviewed in Section 3. Section 4 presents the composition of the new algorithm proposed while the experiments developed to test it are showed in Section 5. Finally, several conclusions are pointed out in Section 6.

2. Preliminaries

2.1. Fuzzy Information Retrieval

FIRSs permits to deal with the uncertainty and imprecision existing in the retrieval process by Extending classical Boolean IRSs in the three following aspects (Bordogna et al., 1995), (Croft, 1994):

- Indexing terms do not absolutely describe (1) or do not describe at all (0) the document contents in the document representations, but they have a partial degree of aboutness between [0,1]. The indexing function F: D×T→ [0,1] is defined as a two-dimensional fuzzy set (a fuzzy relation), that is projected to obtain a fuzzy set associated to each document and term:

$$d_i = \{< t, \mu_{d_i}(t) > | t \in T\} \;\; ; \;\; \mu_{d_i}(t) = F(d_i, t)$$
$$t_j = \{< d, \mu_{t_j}(d) > | d \in D\} \;\; ; \;\; \mu_{t_j}(d) = F(d, t_j)$$

  In this paper, we will work with the normalized *inverted document frequency* (Salton, 1984) to define the indexing function F.
- Document retrieval also becomes a matter of degrees and the relevance of a document to a query is also measured in [0,1]. This allows the FIRS to rank the retrieved documents as regards their relevance to the query as in the vector space model.
- Finally, the query structure is extended by associating a numerical or linguistic weight to each query term. These weights can be interpreted in different ways, e.g., relative importance among the terms involved in the query can be appropriately expressed.

In this contribution we consider the relative importance interpretation for the query weights. For the operation mode of the FIRS matching mechanism when considering this sentence, as well as for a description of the remaining two approaches, the interested reader can refer to (Bordogna et al., 1995), (Croft, 1994).

2.2. Inductive Query by Example

IQBE was proposed in (Chen,1998) as “*a process in which searchers provide sample documents (examples) and the algorithms induce (or learn) the key concepts in order to find other relevants documents*”. This way, IQBE is a process for assisting the users in the formulation process performed by machine learning methods (Mitchell, 1997). It works by taking a set of relevant (and optionally, non relevant documents) provided by a user —that can be obtained from a preliminary query or from a browsing process in the documentary base— and applying an off-line learning process to automatically generate a query describing the user’s needs (as represented by the document set provided by him). The obtained query can then be run in other IRSs to obtain more relevant documents. This way, there is no need that the user interact with the process as in other query refinement techniques as relevance feedback.

All of the machine learning methods considered in that paper a (regression trees, genetic algorithms and simulated annealing) dealt with the vector space model. Besides, Smith and Smith (1997) propose a Boolean query learning process based on GP. As regards the applications in FIRSs, we find the GP algorithm of Kraft et al. that will be reviewed in the next section and our niching GA-P method that extends the latter considering a more sophisticated evolutionary algorithm (Cordón et al., 2000). For descriptions of those of the previous techniques based on evolutionary algorithms refer to (Cordón et al., 1999).

3. The Kraft et al.'s IQBE Process for Extended Boolean Queries

The IQBE technique of Kraft et al. (1997) is a GP algorithm with the following composition:

- *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, where an intermediate population is created from the current one by means of Baker's stochastic universal sampling (Baker, 1987), together with the elitist selection.
- *Genetic Operators*: The usual GP crossover is considered (Koza, 1992), which is based on randomly selecting one edge in each parent and exchanging both subtrees from these edges between the both parents.

On the other hand, 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).

- *Fitness function*: The following function combining the classical precision and recall measures (for more information about them, see (van Rijsbergen, 1979) is considered:

$$\mathrm{F} = \alpha \cdot \frac{\sum_d \mathrm{r_d} \cdot \mathrm{f_d}}{\sum_d \mathrm{f_d}} + \beta \cdot \frac{\sum_d \mathrm{r_d} \cdot \mathrm{f_d}}{\sum_d \mathrm{r_d}}$$

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. Moreover, as simple queries are always prefered by the user, a selection criterion has been incorporated to the algorithm in order to consider more fitted those queries with a lesser complexity among a group of chromosomes with the same fitness value.

4. A New IQBE Process for Extended Boolean Queries Based on SA-P

Kraft et al.'s IQBE algorithm introduced in the previous section can perform well but it suffers from a key limitation of GP: it is very difficult to find proper values for the numerical weights as they are only altered by mutation during the evolution process. Hence, fuzzy queries with the optimal structure can be discarded by the selection procedure as the term weights involved in them are not well adjusted.

The latter problem can be solved by concurrently adapting both the query structure and the term weights (as done by the GA-P algorithm proposed in (Cordón et al., 2000)). In this paper, we do so by a hybrid evolutionary algorithm between Simulated Annealing (SA)[2] and GP, the SA-P paradigm, proposed in (Sánchez et al, 2001). The algorithm was based on encoding both a expressional part (the parse tree) and a value string (the coefficients involved in the expression) and adapt it within an usual SA search scheme by a neighborhood operator based on the classical GP crossover and a string value mutation operator.

Hence, the extended Boolean query is encoded by storing the query structure —terms and logical operators— in the expresional part, and the term weights in the value string using a real coding scheme. The neighboorhood operator (macromutation) generates the candidate fuzzy query by either changing the expresional part —the query structure— or the value string —the query weights— of the current individual $I$. This decision is randomly made with respect to a value string mutation probability *p* (*p*=1 means “only weight mutation” while *p*=0 means “only query structure mutation”). The query structure is mutated by selecting an edge of its parse tree and substituting the subtree located at it by a randomly generated parse tree. The weight vector is mutated by applying intermediate recombination (Mühlenbein and Schlierkamp-Voosen, 1993) between the current values (*weights(I)*) and a randomly generated vector *W* with an amplitude parameter that depends on the current temperature *T* by a constant K_1 as follows: $weights(I)= weights(I)\cdot(T/K_1) + (1-(T/K_1))\cdot W$.

The SA-P algorithm considered is showed as follows:

algorithm IQBE SA-P
needs: *MaxEval /*maximum number of evaluations*/, MaxNeighs /*max. number of neighbors generated per temperature*/, MaxSuccess /*max. number of neighbors accepted per temperature*/, c /*cooling factor*/,* T_0 */*initial temperature*/, p /*value string mutation probability*/,* K_1 */*value string mutation parameter*/*
produces: *Ibest*

```
I=Ibest=random individual;     $T=T_0$;   Eval=1
while (Eval<=MaxEval) do
        num_neighs=num_success=0
        while (num_neighs<MaxNeighs) && (num_success<MaxSuccess)
                Icand=macromutation(I,p,T,K_1); num_neighs=num_neighs+1
                delta=F(I)-F(Icand);      Eval=Eval+1
                v=random value with uniform distribution U(0,1)
                if (delta<0) or (v<exp(-delta/T)) then
                        I=Icand;          num_success=num_success+1
                        if (I>Ibest) then Ibest=I end if
                end if
        end while
        T=c*T
end while
```

As seen in the algorithm, the initial solution is a random fuzzy query. The initial temperature T_0 is computed by means of the following expression: $T_0 = (\mu/-\ln(\phi))\cdot F(I)$, with *I* being the initial solution and ϕ being the probability of acceptance for a solution that can be μ per 1 worse than F(*I*). Both parameters are defined in the interval [0,1].

5. Experiments and Analysis of Results

We have worked with the well known *Cranfield* documentary base to test the performance of our proposal. The 1400 documents have been automatically indexed by first extracting the non-stop words, thus obtaining a total number of 3857 different indexing terms, and then using the normalized IDF scheme to generate the term weights in the document representations. Among the 225 queries associated to this 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.

Both our proposal and Kraft et al.'s algorithm have been run on the previous relevant document sets. In order to make a fair comparison, both algorithms have been run three times with different initializations during the same fixed number of fitness function evaluations (100000). For the sake of simplicity, only the experiments not considering the use of the NOT operator are reported (as done in (Kraft et al., 1997)).

The common parameter values considered are a maximum of 20 nodes for the expression parts, (1.2, 0.8) for the fitness function weights α and β, and 0.1 for the FIRS retrieval threshold σ. Kraft et al.'s algorithm is run with a population of 1600 queries and 0.8 and 0.2 for the crossover and mutation probabilities respectively. Finally, for the SA-P algorithm, the initial temperature computation parameters μ and ϕ are set to 0.5, the maximum number of neighbors generated and accepted per temperature are respectively 500 and 50, the cooling parameter is set to 0.9, the value string mutation probability p takes value 0.5 and the parameter K_1 considered for this mutation is set to 5.

The best result obtained by Kraft et al.'s and our method in each of the seven queries are respectively showed in Tables 2 and 3, where *#q* refers to the query number, *Run* stands for the corresponding algorithm run (1 to 3), *T* for the run time (both algorithms have been run in a 350 Mhz. Pentium II computer with 64 MB of memory, and the time is measured in minutes), *Sz* for the generated query size, *Fit* for the fitness value, *P* and *R* for the precision and recall values, respectively, *#rt* for the number of documents retrieved by the query, and *#rr* for the number of relevant documents retrieved.

#q	Run	T	Sz	Fit	P	R	#rr/#rt
1	1,2	12:48	19	1.282759	1.000000	0. 103448	3/3
2	1,2	12:44	17	1.328000	1.000000	0.160000	4/4
23	1,2,3	12:48	17	1.272727	1.000000	0.090909	3/3
73	3	12:49	17	1.504762	1.000000	0.380952	8/8
157	1,2	12:45	19	1.260000	1.000000	0.075000	3/3
220	1,2	12:40	17	1.400000	1.000000	0.250000	5/5
225	1,2,3	12:41	19	1.328000	1.000000	0.160000	4/4

Table 2: Results obtained by Kraft et al.'s method in the Cranfield collection

#q	Run	T	Sz	Fit	P	R	#rr/#rt
1	2,3	13:04	19	1.393103	1.000000	0.241379	7/7
2	2,3	12:39	17	1.488000	1.000000	0.360000	9/9
23	1,2	13:02	19	1.369697	1.000000	0.212121	7/7
73	1,2,3	12:38	19	1.619048	1.000000	0.523810	11/11
157	1,3	13:28	19	1.380000	1.000000	0.225000	9/9
220	3	13:40	19	1.640000	1.000000	0.550000	11/11
225	1	13:18	19	1.520000	1.000000	0.400000	10/10

Table 3: Results obtained by our SA-P IQBE method in the Cranfield collection

In view of these results, it is clear that our SA-P IQBE algorithm significantly outperforms Kraft et al.'s proposal in all the cases. The lower improvement corresponds to query 73 with a 37.5 percent (a recall value of 0.523810 for the SA-P against to 0.380952 resulting from Kraft et al.'s algorithm), and the highest one to query 157 with a 300 percent (a recall value of 0.225 for the SA-P against another of 0.075 got from

Kraft et al.'s method).

As regards the computation time required, both methods take approximately the same, and the SA-P seems to be a little bit slower. We think that this is a consequence of the inclusion of a selection criterion to get simpler queries in Kraft’s method (see Section 3) which is not considered in the SA-P. This criterion makes the population being composed of simpler queries while the EA is converging, thus making their evaluation —the more time consuming procedure in both algorithms— less demanding.

6. Concluding Remarks

A new IQBE for FIRSs based on a hybrid SA-GP algorithm has been proposed and tested against the well known Kraft et al.’s proposal, outperforming the latter in the 1400 document Cranfield collection in terms of retrieval performance.

An extension of our method allowing it to adapt the retrieval threshold, which is usually a fixed value provided by the user, is also proposed in (Cordón et al., 2002). This increases even more the system effectiveness without augmenting the algorithm run time at all.

Notes

[1] Notice that the composition of several components is not the original one proposed by Kraft et al. but they have been changed in order to improve the algorithm performance. Of course, the basis of the algorithm have been maintained.

[2] SA (Aarts, 1989) is a neighborhood search algorithm which modifies the usual acceptance criteria of the basic local search sometimes permitting accepting a worse solution than the current one to avoid getting trapped in local optima. SA starts from an initial solution and then generates a new candidate solution (close to it) by applying random changes on it. If the candidate solution is better than the current one, then the former replaces the latter. Otherwise, it still could be randomly accepted with a probability that depends on the difference between both solutions and on a parameter called temperature. This temperature is initiated to a high value (meaning that significantly worse candidate solutions are likely to be accepted) and then this value is decreased by a procedure called cooling strategy each time a number of neighbors are generated.

References

Aarts, E.H.L. (1989). Simulated Annealing and Boltzman Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing. Wiley.

Baker, J.E. (1987). Reducing Bias and Inefficiency in the Selection Algorithm. Proc. Second International Conference on Genetic Algorithms (ICGA'87), Hillsdale, USA, pp. 14-21.

Bordogna, G., Carrara, P. & Pasi, G. (1995). Fuzzy Approaches to Extend Boolean Information Retrieval. In: P. Bosc, J. Kacprzyk (Eds.), Fuzziness in Database Management Systems, pp. 231--274.

Chen, H. (1998). A Machine Learning Approach to Inductive Query by Examples: An Experiment using Relevance Feedback, ID3, Genetic Algorithms, and Simulated Annealing. *JASIS*, 49(8): 693-705.

Cross, V. (1994). Fuzzy Information Retrieval. *Journal of Intelligent Information Systems*, 3: 29-56.

Cordón, O., Moya, F., Zarco, C. (1999). A Brief Study on the Application of Genetic Algorithms to Information Retrieval (in spanish). Proc. Fourth ISKO Conference (EOCONSID'99), Granada, Spain, pp. 179-186.

Cordón, O., Moya, F., Zarco, C. (2000). A GA-P Algorithm to Automatically Formulate Extended Boolean Queries for a Fuzzy Information Retrieval System. *Mathware & Soft Computing*, 7(2-3):309-322.

Cordón, O., Moya, F., Zarco, C. (2002). A New Evolutionary Algorithm Combining SA and GP for Relevance Feedback in Fuzzy Information Retrieval Systems. *Soft Computing*, 6(5).

Koza, J. (1992). Genetic Programming. On the Programming of Computers by means of Natural Selection. The MIT Press.

Kraft, D.H., Petry, F.E., Buckles, B.P. & Sadasivan, T. (1997). 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, pp. 155-173.

Mitchell, T.M. (1997). Machine Learning. McGraw-Hill.

Mühlenbein, H., Schlierkamp-Voosen, D. (1993). Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization. *Evolutionary Computation*,1(1):25-49.

Salton, G. & McGill, M.J. (1989). Introduction to Modern Information Retrieval. McGraw-Hill.

Sánchez, L., Couso, I., Corrales, J.A. (2001). Combining GP Operators with SA Search to Evolve Fuzzy Rule Based Classifiers. *Information Sciences*, 136(1-4):175-191.

Smith, M.P., Smith, M. (1997). The Use of Genetic Programming to Build Boolean Queries for Text Retrieval Through Relevance Feedback. *Journal of Information Science*, 23(6): 423-431.

van Rijsbergen, C.J. (1979). Information Retrieval (2nd edition), Butterworth.