Multi-objective Query Optimization Using Topic Ontologies

Rocío L. Cecchini† Carlos M. Lorenzetti‡ Ana G. Maguitman‡

Depto de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur
Av. Alem 1253 - (8000) Bahía Blanca, Argentina
† LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica
‡ LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
e-mail: {rlc,cml,agm}@cs.uns.edu.ar

Abstract. Formulating search queries based on a thematic context is a challenging problem due to the large number of combinations in which terms can be used to reflect the topic of interest. This paper presents a novel approach to learn topical queries that simultaneously satisfy multiple retrieval objectives. The proposed method consists in using a topic ontology to train an Evolutionary Algorithm that incrementally moves a population of queries towards the proposed objectives. We present an analysis of different single- and multi-objective strategies, discuss their strengths and limitations and test the most promising strategies on a large set of labeled Web pages. Our evaluations indicate that the tested strategies that apply multi-objective Evolutionary Algorithms are significantly superior to a baseline approach that attempts to generate queries directly from a topic description.

Keywords: topic ontologies, topical queries, semantic similarity, multi-objective evolutionary algorithms, query effectiveness.

1 Introduction

Reflecting a topic of interest in a query is an important research problem in the area of topic-based search. Automatic topical search can be achieved by automatically formulating queries with terms extracted from a thematic context. Applications for topical search can be built on top of existing search interfaces and can be used in different scenarios, such as searching for information in context [3], fulfilling long term information needs [25], collecting resources for topical Web portals [5], or accessing the Deep Web [13], among others.

From a theoretical perspective, the problem of topical search can be seen as an optimization problem where the objective function to be maximized quantifies the optimality of a query. In this optimization problem, therefore, the search space is defined as the set of possible queries that can be presented to a search interface. Multiple objectives such as high precision and high recall can be used as criteria for evaluating query performance.

Unfortunately, dealing effectively with the problem of topical search is very hard (both theoretically and computationally). If the query is too specific, the response could be empty or it may contain too few documents. On the other hand, if the query is too broad, the answer set may be too large and the most useful material will be hard to identify. In addition, the problem of query optimization does not have optimal substructure, which means that an optimal solution cannot be constructed efficiently from optimal solutions to its subproblems [7]. Therefore, existing methods to solve complex problems by breaking them down into simpler steps are not effective for our purpose. In addition, we may be interested in finding many near optimal queries rather than a single optimal one.

Given the characteristics of this optimization problem, Evolutionary Algorithms (EAs) [12, 10] appear as promising techniques for learning to automatically formulate high-quality topical queries. EAs are general-purpose search procedures based on the mechanisms of natural selection. An important component in EAs is the fitness function, which in combination with the selection mechanism determines which elements of the population are selected to be members of the next generation. Therefore, it is necessary to establish some criteria to determine if one solution is better than another. In the multi-objective case, there is not only one criterion to conclude whether one solution is better than another. The strategy adopted in this work applies the concept of Pareto optimality [21] as well as an aggregative techniques based on the harmonic mean of the given objectives to rank the queries in a manner such that the most promising ones have a higher probability of being selected.

The proposed framework uses a topic ontology to train and test the EAs. The documents classified in the ontology are used to create two large, non-overlapping indices: a training index and a testing index. The framework uses the training index together with information about the content and structure of the ontology to evolve a population of topical queries. In brief, the key features of this work are:

– Novel methods for evolving topical queries based on training data derived from topic ontologies.
– Novel criteria for assessing query quality, based on the content and structure of topic ontologies.
– A study of the effectiveness of strategies based on single- and multi-objective EAs for the problem of topical query optimization.

2 Background

2.1 A Brief Overview of Single- and Multi-Objective Evolutionary Algorithms

EAs [12, 10] are robust optimization techniques based on the principle of natural selection and survival of the fittest, which claims "in each generation the stronger individual survives and the weaker dies". Therefore, each new generation would contain stronger (fitter) individuals in contrast to its ancestors.

To use EAs in optimization problems we need to define candidate solutions by chromosomes consisting of genes and a fitness function to be maximized. A population of candidate solutions (usually of a constant size) is maintained. The goal is to obtain better solutions after some generations. To produce a new generation EAs typically use selection together with the genetic operators of crossover and mutation. Parents are selected to produce offspring, favoring those parents with highest values of the fitness function.

Crossover of population members takes place by exchanging subparts of the parent chromosomes (roughly mimicking a mating process), while mutation is the result of a random perturbation of the chromosome (e.g., replacing a gene by another). Although selection, crossover and mutation can be implemented in many different ways, their fundamental purpose is to explore the search space of candidate solutions, improving the population at each generation by adding better offspring and removing inferior ones.

In Multi-Objective Optimization Problems (MOOPs) the quality of a solution is defined by its performance in relation to several, possibly conflicting, objectives. Traditional methods are very limited because, in general, they become too expensive as the size of the problem grows [16]. EAs are a suitable technique for dealing with MOOPs [6, 8, 10] and are called in this case Multi-Objective Evolutionary Algorithms (MOEAs). There are many approaches to multi-objective optimization using MOEAs, and in general, they can be classified in Pareto or non-Pareto EAs. In the first case, the evaluation is made following the Pareto dominance concept [21]. Dominance is a partial order that could be established among vectors defined over an n-dimensional space. Figure 1 shows an example of using this concept. By means of a Fitness Function we could define a relation between vectors $\mathbf{x}_i$ in a *search space* and vectors $\mathbf{u}_i$ in an *objective space*. A non-dominated set of a feasible region in the objective space defines a *Pareto Front* over that region and the set of its associated vectors in the search space is called *Pareto Optimal Set*. The Pareto-based algorithms use the concept of domination for the selection mechanism to move a population toward the Pareto Front. Whereas in the non-Pareto EAs, the objectives are combined to obtain a single evaluation value to be used for the selection mechanism.

Besides the Pareto or non-Pareto strategy, the EAs can be classified in elitist and non

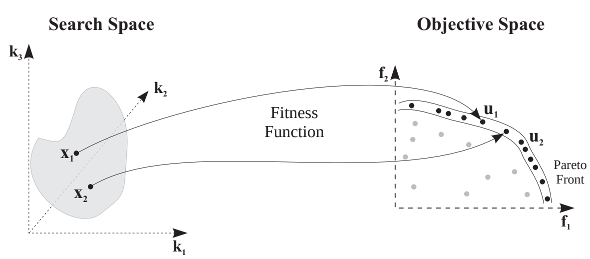


Fig. 1. Illustrative example of Pareto Dominance concept and a Fitness Function application.

elitist EAs. The difference resides in that the first uses a mechanism to retain the non-dominated individuals. In the last years, a great number of elitist Pareto-based EAs were developed. Several of them have shown very good performance in problems with objective space of size less or equal than four [8].

The Non-dominated Sorting Genetic Algorithm – II (NSGA-II) is one of the most studied and efficient EAs [9], consequently it was used in this work. The algorithm begins by creating a random parent population P_0 of size n. The population is sorted based on the non-domination concept. Each solution is assigned a fitness (or rank) equal to its non-dominated level (1 if it belongs to the first front, 2 for the second front, and so on). In this order, minimization of fitness is assumed. After ranking the solutions, a population of n offsprings, Q_0, is created using binary tournament selection, recombination and mutation. The elitism is reached by comparing the current population with previously found best non-dominated solutions. The ith generation follows the next steps:

1. A combined population $R_i = P_i \cup Q_i$ of size $2n$ is formed.
2. R_i is ordered according to non-domination. Since all previously and current population members are included in R_i, elitism is ensured. Solutions belonging to the best front, $\mathcal{F}_1$, are the best solution in the combined population R_i.
3. If the size of $\mathcal{F}_1$ is smaller than n, all members of the set $\mathcal{F}_1$ are chosen for the new population P_{i+1}. The remaining members of the population P_{i+1} are chosen from subsequent non-dominated fronts in the order of their ranking until no more sets can be accommodated. If $\mathcal{F}_j$ is the last front from which individuals can be accommodated in the population, but not all the members can enter in the population, then a decision needs to be made to choose a subset of individuals from $\mathcal{F}_j$. In order to decide which members of this front will win a place in the new population, the NSGA-II uses a selection criterion based on a crowded-comparison operator that favors solutions located in less crowded regions.

In addition to the NSGA-II, an elitist non linear aggregation alternative was used. We refer to this scheme as Aggregative MOEA. The PISA platform [2] was used to implement the strategies analyzed in this work.

2.2 Topic Ontologies and Semantic Similarity

Topic ontologies are means of classifying Web pages based on their content. In these ontologies, topics are typically organized in a hierarchical scheme in such a way that more specific topics are part of more general ones. In addition, it is possible to include cross-references to link different topic in a non-hierarchical scheme. The ODP [1] is the largest human-edited directory of the Web. It classifies millions of pages into a topical ontology combining a hierarchical and non-hierarchical scheme. This topical directory can be used to measure semantic relationships among massive numbers of pairs of Web pages or topics.

Semantic similarity between Web sites is a term used to describe the degree of relatedness between the meanings of the Web sites, as it is perceived by human subjects. Measures of semantic similarity based on taxonomies (trees) are well studied and the most successful approaches estimate semantic similarity in a taxonomy based on the notion of information content (e.g., [15]).

An important distinction between taxonomies and ontologies such as the ODP graph is that edges in a taxonomy are all of the same type ("is-a" links), while in the ODP graph edges can have diverse types (e.g., "is-a", "symbolic", "related"). In this work, we take advantage of an information-theoretic measure proposed by Maguitman et al. [17] to infer similarity from the structure of general ontologies, such as the ODP graph. Intuitively, the semantic similarity between two objects is related to their commonality and to their differences. Given a set of objects in an ontology, the commonality of two objects can be estimated by the extent to which they share information, indicated by

[1] http://dmoz.org.

the most specific class that subsumes both. Once this common classification is identified, the meaning shared by two objects can be measured by the amount of information needed to state the commonality of the two objects. The semantic similarity between two topics is then measured as the ratio between their shared meaning and their individual meanings. We omit a precise description of this semantic similarity measure due to space constraints. Refer to [17] for a detailed introduction of the semantic similarity measure used in this work.

The classification of Web pages into topics as well as the measures of semantic similarity computed between topics can be usefully exploited to formulate topical queries and assess their performance. In particular, these topical ontologies serve as a means to identify relevant (and partially relevant) documents for each topic. Once these relevance assessments are available, appropriate fitness functions that reflect different aspects of query effectiveness can be implemented for the EA strategies.

3 Evolutionary Algorithm Strategies for Evolving Topical Queries

In order to run an EA for evolving topical queries we need to generate an initial population of queries. Each chromosome represents a query and each term corresponds to a gene that can be manipulated by the genetic operators. The vector-space model is used in this approach [1] and therefore each query is represented as a vector in term space. The initial queries are formed with a fixed number of terms extracted from the topic description available from the ODP. The training index is used to implement a search interface and each query is rated according to the quality of the search results when presented to this search interface. Following the classical steps of EAs, the best queries have higher chances of being selected for subsequent generations and therefore as generations pass, queries associated with improved search results will predominate. Furthermore, the mating process continually combines these queries in new ways, generating ever more sophisticated solutions. Although all terms used to form the initial population of queries are part of the topic description, novel terms extracted from relevant documents can be included in the queries after mutation takes place. Mutation consists in replacing a randomly selected query term by another term obtained from a *mutation pool*. This pool initially contains terms extracted from the topic description and is incrementally updated with new terms from the relevant documents recovered by the system.

A new generation in our EAs is the result probabilistically selecting the most effective queries from the current set of queries. In the case when the query effectiveness can be codified as a scalar value (single-objective or aggregative methods) then two queries are chosen at random from the population and the one with highest effectiveness is selected for recombination and to populate the next generations. This method is known as 2-way tournament selection [10]. In addition, elitism is applied to prevent losing the best queries. For NSGA-II, selection is based on the elitist Pareto strategy described in section 2.1.

The recombination of a pair of parent queries into a pair of offspring queries is carried out by copying selected terms from each parent into the descendants. The crossover operator used in our proposal is known as single-point. It results in new queries in which the first n terms are contributed by one parent and the remaining terms by the second parent, where the crossover point n is chosen at random.

In order to determine if a query is effective, we first need to identify the set of relevant documents for a given topic t. Let R_t be the set containing all the documents associated with the topic t, including those in its subtopics. In addition, other topics in the ontology could be semantically similar to t and hence the documents associated with these topics are partially relevant to t. We use $\sigma^S(t, \mathit{topic}(d))$ to refer to the semantic similarity between topic t and the topic assigned to document d. Additionally, we use A_q to refer to the set of documents returned by our search engine using query q as a query, while A_{q10} is the set of top-10 ranked documents returned for query q. We use the following performance measures to determine query effectiveness:

Semantic Precision at rank 10. The well-known measure of precision [1] is the fraction of retrieved documents which are known to be relevant. The selection of a weighting model for assigning scores to the retrieved documents typically influences document ranking and query performance. For this purpose we have used a vector representation of the query together with the TFIDF weighting function [1]. This retrieval configuration will typically result in a large number of matches, sorted by their similarity to the query vector. Therefore, rather than looking at precision, we take precision at rank 10, which is computed as the fraction of the top 10 retrieved documents which are known to be relevant. In addition, because other topics in the ontology could be semantically similar (and therefore partially relevant) to the topic at hand we propose to use a measure of semantic precision at rank 10. For a query q and a topic t, this measure is defined as follows:

$$\mathit{Precision}_S@\mathit{10}(q,t) = \sum_{d \in A_{q10}} \sigma^S(t, \mathit{topic}(d))/|A_{q10}|.$$

Recall. We adopt the traditional performance measure of recall [1] as a second criteria for evaluating query effectiveness. For a query q and a topic t, recall is defined as the fraction of relevant documents R_t that are in the answer set A_q:

$$\mathit{Recall}(q,t) = \frac{|A_q \cap R_t|}{|R_t|}.$$

Harmonic Mean. Finally we use a function F_S* that aggregates *Precision*$_S$*@10* and *Recall* as follows:

$$F_S*(q,t) = \frac{2 \cdot \mathit{Precision}_S@\mathit{10}(q,t) \cdot \mathit{Recall}(q,t)}{\mathit{Precision}_S@\mathit{10}(q,t) + \mathit{Recall}(q,t)}.$$

The F_S* is an adaptation of the F_1 measure, which is the weighted harmonic mean of precision and recall [24].

In this work we study the following EA strategies for evolving topical queries:

- Single-objective EAs implementing *Precision*$_S$*@10* as its fitness function.
- Single-objective EAs implementing *Recall* as its fitness function.
- NSGA-II with *Precision*$_S$*@10* and *Recall* as its fitness function.
- Aggregative MOEA based on F_S*.

4 Evaluation

Our evaluations were run on 448 topic from the third level of the ODP hierarchy. For each topic we collected all of its URLs as well as those in its subtopics. The language of the topics used for the evaluation was restricted to English and only topics with at least 100 URLs were considered. The total number of collected pages was more than 350K. We divided each topic in such a way that 2/3 of its pages were used to create a training index and 1/3 to create a testing index. The Terrier framework [20] was used to index these pages and to create a search engine. We used the stopword list provided by Terrier and Porter stemming was performed on all terms. In addition we took advantage of the ODP ontology structure to associate a semantic similarity measure to each pair of topics. For each analyzed topic a population of 250 queries was randomly initialized using the topic ODP description. The size of each query was a random number between 1 and 32. The crossover probability was set to 0.7 and the mutation probability was 0.03. The EAs were run for 300 generations.

4.1 Monitoring the Evolution of the EAs on the Training Set

In our first experimental setting, we run a single-objective EA with the purpose of maximizing *Precision*$_S$*@10*. The charts shown in figure 2 represent the evolution of *Precision*$_S$*@10* and *Recall* for the ODP topic *Business/Industrial_Goods_-and_Services/-Machinery_and_Tools* (MACHINERY_AND_TOOLS). In these figures we have plotted the averaged *Precision*$_S$*@10* and *Recall* for the whole population (250 queries). As can be observed, near-optimal queries were obtained after a small number of generations. However, this was at the cost of very low *Recall* values.

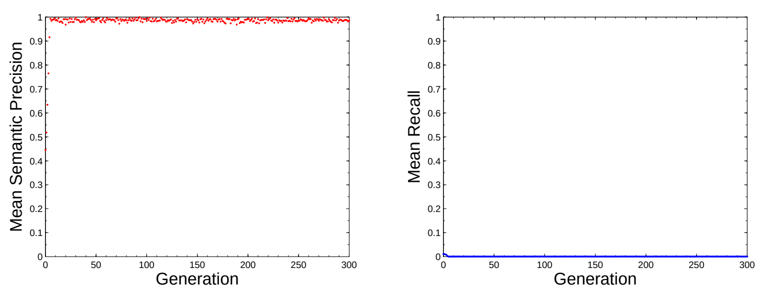


Fig. 2. The evolution of *Precision*$_S$*@10* (left) and *Recall* (right) for the topic MACHINERY_AND_TOOLS when the objective to be maximized is *Precision*$_S$*@10*.

Our second single-objective EA strategy attempted to maximize *Recall*. As shown in figure 3, *Precision*$_S$*@10* behaves erratically for this case. Although these results are shown for a single topic, analysis of the rest of the topics yielded similar behavior.

While increasing the level of one performance measure at the cost of reducing the other is sometimes acceptable, we are typically interested in improving both measures. Therefore, we run the NSGA-II algorithm to evolve topical queries that simultaneously

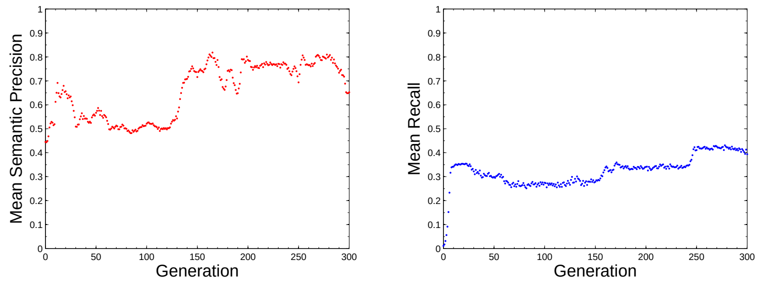


Fig. 3. The evolution of *Precision*$_S$*@10* (left) and *Recall* (right) for the topic MACHINERY_AND_TOOLS when the objective to be maximized is *Recall*.

attempted to achieve high levels of *Precision*$_S$*@10* and *Recall*. The tables in figure 4 present the means and confidence intervals over 50 topics for the first and last generations based on F_S*, *Precision*$_S$*@10* and *Recall*. In addition they show the improvement achieved by the NSGA-II algorithm when the performance of the first-generation queries is compared to the evolved ones. These comparison tables show that the NSGA-II algorithm achieved a significant query quality improvement throughout the successive generations. In other words, the algorithm was able to evolve queries with quality considerably superior to that of the queries generated directly from the topic description.

Average F_S*

	N	mean	95% C.I.	improvement
First Generation	50	0.071	[0.065,0.076]	
Last Generation	50	0.704	[0.665,0.743]	892%

Average *Precision*$_S$*@10*

	N	mean	95% C.I.	improvement
First Generation	50	0.332	[0.321,0.342]	
Last Generation	50	0.942	[0.915,0.969]	184%

Average *Recall*

	N	mean	95% C.I.	improvement
First Generation	50	0.044	[0.040,0.048]	
Last Generation	50	0.578	[0.533,0.624]	1208%

Fig. 4. First generation vs. last generation of queries evolved with NSGA-II: mean, confidence intervals and improvement for average query quality based on 50 topics.

Finally, we monitored the evolution of the aggregative MOEA throughout the 300 generations. The tables in figure 5 summarize the statistics for the performance achieved based on 50 topics. Once again, we observe that the analyzed strategy achieved a significant query quality improvement throughout the successive generations. In addition we observe that the performance of the aggregative MOEA is similar to that of NSGA-II. This allows us to conclude that for the objectives analyzed here the results of applying

Average F_S*				
	N	mean	95% C.I.	improvement
First Generation	50	0.072	[0.066,0.077]	
Last Generation	50	0.713	[0.676,0.751]	896%
Average *Precision*$_S$*@10*				
	N	mean	95% C.I.	improvement
First Generation	50	0.328	[0.318,0.339]	
Last Generation	50	0.947	[0.924,0.970]	188%
Average *Recall*				
	N	mean	95% C.I.	improvement
First Generation	50	0.045	[0.041,0.049]	
Last Generation	50	0.587	[0.544,0.630]	1214%

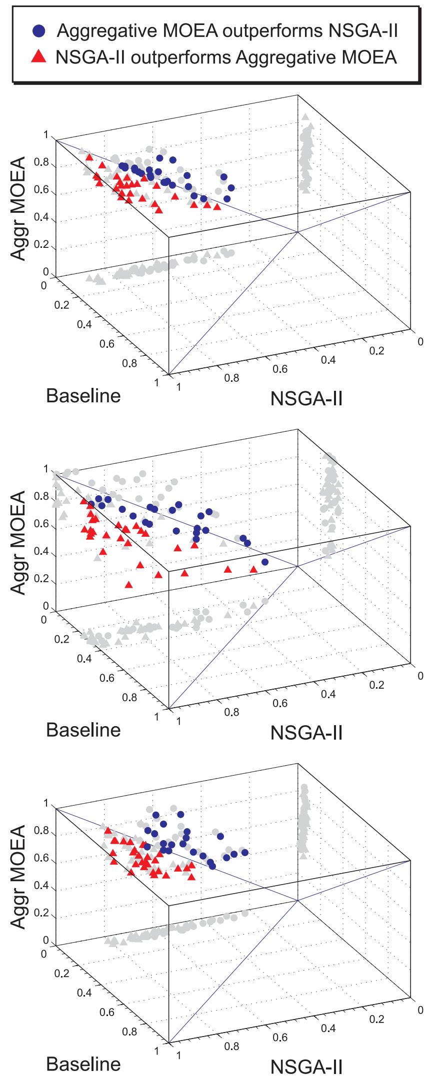
Fig. 5. First generation vs. last generation of queries evolved with the aggregative MOEA: mean, confidence intervals and improvement for average query quality based on 50 topics.

an aggregative approach to rank and evolve queries are comparable to those obtained by a non-aggregative, more computationally expensive approach.

4.2 Query Performance Evaluation on the Test Set

In order to determine if the evolved queries are effective when used on a new corpus we computed F_S*, *Precision*$_S$*@10* and *Recall* for each of the 50 topics on the test set. The question addressed here is whether the evolved queries are superior to the baseline queries (i.e., queries generated directly from the initial topic description).

The tables in figure 6 summarize the statistics for the 50 topics considered in our evaluation showing that the effectiveness of the queries evolved by NSGA-II and the aggregative MOEA result in statistically significant improvements over the baseline. The charts presented on the right-hand side of the figure depict the query performance for the individual topics. Each of the 50 topics corresponds to a trial and is represented by a point. The point's vertical coordinate (z) corresponds to the performance of the aggregative MOEA, while the point's other two coordinates (x and y) correspond to NSGA-II and the baseline. In addition we can observe the projection of each point on the x-y, x-z and y-z planes. For the y-z plane, we can observe that all the points appear above the diagonal, which means that for all the tested topics the aggregative MOEA method is superior to the baseline. Similarly, for the x-y plane, we observe that NSGA-II outperforms the baseline for all the tested cases. The x-z plane compares the performance of NSGA-II against the aggregative MOEA. Note that different markers are used to illustrate the cases in which each of the EA strategies performs better than the other: the cases in which the aggregative method outperforms NSGA-II are represented by circles while the cases in which NSGA-II outperforms the aggregative MOEA are represented by triangles. This comparison allows us to conclude that NSGA-II and the aggregative MOEA have similar performance on the test set and both strategies are able to evolve queries with quality considerably superior to that of the queries generated directly from the thematic context.



Average F_S*			
	mean	95% C.I.	improvement
Baseline	0.074	[0.069,0.080]	
NSGA-II	0.622	[0.584,0.660]	736%
Aggr. MOEA	0.644	[0.601,0.687]	762%

Average $Precision_S$*@10*			
	mean	95% C.I.	improvement
Baseline	0.293	[0.283,0.303]	
NSGA-II	0.714	[0.658,0.770]	144%
Aggr. MOEA	0.759	[0.699,0.819]	160%

Average *Recall*			
	mean	95% C.I.	improvement
Baseline	0.051	[0.046,0.055]	
NSGA-II	0.580	[0.542,0.617]	1044%
Aggr. MOEA	0.584	[0.544,0.623]	1045%

Fig. 6. A comparison of the baseline, NSGA-II and the aggregative MOEA on the test set

5 Conclusions

This paper addresses the problem of evolving queries based on a thematic context. It analyzes single- and multi-objective EA strategies that take advantage of a topic ontology and measures of semantic similarity derived from this ontology to automatically optimize query performance. We noted that the single-objective EAs present limitations that can be overcome by applying Pareto-based and aggregative techniques. We have tested the best strategies on 50 different topics selected from ODP and observed that the queries evolved from the training set can be effectively used on a new corpus. This opens the possibility for developing topical retrieval systems that can be trained using labeled documents and then used to retrieve topic-relevant material from the Web.

The techniques presented in this article are applicable to any domain for which it is possible to generate term-based characterizations of a topic. In [4] we proposed to apply single-objective genetic algorithms to evolve conjunctive queries. In that case we used the Web as a corpus for training the algorithm and the optimization criteria were based on the similarity of the retrieved material to the topic of interest. Other attempts to apply EAs in information retrieval include the design of techniques to evolve better document descriptions to aid indexing or clustering [11, 23], term-weight reinforcement in query optimization [26, 22], and optimization of keywords and logical operators [19]. A related research area deals with the development of evolving agents that crawl the Web to search for topical material [18]. A comprehensive literature review of Web-based evolutionary algorithms can be found in [14]. Differently from most of the existing EA proposals to document retrieval, which attempt to tune the weights of the individual terms, our methods take each query as an individual. The proposed method is fully automatic as long as a training corpus is available and the objective functions have been defined. A powerful aspect of this method is the use of a mutation pool containing new candidate terms collected throughout the successive generations of queries.

As part of our future work we expect to apply genetic programming to evolve queries with more complex syntaxes, including boolean operators and other special commands. In addition, we plan to run additional experiments with other parameter settings and to apply other objective functions coming from the information retrieval and Web search communities as well as ad-hoc ones.

Acknowledgement

This research work is supported by CIC (TICs-Sinergia 2008), Universidad Nacional del Sur (PGI 24/ZN13) and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

References

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, 1999.
2. Stefan Bleuler, Marco Laumanns, Lothar Thiele, and Eckart Zitzler. PISA A Platform and Programming Language Independent Interface for Search Algorithms. In *Evolutionary Multi-Criterion Optimization*. Springer-Verlag, 2003.
3. Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. Information access in context. *Knowledge based systems*, 14(1–2):37–53, 2001.
4. Rocío L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman, and Nélida Beatríz Brignole. Using genetic algorithms to evolve a population of topical queries. *Information Processing and Management*, 44(6):1863–1878, 2008.
5. Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640, 1999.

6. Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, 2007.
7. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
8. Kalyanmoy Deb. *Multi–Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2004.
9. Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation*, 6(2):182–197, 2002.
10. Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.
11. Michael Gordon. Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988.
12. John H. Holland. *Adaptation in natural and artificial systems*. Ann Arbor: The University of Michigan Press, 1975.
13. Henry Kautz, Bart Selman, and Mehul Shah. The hidden Web. *AI Magazine*, 18(2):27–36, 1997.
14. Ibrahim Kushchu. Web-based evolutionary and adaptive information retrieval. *IEEE Transactions on Evolutionary Computation*, 9(2):117–125, 2005.
15. Dekang Lin. An information-theoretic definition of similarity. In *International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann Publishers Inc., 1998.
16. Fu Lin and Guiming He. An improved genetic algorithm for multi-objective optimization. In *PDCAT*, pages 938–940. IEEE Computer Society, 2005.
17. Ana G. Maguitman, Filippo Menczer, Heather Roinestad, and Alessandro Vespignani. Algorithmic detection of semantic similarity. In *WWW '05*, pages 107–116, ACM, 2005.
18. Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology*, 4(4):378–419, 2004.
19. Zacharis Z. Nick and Panayiotopoulos Themis. Web search using a genetic algorithm. *IEEE Internet Computing*, 5(2):18–26, 2001.
20. Iadh Ounis, Christina Lioma, Craig Macdonald, and Vassilis Plachouras. Research directions in Terrier: a search engine for advanced retrieval on the web. *Novatica/UPGRADE Special Issue on Web Information Access*, VIII(1):49–56, 2007.
21. Vilfredo Pareto. *Cours d'Economie Politique*. Droz, Genève, 1896.
22. Frederick E. Petry, Bill P. Buckles, Dev Prabhu, and Donald H. Kraft. Fuzzy information retrieval using genetic algorithms and relevance feedback. In *ASIS Annual Meeting*, volume 30, pages 122–125, 1993.
23. Vijay Raghavan and Brijesh Agarwal. Optimal determination of user-oriented clusters: an application for the reproductive plan. In *International Conference on Genetic algorithms and their application*, pages 241–246, Lawrence Erlbaum Associates, Inc., 1987.
24. Cornelis J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 1979.
25. Gabriel Somlo and Adele E. Howe. Querytracker: An agent for tracking persistent information needs. In *AAMAS '04*, pages 488–495, IEEE Computer Society, 2004.
26. Jing-Jye Yang and Robert Korfhage. Query optimization in information retrieval using genetic algorithms. In *International Conference on Genetic Algorithms*, pages 603–613, Morgan Kaufmann Publishers Inc., 1993.