Inteligencia Artificial 44(2009), 14-26
doi: 10.4114/ia.v13i44.1042

INTELIGENCIA ARTIFICIAL

http://erevista.aepia.org/

Evolving Disjunctive and Conjunctive Topical Queries based on Multi-objective Optimization Criteria

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

Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur, Av. Alem 1253, Bahía Blanca, 8000 (Argentina)
phone: 54-291-4595135 fax: 54-291-4595136
† LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica
‡ LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
{rlc,cml,agm}@cs.uns.edu.ar

Abstract In this work we propose techniques based on single- and multi-objective evolutionary algorithms to automatically evolve a population of topical queries. The developed techniques can be applied in the implementation of a topical search system. We report on the results of different strategies that attempt to evolve conjunctive and disjunctive queries. Our analysis reveals the limitations of the single-objective approach and highlights the advantages of applying multi-objective evolutionary algorithms for the problem at hand. In addition, we observe that disjunctive queries have the potential to achieve better retrieval performance than conjunctive queries. Finally, we show that the multi-objective evolutionary approach results in better performance than a baseline and other state-of-the-art techniques for query refinement.

Keywords: topical search, conjunctive queries, disjunctive queries, multi-objective evolutionary algorithms.

1 Introduction

A major challenge for human information seekers is how to formulate queries that effectively reflect their information needs. Automatic topical search refers to automatically formulating queries with terms extracted from a thematic context. The resources collected by the formulation of topical queries can be used in different scenarios, such as responding to contextualized information needs [24, 7], fulfilling long term information needs [36], collecting resources for topical Web portals [10], or accessing the Deep Web [20], among others.

As an example of application scenario consider a journalist writing an article about the H1N1 pandemic. The journalist has collected a small set of articles related to the topic at hand and would like to retrieve additional material from other sources. The journalist can be assisted by an intelligent system that has the purpose of collecting relevant material from the Web. The system will monitor the journalist's task, generate an initial set of queries and incrementally refine these queries to better reflect the topic of interest. The initial queries will be generated directly from the journalist's context (e.g., the document that is being edited). These initial queries will be incrementally refined based on the small collection of readily available material, which contains documents related to the H1N1 pandemic. In

ISSN: 1988-3064(on-line)

subsequent steps, the refined topical queries are used by the system to retrieve relevant material from a larger corpus containing novel material, such as the Web.

The effectiveness of a topical query depends on the task at hand. If the criteria for evaluating query performance can be quantitatively specified then 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. A particularity of this optimization problem is that the query space is a high-dimensional space, where each possible term accounts for a new dimension. Usually, high-dimensional space problems are computationally very intensive and cannot be effectively solved using analytical methods. On the other hand, a query can be considered effective even if it is not an optimal one, at the same time as multiple queries can provide satisfactory results. Therefore, we may be interested in finding many near optimal queries rather than a single optimal one. Another aspect of this optimization problem is that several objective such as high precision and high recall can be used as criteria for evaluating query performance.

On the basis of the above discussion, Evolutionary Algorithms (EAs) [18, 15] are applicable to the problem of 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 [29] as well as an aggregative technique 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 starts by generating an initial population of queries using terms extracted from a topic description and incrementally evolves those queries based on their ability to retrieve results satisfying a number of objectives. This approach allowed us to address some interesting research question:

- **Potential for evolution of queries:** Is it possible to evolve queries in such a way that they significantly outperform those generated directly from the initial topic description?
- **Generalization of evolved queries:** Are the queries evolved from the training data useful on a new corpus?
- **Conjunctive vs. disjunctive queries:** For the objectives analyzed here, how good is the performance of conjunctive queries when compared to disjunctive queries?

2 Query Refinement and Context-Based Retrieval

In text-based Web search, users' information needs and candidate text resources are typically characterized by terms. Query refinement is usually achieved by replacing or extending the terms of a query, or by adjusting the weights of a query vector. Relevance feedback is a query refinement mechanism used to tune queries based on the relevance assessments of the query's results. A driving hypothesis for relevance feedback methods is that it may be difficult to formulate a good query when the collection of documents is not known in advance, but it is easy to judge particular documents, and so it makes sense to engage in an iterative query refinement process. A typical relevance feedback scenario will involve the following steps:

Step 1: A query is formulated.

Step 2: The system returns an initial set of results.

Step 3: A relevance assessment on the returned results is issued (relevance feedback).

Step 4: The system computes a better representation of the information needs based on this feedback.

Step 5: The system returns a revised set of results.

Depending on the level of automation of step 3 we can distinguish three forms of feedback:

- **Supervised Feedback**: Requires explicit feedback, which is typically obtained from users who indicate the relevance of each of the retrieved documents (e.g., [34]).
- **Unsupervised Feedback**: It applies blind relevance feedback, and typically assumes that the top k documents returned by a search process are relevant (e.g., [6]). This also known as pseudo-relevance feedback.
- **Semi-supervised Feedback**: The relevance of a document is inferred by the system. A common approach is to monitor the user behavior (e.g., documents selected for viewing or time spent viewing a document). Provided that the information seeking process is performed within a thematic context, another automatic way to infer the relevance of a document is by computing the similarity of the document to the user's current context (e.g., [19]).

The best-known algorithm for relevance feedback has been proposed by Rocchio [34]. Given an initial query vector $\overrightarrow{q}$ a modified query $\overrightarrow{q_m}$ is computed as follows:

$$\overrightarrow{q_m} = \alpha \overrightarrow{q} + \beta \sum_{\overrightarrow{d_j} \in D_r} \overrightarrow{d_j} - \gamma \sum_{\overrightarrow{d_j} \in D_n} \overrightarrow{d_j}.$$

where D_r and D_n are the sets of relevant and non-relevant documents respectively and α, β and γ are tuning parameters. A common strategy is to set α and β to a value greater than 0 and γ to 0, which yields a positive feedback strategy. When user relevance judgments are unavailable, the set D_r is initialized with the top k retrieved documents and D_n is set to $\emptyset$. This yields an unsupervised relevance feedback method.

Several successors of Rocchio's method have been proposed with varying success. One of them is selective query expansion [2], which monitors the evolution of the retrieved material and is disabled if query expansion appears to have a negative impact on the retrieval performance. Other successors of Rocchio's method use an external collection different from the target collection to identify good terms for query expansion. The refined query is then used to retrieve the final set of documents from the target collection [23]. A successful generalization of Rocchio's method is the Divergence from Randomness mechanism with Bose-Einstein statistics (Bo1) [1]. To apply this model, we first need to assign weights to terms based on their informativeness. This is estimated by the divergence between the term distribution in the top-ranked documents and a random distribution as follows:

$$w(t) = tf_x . log_2 \frac{1 + P_n}{P_n} + log_2(1 + P_n)$$

where tf_x is the frequency of the query term in the top-ranked documents and P_n is the proportion of documents in the collection that contain t. Finally, the query is expanded by merging the most informative terms with the original query terms.

During recent years several techniques that formulate queries from the user context have been proposed [7, 21]. Other methods support the query expansion and refinement process through a query or browsing interface requiring explicit user intervention [35, 4]. Limited work, however, has been done on methods that simultaneously take advantage of the user context and results returned from a corpus to refine queries.

3 An Overview of Evolutionary Algorithms

EAs [18, 15] 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 computationally intensive as the size of the problem grows [31, 25]. EAs are a suitable technique for dealing with MOOPs [12, 13, 15] 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 [29] discussed below. In the second case, the objectives are combined to obtain a single evaluation value.

There are some basic definitions based on the Pareto concept that must be considered:

Definition 1 Pareto Dominance *[11]: A vector* $\mathbf{u} = (u_1, u_2, \ldots, u_k)$ *is said to* **dominate** *another vector* $\mathbf{v} = (v_1, v_2, \ldots, v_k)$ *(denoted by* $\mathbf{u} \preceq \mathbf{v}$*) if and only if* $\mathbf{u}$ *is partially less than* $\mathbf{v}$*, i.e.,* $\forall i \in \{1, \ldots, k\}$, $u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.

Definition 2 Pareto Optimality *[11]: A solution* $\mathbf{x} \in \Omega$ *is said to be Pareto Optimal with respect to a* Ω*, if and only if* $\nexists\ \mathbf{x}^* \in \Omega$ *for which* $\mathbf{v} = F(\mathbf{x}^*) = (f_1(\mathbf{x}^*), \ldots, f_k(\mathbf{x}^*))$ *dominates* $\mathbf{u} = F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$. *Where* Ω *is a feasible region for the MOOP.*

Definition 3 Pareto Optimal Set *[11]: For a given MOOP, F(*$\mathbf{x}$*), and a feasible region for that MOOP,* Ω*, the* ***Pareto Optimal Set****,* $\mathcal{P}^*$*, is defined as:*

$$\mathcal{P}^* := \{\mathbf{x} \in \Omega,\ \nexists\ \mathbf{x}^* \in \Omega,\ F(\mathbf{x}^*) \preceq F(\mathbf{x})\}$$

Definition 4 Pareto Front *[11]: For a given MOOP, F(*$\mathbf{x}$*), in a feasible region for that MOOP,* Ω*, and a* ***Pareto Optimal Set****,* $\mathcal{P}^*$*, the* ***Pareto Front*** $\mathcal{PF}^*$ *is defined as:*

$$\mathcal{PF}^* := \{\mathbf{u} = F(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}^*\}$$

Besides the Pareto or non-Pareto strategy, the EAs can be classified in elitist and non 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 [13].

The Non-dominated Sorting Genetic Algorithm – II (NSGA-II) is one of the most studied and efficient EAs [14], consequently it was used in this work. The algorithm begins 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 lesser crowded regions.

In addition to the NSGA-II, an elitist non linear aggregation alternative was used. This scheme was implemented using an adaptation of the well known F_1 measure. This measure is reviewed in section 4.2. The PISA platform [5] was used to implement the strategies analyzed in this work.

4 Evolving Topical Queries with Multi-objective Evolutionary Algorithms

In order to evolve topical queries we start with a population of queries composed of terms extracted from an initial description of the given topic and rate the effectiveness of each query according to the quality of the search results. 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.

4.1 Population and Representation of Chromosomes

Each chromosome corresponds to a query, which is represented as a list of terms. For our analysis of conjunctive queries, terms are assumed to be connected by the AND operator while for disjunctive queries terms are connected by the OR operator. Each term corresponds to a gene that can be manipulated by the genetic operators. The population is initialized with a fixed number of queries randomly generated with terms from the thematic description. Novel terms can be included in the queries after mutation takes place. These novel terms are obtained from a *mutation pool*, which is an ever increasing set of terms that may or may not be part of the initial context.

4.2 Fitness Function

The fitness function defines the criterion for assessing the quality of a query. One of the objective functions considered in our analysis is precision at rank 10 (*Precision@10*), which is the fraction of the top 10 retrieved documents which are known to be relevant. To define this fitness function we associate with the search space Q and topics T a function $\textit{Precision@10} : Q \times T \to [0, 1]$. This objective function can numerically evaluate an individual query q in terms of precision at rank 10 for a given topic t as follows:

$$\textit{Precision@10}(q,t) = \frac{|A_{q10} \cap R_t|}{|A_{q10}|},$$

where A_{q10} is the set of top-10 ranked documents returned by a search engine when $\mathbf{q}$ is used as a query, and R_t is the set containing all the documents associated with topic t, including those in its subtopics.

Another fitness function adopted in this work, $\textit{Recall} : Q \times T \to [0, 1]$, is defined as the fraction of relevant documents R_t that are in the answer set A_q:

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

Finally we use a function $F^* : Q \times T \to [0, 1]$ that aggregates *Precision@10* and *Recall* as follows:

$$F^*(q,t) = \frac{2 \cdot \textit{Precision@10}(q,t) \cdot \textit{Recall}(q,t)}{\textit{Precision@10}(q,t) + \textit{Recall}(q,t)}.$$

The F^* is an adaptation of the F_1 measure, which is the weighted harmonic mean of precision and recall [33]. Maximization of fitness is assumed to prefer a query over another for the selection process. We have used a vector representation of the query together with the TFIDF weighting function [3] for assigning scores to the retrieved documents.

4.3 Genetic Operators

A new generation in our EAs is determined by a set of operators that select, recombine and mutate queries of the current population.

- **Selection:** A new population is generated by probabilistically selecting the highest-quality 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. In addition, elitism is applied to prevent losing the best queries. For the multi-objective case, selection is based on the elitist Pareto strategy described in section 3.
- **Crossover:** Some of the selected queries are carried out into the next generations as they are, while others are recombined to create new queries. 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.
- **Mutation:** Small random changes can be produced to the new population of queries. These changes consist in replacing a randomly selected query term t^q by another term t^p . The term t^p is obtained from the *mutation pool*, which is a set of terms that initially contains terms extracted from the thematic context and is incrementally updated with new terms from the relevant documents recovered by the system.

5 Evaluation

To run our evaluations we collected 448 topics from the Open Directory Project (ODP)[1]. The topics were selected from the third level of the ODP hierarchy. A number of constraints were imposed on this selection with the purpose of ensuring the quality of our corpus. The minimum size for each selected topic was 100 URLs and the language was restricted to English. For each topic we collected all of its URLs as well as those in its subtopics. The total number of collected pages was more than 350K. The Terrier framework [28] was used to index these pages and to run our experiments. We used the stopword list provided by Terrier and Porter stemming was performed on all terms. We divided each topic in such a way that 2/3 of its pages were used for training and 1/3 for testing. The following scenarios were analyzed:

- A **Single-objective EA** with *Precision@10* as the objective function.
- A **Single-objective EA** with *Recall* as the objective function.
- **NSGA-II** with both *Precision@10* and *Recall* as objective functions.
- An **aggregative MOEA** with *F** as the objective function.

The EAs were run for 10 different 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.

5.1 Analyzing the Evolution of the Single-objective EAs

In our first experimental setting, we run a single-objective EA for 200 generations with the purpose of maximizing *Precision@10*. Both conjunctive (AND) and disjunctive (OR) queries were tested. In figure 1 we show the evolution of *Precision@10* (left) and *Recall* (right) for the ODP topic *BUSINESS/BUSINESS_SERVICES/CONSULTING* (Consulting). As can be observed in this figure, near-optimal

[1] http://dmoz.org

queries were obtained for both AND- and OR-queries after a small number of generations. However, this was at the cost of very low *Recall* values.

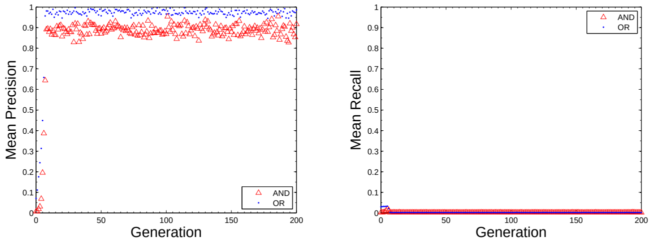

Figure 1: The evolution of *Precision@10* (left) and *Recall* (right) for the topic CONSULTING when the objective to be maximized is *Precision@10*.

Unsurprisingly, very low *Precision@10* values were achieved when the objective to be maximized was *Recall*, as shown in figure 2. Note, in addition that although high *Recall* values were achieved for OR-queries, this was not the case for AND-queries. Although these results are shown for a single topic, analysis of the rest of the topics yielded similar behavior.

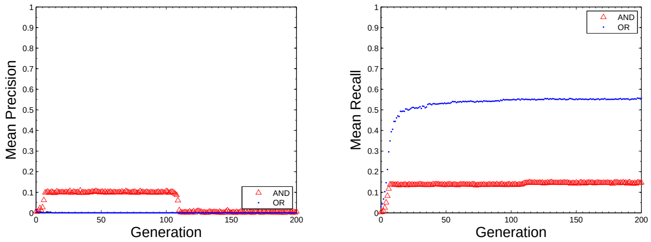

Figure 2: The evolution of *Precision@10* (left) and *Recall* (right) for the topic CONSULTING when the objective to be maximized is *Recall*.

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.

5.2 Analyzing the Evolution of the NSGA-II

In order to evolve topical queries that simultaneously attempted to achieve high levels of *Precision@10* and *Recall* we run the NSGA-II algorithm for 300 generations. In figure 3 we plotted the performance achieved at each generation for the topic CONSULTING by looking at *Precision@10* (left), *Recall* (center) and *F** (right). It is interesting to note that when OR-queries were evolved, NSGA-II allowed to achieve very high levels of *Precision@10* without compromising *Recall*. In the case of AND-queries, although high *Precision@10* is eventually achieved, the values for *Recall* remain low.

The trend observed in figure 3 for topic CONSULTING was also observed for the other topics in our corpus. Table 1 presents the means over 10 topics for the first and last generations based on *Precision@10*, *Recall* and *F**. This comparison shows that the NSGA-II algorithm achieved a substantial improvement throughout the successive generations.

5.3 Analyzing the Evolution of the Aggregative MOEA

Finally, we monitored the evolution of the aggregative MOEA throughout 300 generations. The charts in figure 4 show these values for the topic CONSULTING while table 2 summarizes the mean performance achieved for the 10 topics considered in our evaluation.

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 an aggregative approach to

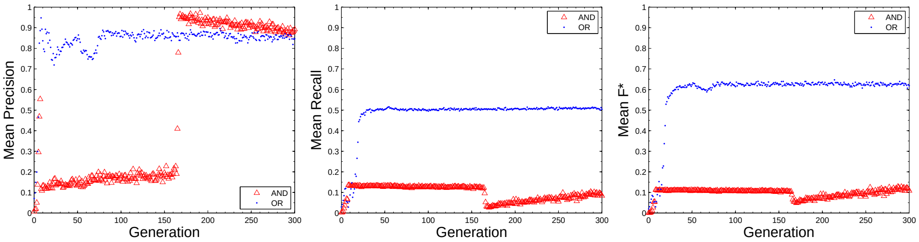

Figure 3: The evolution of *Precision@10* (left), *Recall* (center) and *F** (right) for the topic Consulting when applying the NSGA-II algorithm.

NSGA-II: AND-queries			
TRAINING	mean *Precision@10*	mean *Recall*	mean *F**
First Generation	0.038	0.059	0.020
Last Generation	0.689	0.196	0.176
NSGA-II: OR-queries			
TRAINING	mean *Precision@10*	mean *Recall*	mean *F**
First Generation	0.055	0.049	0.022
Last Generation	0.953	0.653	0.766

Table 1: First generation vs. last generation of queries evolved with NSGA-II.

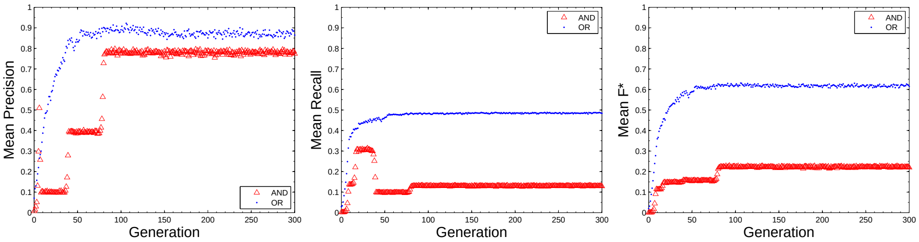

Figure 4: The evolution of *Precision@10* (left), *Recall* (center) and *F** (right) for the topic Consulting when applying the aggregative MOEA.

Aggregative MOEA: AND-queries			
TRAINING	mean *Precision@10*	mean *Recall*	mean *F**
First Generation	0.042	0.060	0.022
Last Generation	0.570	0.358	0.372
Aggregative MOEA: OR-queries			
TRAINING	mean *Precision@10*	mean *Recall*	mean *F**
First Generation	0.054	0.049	0.022
Last Generation	0.948	0.635	0.749

Table 2: First generation vs. last generation of queries evolved with the aggregative MOEA.

rank and evolve queries are comparable to those obtained by a non-aggregative, more computationally intensive approach. In addition, OR-queries have better potential for evolution than AND-queries.

5.4 Evaluating Query Performance on the Test Set

In order to determine if the evolved queries are effective when used on a new corpus we computed *Precision@10*, *Recall* and *F** for each of the 10 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). Tables 3 and 4 present this comparison for both NSGA-II and the aggregative MOEA. This comparison shows that the tested algorithms are able to evolve queries with quality considerably superior to that of the queries generated directly from the thematic context. In particular, OR-queries achieved much higher performance than AND-queries.

NSGA-II: AND-queries			
TESTING	mean *Precision@10*	mean *Recall*	mean *F**
Baseline	0.016	0.075	0.011
Evolved Queries	0.409	0.193	0.156
NSGA-II: OR-queries			
TESTING	mean *Precision@10*	mean *Recall*	mean *F**
Baseline	0.015	0.056	0.009
Evolved Queries	0.572	0.637	0.577

Table 3: Baseline vs. queries evolved with NSGA-II.

Aggregative MOEA: AND-queries			
TESTING	mean *Precision@10*	mean *Recall*	mean *F**
Baseline	0.018	0.077	0.013
Evolved Queries	0.500	0.336	0.338
Aggregative MOEA: OR-queries			
TESTING	mean *Precision@10*	mean *Recall*	mean *F**
Baseline	0.015	0.057	0.009
Evolved Queries	0.543	0.643	0.538

Table 4: Baseline vs. queries evolved with the aggregative MOEA.

Finally, we conducted an evaluation in order to compare the performance of the aggregative MOEA and NSGA-II to a state-of-the-art query refinement technique. For this purpose we decided to use the Bo1 method discussed in section 2. Table 5 presents a comparison between the multi-objective evolutionary strategies based on OR-queries and the Bo1 method. We observe that the techniques based on evolutionary algorithms are superior to Bo1.

OR-queries			
TESTING	mean *Precision@10*	mean *Recall*	mean *F**
Aggregative MOEA	0.543	0.643	0.538
NSGA-II	0.572	0.637	0.577
Bo1	0.013	0.127	0.014

Table 5: A comparison between the multi-objective evolutionary strategies based on OR-queries and the Bo1 method.

6 Conclusions and Future Work

This paper analyzes different strategies for evolving topical queries with single- and multi-objective EAs. We noted that the single-objective EAs present limitations that can be overcome by applying Pareto-based and aggregative techniques. Because the aggregative MOEA is less computationally intensive than NSGA-II, we conclude it is a better choice for the problem studied here. In addition, the aggregative techniques are more flexible, allowing to define aggregate fitness functions that favor one objective over the other.

We have also compared the potential for evolution that OR-queries have in comparison to AND-queries, and observed that the former are considerably superior to the latter. Interestingly, most popular search engines, such as Google or Yahoo, use conjunctive matching, which means that it is mandatory for all the query term to appear in a document (or to be associated with the document in some way) for it to be considered. A reason for using conjunctive matching is that user information needs are more intuitively defined using the AND semantics: as the number of terms increases the answer set is more selective and more precisely fits the information needs of the user. However, as observed here, automatically generated queries can achieve better performance for topical search when disjunctive matching is applied.

Another important result derived from our evaluations is that the evolved queries do not overfit the training data. Therefore, once a population of topical queries is available, it can be used to retrieve topical material from sources such as the Web, where no relevance assessments are typically available. We also observe that the multi-objective evolutionary strategies based on OR-queries are superior to a baseline and other state-of-the-art query refinement techniques.

Other attempts to apply EAs in information retrieval include the design of techniques to evolve better document descriptions to aid indexing or clustering [17, 32], term-weight reinforcement in query optimization [16, 37, 30], and optimization of keywords and logical operators [27]. A related research area deals with the development of evolving agents that crawl the Web to search for topical material [26]. A comprehensive literature review of Web-based evolutionary algorithms can be found in [22].

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. The use of this incrementally generated pool of terms has shown to be effective in aiding the exploration of query space [8].

The techniques presented in this article are applicable to any domain for which it is possible to generate term-based characterizations of a context. In [9] we proposed to apply single-objective genetic algorithms to evolve AND-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. In the present work, instead of using unlabeled material from the Web we take advantage of a taxonomy of topics from ODP and its associated webpages, which are labeled as relevant or irrelevant to the specific topics.

In the future we expect to run additional experiments applying other objective functions coming from the information retrieval and Web search communities as well as ad-hoc ones. Moreover, we plan to test different parameter settings for the EAs. In this work we look at queries with simple syntaxes. An interesting follow-up study concerns applying genetic programming to evolve queries with more complex syntaxes, including boolean operators and other special commands.

Acknowledgement

This research work is supported by CIC (TICs-Sinergia 2008), Agencia Nacional de Promoción Científica y Tecnológica (PICT 2005 Nro. 32373), Universidad Nacional del Sur (PGI 24/ZN13) and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET).

References

[1] Giambattista Amati. *Probabilistics Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Department of Computing Science, University of Glasgow, UK, June 2003. doi: 10.1145/582415.582416.

[2] Giambattista Amati, Claudio Carpineto, and Giovanni Romano. Query difficulty, robustness and selective application of query expansion. In *Advances in Information Retrieval, 26th European Conference on IR research*, pages 127–137, Berlin / Heidelberg, 2004. Springer. doi: 10.1007/b96895.

[3] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Reading, MA, USA, May 1999.

[4] Bodo Billerbeck, Falk Scholer, Hugh E. Williams, and Justin Zobel. Query expansion using associated queries. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, pages 2–9, New York, NY, USA, 2003. ACM Press. doi: 10.1145/956863.956866.

[5] Stefan Bleuler, Marco Laumanns, Lothar Thiele, and Eckart Zitzler. PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 494–508. Springer-Verlag, Berlin, 2003. doi: 10.1007/3-540-36970-8_35.

[6] Chris Buckley, Amit Singhal, and Mandar Mitra. New Retrieval Approaches Using SMART. In D. K. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, volume Special Publication 500-236, Gaithersburg, MD, USA, 1995. National Institute of Standards and Technology (NIST). doi: 10.1016/S0950-7051(00)00105-2.

[7] Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. Information Access in Context. *Knowledge Based Systems*, 14(1–2):37–53, 2001. doi: 10.1016/S0950-7051(00)00105-2.

[8] Rocío L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman, and Nélida B. Brignole. Genetic algorithms for topical web search: A study of different mutation rates. In *Anales del XIII Congreso Argentino de Ciencias de la Computación (CACIC)*, Corrientes, Argentina, October 2007. Universidad Nacional del Nordeste.

[9] Rocío L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman, and Nélida B. Brignole. Using genetic algorithms to evolve a population of topical queries. *Information Processing and Management*, 44(6):1863–1878, 2008. doi: 10.1016/j.ipm.2007.12.012.

[10] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, 31(11–16):1623–1640, 1999. doi: 10.1016/S1389-1286(99)00052-3.

[11] Carlos A. Coello Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11-12):1245–1287, 2002. doi: 10.1016/S0045-7825(01)00323-1.

[12] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation. Springer-Verlag New York, Inc, Secaucus, NJ, USA, 2nd edition, September 2007. doi: 10.1007/978-0-387-36797-2.

[13] Kalyanmoy Deb. *Multi–Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Ltd., Chichester, W Sussex, UK, 2001.

[14] 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. doi: 10.1109/4235.996017.

[15] Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Natural Computing Series. Springer, Heidelberg, 1st edition, 2003.

[16] Ophir Frieder and Hava Tova Siegelmann. On the allocation of documents in multiprocessor information retrieval systems. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 230–239, New York, NY, USA, 1991. ACM. doi: 10.1145/122860.122884.

[17] Michael Gordon. Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988. doi: 10.1145/63039.63044.

[18] John H. Holland. *Adaptation in natural and artificial systems*. Bradford Series in Complex Adaptive Systems. The University of Michigan Press, Ann Arbor, MI, USA, 1975.

[19] Chris Jordan and Carolyn R. Watters. Extending the Rocchio Relevance Feedback algorithm to provide Contextual Retrieval. In *Advances in Web Intelligence, Proceedings of the Second International Atlantic Web Intelligence Conference (AWIC)*, volume 3034 of *Lecture Notes in Computer Science*, pages 135–144, Berlin / Heidelberg, 2004. Springer. doi: 10.1007/b97883.

[20] Henry Kautz, Bart Selman, and Mehul Shah. The Hidden Web. *AI Magazine*, 18(2):27–36, 1997.

[21] Reiner Kraft, Chi Chao Chang, Farzin Maghoul, and Ravi Kumar. Searching with context. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web*, pages 477–486, New York, NY, USA, 2006. ACM. doi: 10.1145/1135777.1135847.

[22] Ibrahim Kushchu. Web-based evolutionary and adaptive information retrieval. *IEEE Transactions on Evolutionary Computation*, 9(2):117–125, April 2005. doi: 10.1109/TEVC.2004.842093.

[23] Kui Lam Kwok and Margaret S. Chan. Improving two-stage ad-hoc retrieval for short queries. In *SIGIR '98: Proceedings of the 21st Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 250–256, New York, NY, USA, 1998. ACM. doi: 10.1145/290941.291003.

[24] David B. Leake, Travis Bauer, Ana Maguitman, and David C. Wilson. Capture, Storage and Reuse of Lessons about Information Resources: Supporting Task-Based Information Search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems.*, pages 33–37, Austin, Texas, 2000. AAAI Press.

[25] Fu Lin and Guiming He. An Improved Genetic Algorithm for Multi-Objective Optimization. In *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT)*, pages 938–940, Los Alamitos, CA, USA, 2005. IEEE Computer Society. doi: 10.1109/PDCAT.2005.84.

[26] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology (TOIT)*, 4(4):378–419, November 2004. doi: 10.1145/1031114.1031117.

[27] Zacharis Z. Nick and Panayiotopoulos Themis. Web search using a genetic algorithm. *IEEE Internet Computing*, 5(2):18–26, 2001. doi: 10.1109/4236.914644.

[28] 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, Ricardo Baeza-Yates et al. (Eds), Invited Paper*, VIII(1):49–56, February 2007.

[29] Vilfredo Pareto. *Cours d'économie politique*. Rouge, Lausanne, Vaud, Switzerland, 1896. doi: 10.1177/000271629700900314.

[30] Frederick E. Petry, Bill P. Buckles, Dev Prabhu, and Donald H. Kraft. Fuzzy information retrieval using genetic algorithms and relevance feedback. In Susan Bonzi, editor, *Proceedings of the 56th Annual Meeting of the American Society for Information Science (ASIS)*, volume 30, pages 122–125, Medford, NJ, USA, October 1993. Learned Information.

[31] Gregorio Toscano Pulido. Optimización Multiobjetivo Usando un Micro Algoritmo Genético. Master's thesis, Maestría en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México, September 2001.

[32] Vijay Raghavan and Brijesh Agarwal. Optimal determination of user-oriented clusters: an application for the reproductive plan. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 241–246, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.

[33] Cornelis J. van Rijsbergen. *Information Retrieval.* Butterworth-Heinemann, Newton, MA, USA, 1979.

[34] Joseph John Rocchio. Relevance feedback in information retrieval. In Gerard Salton, editor, *The Smart retrieval system - experiments in automatic document processing*, chapter 14, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, USA, 1971.

[35] Falk Scholer and Hugh E. Williams. Query Association for Effective Retrieval. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 324–331, New York, NY, USA, 2002. ACM Press. doi: 10.1145/584792.584846.

[36] Gabriel Somlo and Adele E. Howe. QueryTracker: An Agent for Tracking Persistent Information Needs. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 488–495, Los Alamitos, CA, USA, 2004. IEEE Computer Society. doi: 10.1109/AAMAS.2004.220.

[37] Jing-Jye Yang and Robert Korfhage. Query Optimization in Information Retrieval using Genetic Algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 603–613, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.