This is a preprint of an article published in
Journal of the American Society for Information Science and Technology
JASIST 61(6): 1258-1274 (2010)

Multi-Objective Evolutionary Algorithms for Context-Based Search

Rocío L. Cecchini
LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, Email: rlc@cs.uns.edu.ar

Carlos M. Lorenzetti and Ana G. Maguitman
LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial, Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, Email: {cml, agm}@cs.uns.edu.ar

Nélida B. Brignole
LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica, Planta Piloto de Ingeniería Química, CCT-CONICET-Bahía Blanca, Argentina, Email: nbb@cs.uns.edu.ar

Abstract

Formulating high-quality queries is a key aspect of context-based search. However, determining the effectiveness of a query is challenging because multiple objectives, such as high precision and high recall, are usually involved. In this work we study techniques that can be applied to evolve contextualized queries when the criteria for determining query quality are based on multiple objectives. We report on the results of three different strategies for evolving queries: (1) single-objective, (2) multi-objective with Pareto-based ranking, and (3) multi-objective with aggregative ranking. After a comprehensive evaluation with a large set of topics we discuss the limitations of the single-objective approach and observe that both the Pareto-based and aggregative strategies are highly effective for evolving topical queries. In particular, our experiments lead us to conclude that the multi-objective techniques are superior to a baseline as well as to well-known and ad-hoc query reformulation techniques.

Keywords: context-based search, topical queries, multi-objective evolutionary algorithms

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).

Introduction

Context-based search is the process of seeking material based on a topic of interest (Budzik, Hammond, & Birnbaum, 2001; Maguitman, Leake, & Reichherzer, 2005; Kraft, Chang, Maghoul, & Kumar, 2006). Consider, for example, 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. This local collection of documents is indexed and tagged as relevant, while other documents are added to the index and tagged as irrelevant. The journalist can be assisted by an intelligent system that monitors the journalist's task, generates an initial set of queries and incrementally refines these queries to better reflect the topic of interest. The initial queries could be generated directly from the journalist's context (e.g., the document that is being edited) or from a short description provided explicitly by the journalist. 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 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 availability of powerful search interfaces makes it possible to develop a plethora of applications for information access in context such as the one described above. The matching and ranking mechanisms employed by existing search services are commonly fixed by the service provider. The criteria considered for retrieval change from service to service (e.g., content relevance, document popularity, document freshness, meta-information) and are typically obscure to those using the interface. The only access point to relevant material is through the submission of queries. As a consequence, learning to automatically formulate effective topical queries is an important research problem in the area of context-based search.

The effectiveness of a query depends on what the user is trying to achieve. If the goal is to attain good coverage, a broad query will be more effective than a specific one. In this case the user will try to retrieve most of the relevant material, albeit diluted with irrelevant answers. On the other hand, if the goal is to attain good precision, specific queries will be preferred over broad ones.

If the criteria for evaluating query performance can be quantitatively specified then the problem of topical search can be seen as an optimization problem. In this case, the objective function to be maximized quantifies the optimality of a query so that a particular query may be ranked against all the other queries. In this optimization problem, therefore, the search space is defined as the set of possible queries that can be presented to a search interface. However, ranking these queries is not straightforward due to the fact that the measure of query effectiveness can be defined using more than one, possibly conflicting, objectives. As a consequence we are facing a multi-objective optimization problem that may need to balance a number of criteria, such as precision, recall or other customized metrics.

Each query is a sequence of terms. Therefore, another particularity of this optimization problem is that the query space is a high-dimensional space, where each possible term accounts for a new dimension. This kind of problems are computationally very expensive and cannot be effectively solved using analytical methods. 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 (Cormen, Leiserson, & Rivest, 1990). Therefore, existing methods to solve complex problems by breaking them down into simpler steps are not effective for our purpose. On the other hand, a query can be considered effective even if it is not an optimal one, at the same time as multiple and diverse queries can provide satisfactory results. Therefore, we may be interested in finding many near optimal queries rather than a single optimal one.

On the basis of the characteristics described above, Evolutionary Algorithms (EAs) (Holland, 1975; Eiben & Smith, 2003) are applicable to the problem of learning to automatically formulate optimal (or near optimal) 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 strategies adopted in this work apply two techniques to simultaneously account for multiple objectives. The first technique is based on the concept of Pareto optimality (Pareto, 1896), which is applied to rank queries in a manner such that the non-dominated solutions (a concept that will be discussed in detail later) have a higher probability of being selected. The second technique aggregates multiple measures of effectiveness into a single objective measure.

Application Scenarios

There are multiple application scenarios in which learning to formulate topical queries can help access useful information. Some of these applications are described next.

Responding to contextualized information needs. Task-based search systems exploit user interaction with computer applications to determine the user's current task and contextualize information needs (D. B. Leake, Bauer, Maguitman, & Wilson, 2000; Budzik et al., 2001). Basic keyword searches could very easily miss task-relevant material. By learning to evolve high-quality topical queries, a task-based search system can automatically generate suggestions that are richly contextualized within the user's task.

Fulfilling long-term information needs. Many users have persistent information needs and several methods have been proposed to identify and track these needs (e.g., (Somlo & Howe, 2004)). Once the profile of a user's interests is constructed it can be taken as a starting point to automatically formulate queries. By taking this approach a system can periodically collect and make readily available material relevant to the user long-term interests.

Collecting resources for topical Web portals. Topical Web portals have the purpose of gathering resources on specific subjects. The collected material is used to build specialized search and directory sites. Typically, focused crawlers are in charge of mining the Web to harvest topical content and populate the indices of these portals (Chakrabarti, Berg, & Dom, 1999; Menczer, Pant, & Srinivasan, 2004). As an alternative to focused crawlers, this process can be supported by formulating topical queries to a search engine and selecting from the answer set those resources that are related to the topic at hand.

Accessing the Deep Web. Most of the Web's information can be found in the form of dynamically generated pages and constitutes what is known as the Deep Web (Kautz, Selman, & Shah, 1997). The pages that constitute the Deep Web do not exist until they are created dynamically as the result of a query presented to search forms available in specific sites (e.g., pubmedcentral.nih.gov, amazon.com). Therefore, learning to formulate effective queries for specific Deep Web sites can help automate the process of harvesting Deep Web topical resources. Learning to formulate queries for Deep Web sites that are not specified in advance is more challenging than learning to formulate queries for specific sites because in the former case the syntax of a proper query is not known in advance. Despite heterogeneity in automatically generated pages, (He, Meng, Lu, Yu, & Wu, 2007) proposed a model to describe the contents of Web search interfaces in order to understand them. They implemented a system to automatically construct a schema for any Web search interface and show the effectiveness of that system in real interfaces.

Support for Knowledge Management. Effective knowledge management may require going beyond initial knowledge capture, to support decisions about how to extend previously-captured knowledge (D. Leake et al., 2003; Maguitman et al., 2005). The Web provides a rich source of information on potential new material to include in a knowledge model. Thus material can be accessed by means of contextualized queries presented to a conventional search engine, where the context is given by the knowledge model under construction. Using the Web as a huge repository of collective memory, a system that learns to formulate topical queries can facilitate the process of capturing knowledge to help extend organizational memories.

Research Questions

A thematic context can be naturally characterized by a selection of terms. From a pragmatic perspective, the selection of terms should be directed towards the completion of the task at hand. Even for the same topic, a given term will have different importance depending on whether it is needed for query construction, index generation, topic summarization or similarity assessment. For example, a term that is a useful descriptor, and therefore useful in topic summarization, may lack discriminating power, rendering it ineffective as a query term, due to low precision of search results, unless it is combined with other terms that can discriminate between good and bad results. The central question addressed in this paper is how to automatically guide the process of combining terms to formulate topical queries and rank them according to the user's objectives.

This paper describes and evaluates a framework that addresses the problem of formulating topical queries when multiple objectives need to be balanced. 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. Specifically, we are studying the following research questions:

- Question 1: How to evolve queries when we are attempting to simultaneously satisfy multiple objectives such as high precision and high recall?
- Question 2: Are the evolved topical queries useful beyond the training set? In other words, are the queries evolved by our strategy effective when tested for the same topic but

on a new corpus?

• Question 3: How effective are the queries evolved by the EAs, in comparison to queries formulated directly from the thematic context, or to queries generated using state-of-the-art query refinement techniques?

Background and Related Work

Multi-Objective Evolutionary Algorithms

EAs (Holland, 1975; Eiben & Smith, 2003) 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"(Darwin, 1859). Therefore, each new generation would contain stronger (fitter) individuals than their ancestors.

To use EAs in optimization problems we need to define candidate solutions by chromosomes consisting of genes and a fitness function to be minimized/maximized. A population of candidate solutions 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). A simple EA works as follows:

Step 1: Start with a randomly generated population.

Step 2: Evaluate the fitness of each individual in the population.

Step 3: Select individuals to reproduce based on their fitness.

Step 4: Apply crossover with probability Pc.

Step 5: Apply mutation with probability Pm.

Step 6: Replace the population by the new generation of individuals.

Step 7: Go to step 2.

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. A number of introductory books and survey articles are available for a complete study of the topic (Goldberg, 1989; M. Srinivas & Patnaik, 1994; Mitchell, 1996; Eiben & Smith, 2003).

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 (Pulido, 2001; Lin & He, 2005). EAs are a suitable technique for dealing with MOOPs (Coello Coello, Lamont, & Van Veldhuizen, 2007; Deb, 2001; Eiben & Smith, 2003) and are called in this case Multi-Objective Evolutionary Algorithms (MOEAs).

Pareto 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 (Pareto, 1896). Dominance is a partial order that could be established among vectors defined over an n-dimensional space. Figure 1 shows an example of how we apply 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.

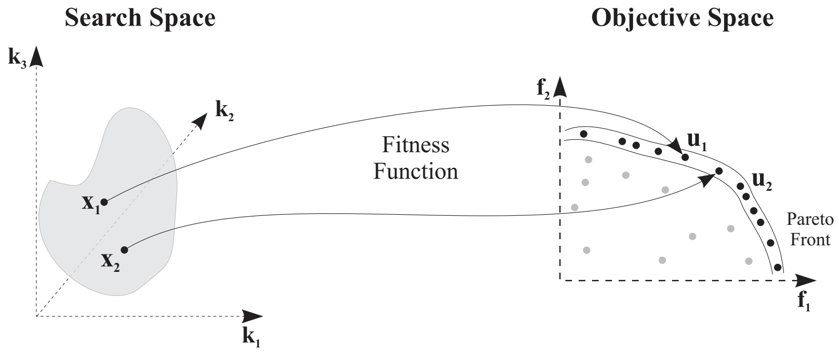

Figure 1. Illustrative example of the concept of Pareto Dominance and a Fitness Function application.

There are some basic definitions based on the Pareto concept that must be considered. We assume a multi-objective maximization problem (the definitions for a minimization problem are similar):

Definition 1 Pareto Dominance *(Coello Coello et al., 2007): 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* $\mathbf{u} \succeq \mathbf{v}$*) if and only if* $\mathbf{u}$ *is partially greater than* $\mathbf{v}$*, i.e.,* $\forall i \in \{1, \ldots, k\},\ u_i \geq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i > v_i$*.*

Definition 2 Pareto Optimality *(Coello Coello et al., 2007): 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 *(Coello Coello et al., 2007): For a given MOOP,* $F(\mathbf{x})$*, and a feasible region for that MOOP,* Ω*, the* ***Pareto Optimal Set****,* $\mathcal{P}^*$*, is defined as the set of Pareto optimal solutions:*

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

Definition 4 Pareto Front *(Coello Coello et al., 2007): 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 the representation of the Pareto optimal set in fitness space:*

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

Aggregative MOEAs. In the non-Pareto EAs, the objectives are combined to obtain a single evaluation value to be used for the selection mechanism. For example, if the harmonic mean is used to aggregate k objectives represented by the fitness functions $f_1, \ldots, f_k$, a single evaluation value $H(\boldsymbol{x})$ can be computed as follows:

$$H(\boldsymbol{x}) = \frac{k}{\sum_{i=1}^{k} f_i(\boldsymbol{x})}.$$

Elitism. Besides the Pareto or non-Pareto strategies, the EAs can be classified in elitist and non-elitist EAs. Elitism in the context of single-objective EAs means that the best solution found so far during the search always survives to the next generation. For the multi-objective case, all non-dominated solutions discovered by a multi-objective EA are considered elite solutions. Elitist EAs favor the best individuals of a population by carrying them directly over the next generation. Therefore, the difference between elitist and non-elitist MOEAs resides in that the first ones use a mechanism to retain the non-dominated individuals. Elitist EAs tend to outperform their non-elitist counterpart. However, implementation of elitism in MOEAs is not straightforward. Two common strategies to maintain elitism are (i) maintaining elitist solutions in the population, and (ii) sorting elitist solutions in an external archive.

Niching. Another important aspect of multi-objective optimization problems is that they usually require the identification of multiple optima, either global or local. Niching methods have been developed to maintain population diversity and simultaneously explore many peaks. The best-known strategies aimed at niching are crowding (De Jong, 1975) and fitness sharing (Goldberg & Richardson, 1987). In both cases, the goal is to obtain solutions uniformly distributed over the Pareto front.

Crowding methods favor individuals from less crowded regions. A notion of crowding distance is defined to quantify how crowded a region is. The crowding distance of a solution i is a measure of the search space around i which is not occupied by any other solution in the population. Details on how this measure is implemented can be found in (Deb, 2001).

Fitness sharing is a diversity preserving mechanism that modifies the search landscape by reducing the payoff in densely populated regions. The fitness of an individual is lowered by an amount nearly equal to the number of similar individuals in the population. The main disadvantage of this method is that it is necessary to provide the niche size parameter, which defines a neighborhood of solutions in the objective space.

A Review of MOEAs. There is a vast literature in MOEAs. The first multi-objective EA was proposed by Schaffer (1985) and is known as VEGA (Vector Evaluated Genetic Algorithm). A characteristic of this MOEA is that each subpopulation is evaluated with respect to a different objective and it tends to converge to the extreme of each objective. Afterwards, several Pareto-based EAs were developed. One of the best known MOEAs that applies Pareto ranking is MOGA (Multi-objective Genetic Algorithm) (Fonseca & Fleming, 1993). This MOEA is a simple extension of single objective EAs and typically has slow convergence. A salient aspect of MOGA is the use of fitness sharing as a diversity preserving mechanism. Another Pareto-based algorithm that focuses on preserving diversity is NPGA (Niched Pareto Genetic Algorithm) (Horn, Nafpliotis, & Goldberg, 1994). A central feature of this algorithm is the use of a very simple selection process that uses niche count as tie-breaker. SPEA (Strenght Pareto Evolutionary Algorithm) (Zitzler & Thiele, 1999) is a well-tested elitist MOEA that attempts to approximate the Pareto-optimal set of solutions by applying ranking based on an external archive of non-dominated solutions. It preserves diversity by applying clustering to eliminate part of the external population. An improved version of SPEA is SPEA-2 (Zitzler, Laumanns, & Thiele, 2001), which implements a fine-grained fitness assignment strategy, a density estimation technique based on the k-th nearest neighbor, and an enhanced archive truncation method. Another MOEA that applies ranking based on non-domination sorting and a fitness sharing mechanism is NSGA (Nondominated Sorting Genetic Algorithm) (N. Srinivas & Deb, 1994). Differently from most MOEAs, the NSGA algorithm has fast convergence. NSGA-II (Nondominated Sorting Genetic Algorithm – II) (Deb, Agrawal, Pratap, & Meyarivan, 2002) is another MOEA based on non-domination sorting. Some of the key aspects of this algorithm are its diversity-preserving mechanism based on crowding distances, the application of elitism and its fast convergence.

NSGA-II is one of the most studied and efficient MOEAs, 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 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 rank is assumed. After ranking the solutions, a population of n offsprings, Q_0, is created using recombination, mutation and a diversity-preserving binary tournament selection operator based on crowding distances. According to this selection operator, known as crowded tournament selection, a solution i wins a tournament with another solution j if he following conditions are true:

- If solution i has a better (smaller) rank than solution j,
- If the ranks of both solutions are the same, the solution located in the less crowded region is preferred.

The NSGA-II procedure for the ith generation is outlined in the following steps:

1. Let Q_i be the offspring population, which is created from the parent population P_i using the crowded tournament selection operator, recombination and mutation.
2. A combined population $R_i = P_i \cup Q_i$ of size $2n$ is formed.
3. R_i is ordered according to non-domination (i.e., each solution is assigned a rank). Since all previous 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.
4. 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, NSGA-II uses once again the crowding distance strategy to favor solutions located in less crowded regions.

In addition to NSGA-II, an elitist non-linear aggregation alternative was used. This scheme was implemented by computing the harmonic mean of the individual fitness functions. It is worth mentioning that because the aggregative MOEA constructs a linear ranking the selection mechanism does not require the computation of the Pareto fronts, which significantly reduces the computational cost of the algorithm. The PISA platform (Bleuler, Laumanns, Thiele, & Zitzler, 2003) was used to implement the strategies analyzed in this work.

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., (Rocchio, 1971)).
- **Unsupervised Feedback**: It applies blind relevance feedback, and typically assumes that the top k documents returned by a search process are relevant (e.g., (Buckley, Singhal, & Mitra, 1995)). This is 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., (Jordan & Watters, 2004)).

A Review of Context-based Retrieval Systems. During recent years several techniques that formulate queries from the user context have been proposed. For example, Watson (Budzik et al., 2001) uses contextual information from documents that users are manipulating to automatically generate Web queries from the documents, using a variety of term-extraction and weighting techniques to select suitable query terms. Watson then filters the matching results, clusters similar HTML pages, and presents the pages to the user as suggestions. Another context-based system is the Remembrance Agent (Rhodes & Starner, 1996) which operates inside the Emacs text editor and continuously monitors the user's work to find relevant text documents, notes, and emails previously indexed. Other systems such as Letizia (Lieberman, 1995) and WebWatcher (Armstrong, Freitag, Joachims, & Mitchell, 1995) use contextual information compiled from past browsing behavior to provide suggestions on related Web pages or links to explore next. SenseMaker (Baldonado & Winograd, 1997) is an interface that facilitates the navigation of information spaces by providing task specific support for consulting heterogeneous search services. The system helps users to examine their present context, move to new contexts or return to previous ones. SenseMaker presents the collection of suggested documents in bundles (their term for clusters), which can be progressively expanded, providing a user-guided form of incremental search.

The EXTENDER system (Maguitman, Leake, Reichherzer, & Menczer, 2004; Maguitman et al., 2005) applies an incremental technique to build up context descriptions. Its task, is to generate brief descriptions of new topics relevant to a knowledge model under construction. Suitor (Maglio, Barrett, Campbell, & Selker, 2000) is a collection of "attentive agents" that gather information from the users by monitoring users' behavior and context, including eye gaze, keyword input, mouse movements, visited URLs and software applications on focus. This information is used to retrieve context-relevant material from the Web and databases. Other methods support the query expansion and refinement process through a query or browsing interface requiring explicit user intervention (Scholer & Williams, 2002; Billerbeck, Scholer, Williams, & Zobel, 2003). 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.

Rocchio's Method. The best-known algorithm for relevance feedback has been proposed by Rocchio (1971). 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 (Amati, Carpineto, & Romano, 2004), 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 (Kwok & Chan, 1998).

The Bo1 and Bo1* *Methods.* A successful generalization of Rocchio's method is the Divergence from Randomness mechanism with Bose-Einstein statistics (Bo1) (Amati, 2003).

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.

The main problem of the Bo1 query refinement method is that its effectiveness is correlated with the quality of the top-ranked documents returned by the first-pass retrieval. If relevance feedback is available, it is possible to implement a supervised version of the Bo1 method, which we will refer to as Bo1*. This new method is identical to the Bo1 method except that rather than considering the top-ranked documents to assign weights to terms, we look only at the top-ranked documents which are known to be relevant. Once the initial queries have been refined by applying the Bo1* method on the training set, they can be used on a different set. The Bo1* method can be regarded as a supervised version of the Bo1 method and because it uses relevance information it offers a more fair basis for comparison against the MOEA-based methods proposed in this work.

Evolutionary Algorithms for Information Retrieval

Initial attempts to use EAs in information retrieval date back to the late 1980's. The focus at that time was on the use of techniques to evolve better document descriptions to aid indexing or clustering (Gordon, 1988; Raghavan & Agarwal, 1987). EA techniques have also been applied to term-weight reinforcement in query optimization (Frieder & Siegelmann, 1991; Yang & Korfhage, 1993; Petry, Buckles, Prabhu, & Kraft, 1993). Nick and Themis (2001) propose to use EAs for Web search by describing an intelligent Web assistant that uses EAs to evolve a set of keywords and logical operators. The evolved set of keywords and operators are then used to construct high-dimensional concepts based on the user's interest. Other Web search approaches have focused on providing the user with a reduced set of pages covering a predefined topic of interest (Caramia, Felici, & Pezzoli, 2004) and on extending the initial set of results by means of improved queries (Leroy, Lally, & Chen, 2003).

Chen, Shankaranarayanan, and She (1998) propose to apply inductive query-by-example (IQBE) techniques based on genetic algorithms and other machine learning techniques to identify new documents which are similar to documents initially suggested by users. These techniques were shown to outperform traditional techniques based on relevance feedback. A comparative study of different MOEA strategies to learn boolean queries in the context of the IQBE paradigm is presented by López-Herrera, Herrera-Viedma, and Herrera (2009). Based on a number of experiments, the authors of this study conclude that NSGA-II is one of the best MOEAs for the purpose of learning boolean queries.

A related research area deals with the development of evolving agents that crawl the Web to search for topical material (Hsinchun, Yi-Ming, Ramsey, & Yang, 1998; Martin-Bautista, Vila, & Larsen, 1999; Menczer et al., 2004). The ultimate goal of a topical crawler is similar to the goal of our research, i.e., to collect resources relevant to a topic. However, the techniques used by topical crawlers are different from the ones discussed in this article. Our approach assumes that there is an underlying index, which can be accessed through queries formulated to a search interface. Topical crawlers, on the other hand, build their own indices by visiting pages on the Web graph. A comprehensive literature review of Web-based evolutionary algorithms can be found in (Kushchu, 2005).

A MOEA Approach to Evolve Topical Queries

The goal of this research work is to automatically formulate and evolve topical queries to retrieve material based on a thematic context. In order to accomplish this goal 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. If the effectiveness of a query is determined by a single objective, then a linear order is imposed on the formulated queries. On the other hand, if multiple-objectives are involved, a linear order can be inferred by applying aggregation techniques, or if the objectives are not aggregated, a Pareto ranking scheme can be applied.

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. In particular, the mutation mechanisms can be implemented in such a way that novel terms, i.e., terms that are not in the initial user context, are brought into play.

Population and Representation of Chromosomes

The search space Q is constituted by all the possible queries that can be formulated to a search interface. Thus the population of chromosomes is a subset of such queries. Consequently, each chromosome is represented as a list of terms $t^1, t^2, \ldots, t^q$, where each term t^i 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. The number of terms in each of the initial queries will be random, with a constant upper bound on the query size. Although the sizes of the initial queries are never more than a predefined constant, the sizes of some queries in subsequent generations can exceed this limit. This is because applying the crossover operator can change the offspring size.

While all terms in the initial population of queries come from the initial 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.

Fitness Functions

The fitness function defines the criterion for assessing the quality of a query. Our conception of high-quality query is based on the query's ability to retrieve material relevant to the thematic context when submitted to a search engine. There are different ways in which information needs can be quantified (Cooper, 1968). For example, a user may be interested in obtaining only one relevant document, some arbitrary number n of documents, all the relevant documents, or a given proportion of the relevant documents, among other possibilities. This will result in different formulations of the fitness function.

The selection of a weighting model for assigning scores to the retrieved documents will strongly influence query performance. For this purpose we have used a vector representation of the query together with the TFIDF weighting function (Baeza-Yates & Ribeiro-Neto, 1999). In this model, queries are interpreted following the disjunctive semantics, which means that a single matching term is sufficient to retrieve a document. 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, *Precision@10* (Hawking & Craswell, 2001), 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 $\mathit{Precision@10} : Q \times T \to [0, 1]$ which can numerically evaluate an individual query q in terms of precision at rank 10 for a given topic t as follows:

$$\mathit{Precision@10}(q,t) = \frac{|D_{q10} \cap D_t|}{|D_{q10}|},$$

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

The fitness function $\mathit{Recall} : Q \times T \to [0, 1]$ is defined as the fraction of relevant documents D_t that are in the answer set D_q:

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

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

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

The $F^\star$ measure is an adaptation of the F_1 measure, which is the weighted harmonic mean of precision and recall (Rijsbergen, 1979).

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), the classical binary tournament selection operator is used. In this case, 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. In addition, elitism is applied to prevent losing the best queries. For the multi-objective case, the crowded tournament selection operator discussed earlier is used.
- **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 a *mutation pool* (described next).

Mutation Pool

The mutation pool is a set that initially contains terms extracted from the thematic context under analysis. As the system collects relevant content, the mutation pool is updated with new terms from the relevant documents recovered by the system. This procedure brings new terms to the scene, allowing a broader exploration of the search space.

Evaluation

Data Collection and Experimental Setup

To run our evaluations we collected 448 topics from the Open Directory Project (ODP – http://dmoz.org). 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. For each topic we collected all of its URLs as well as those in its subtopics. The minimum size for each selected topic was 100 URLs and the language was restricted to English. The total number of collected pages was more than 350.000. The Terrier framework (Ounis, Lioma, Macdonald, & Plachouras, 2007) 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 $\frac{2}{3}$ of its pages were used to create training indices of different sizes and the remaining $\frac{1}{3}$ of the corpus was used for testing. For each topic, we used its ODP description to characterize it as an initial set of terms. The ODP topic description is available in the ODP distribution files and typically consists of a few sentences briefly describing the content of the pages indexed under that category.

One of our research questions focuses on identifying mechanisms for evolving and ranking queries that simultaneously satisfy multiple objectives. For that purpose we analyzed three different settings:

1. Single-objective EAs: We attempt to optimize each objective independently of the other. Independent runs are performed for each objective. In our analysis, we have considered the *Precision@10* and *Recall* objectives, discussed earlier.

2. Non-Aggregative Multi-objective EA: Multiple objectives are simultaneously optimized using NSGA-II, with a different fitness function used for each objective.

3. Aggregative Multi-objective EA: Although we attempt to optimize multiple objectives in the same run, a single fitness function that aggregates them as a scalar value is used. For that purpose, we have used the $F^\star$ measure introduced earlier.

The evaluations were performed in two stages. For the first stage we used the training set to evolve queries and monitored the performance of the queries at each generation. We evaluated if the queries at the initial generations were outperformed by those in subsequent generations. In this case, the performance of the system during the initial generation can be taken as a baseline in the sense that the queries for the initial population are formed using terms selected directly from the ODP topic description (no evolutionary mechanisms have yet been applied at that point). This analysis allows us to study how effective are the queries evolved by our strategy based on EAs, in comparison to queries generated directly from the initial topic description.

For the second stage we took the queries evolved in the first stage and tried them on the test set. The goal was to determine if the evolved queries for a particular topic were effective on a new corpus (the test set). Note that the training and test sets contain the same topics (and therefore the same topic descriptions) but different documents. In this stage we compared the MOEA strategies to a baseline strategy that forms queries using terms extracted directly form the topic descriptions, and to the Bo1 and Bo1* query refinement techniques, which were presented earlier in this article.

The Performance Evaluation

Out of the 448 topics used to populate the indices, a subset of 110 randomly selected topics was used to evaluate the three settings discussed in the previous section. For the training stage we run the EAs with a population of 250 queries, a crossover probability of 0.7 and a mutation probability of 0.03. The selection of values for these parameters was guided by previous studies (Cecchini, Lorenzetti, Maguitman, & Brignole, 2008). For each analyzed topic the population of queries was randomly initialized using its ODP description. The size of each query was a random number between 1 and 32. We should point out that existing search engines use up to a fixed number of query terms and ignore subsequent ones (e.g., Google's query size limit is 32 terms). Applying the crossover operation could eventually increase the query size beyond this limit. The terms that go beyond that limit are ignored when the query is formulated (because they occur beyond the query size limit). But may be taken into account in subsequent generations when these terms become part of an offspring query after recombination takes place.

Single-objective EAs. In our first experimental setting, we run a single-objective EA for 200 generations on a training index containing $\frac{2}{3}$ of the corpus with the purpose of maximizing *Precision@10.* In figure 2 we show the evolution of *Precision@10* (left) and *Recall* (right) for the ODP topic *BUSINESS/BUSINESS_SERVICES/CONSULTING* (CONSULTING). In the charts we plot the average performance over the full population of queries at each generation. As can be observed on the left of this figure, when the objective to be maximized is *Precision@10*, near-optimal queries were obtained after a small number of generations. However, this was at the cost of very low *Recall* values.

Unsurprisingly, very low *Precision@10* values were achieved when the objective to be maximized was *Recall*, as shown in figure 3. 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.

NSGA-II. In order to evolve queries that simultaneously attempted to achieve high levels of *Precision@10* and *Recall* we run NSGA-II for 300 generations on a training index

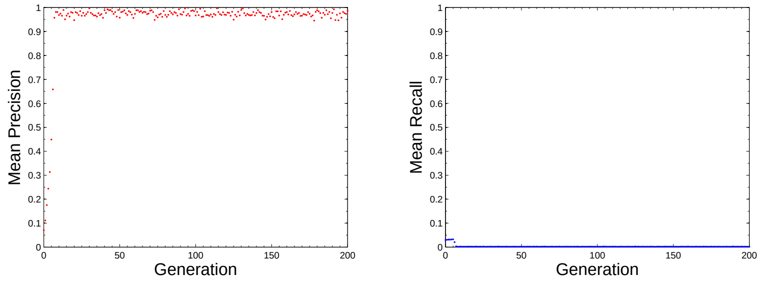

Figure 2. The evolution of *Precision@10* (left) and *Recall* (right) for the topic CONSULTING when the objective to be maximized is *Precision@10*.

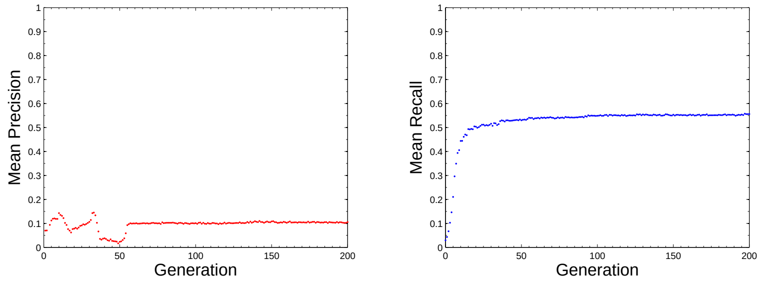

Figure 3. The evolution of *Precision@10* (left) and *Recall* (right) for the topic CONSULTING when the objective to be maximized is *Recall*.

containing $\frac{2}{3}$ of the corpus. In figure 4 we plotted the average performance achieved at each generation for the topic CONSULTING by looking at *Precision@10* (left), *Recall* (center) and $F^\star$ (right). It is interesting to note that this strategy allowed to achieve very high levels of *Precision@10* without compromising *Recall*. As a result, the $F^\star$ measure reflects very good performance.

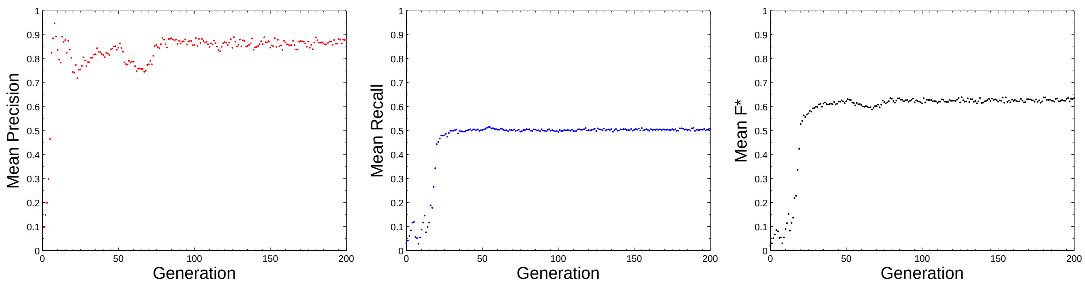

Figure 4. The evolution of *Precision@10* (left), *Recall* (center) and $F^\star$ (right) for the topic CONSULTING when running NSGA-II.

The trend observed in figure 4 for the topic CONSULTING was also observed for the other topics in our corpus. Table 1 presents the means and confidence intervals over 110 topics for the first and last generations based on *Precision@10*, *Recall* and $F^\star$. In addition it shows the average improvement achieved by NSGA-II when the performance of the first-

Table 1: First generation vs. last generation of queries evolved with NSGA-II: mean, confidence intervals and improvement for average query quality based on 110 topics.

Average *Precision@10*				
	N	mean	95% C.I.	improvement
First Generation	110	0.0611	[0.0568,0.0654]	
Last Generation	110	0.9204	[0.8992,0.9415]	1407%

Average *Recall*				
	N	mean	95% C.I.	improvement
First Generation	110	0.0459	[0.0429,0.0488]	
Last Generation	110	0.5981	[0.5680,0.6283]	1204%

Average $F^{\star}$				
	N	mean	95% C.I.	improvement
First Generation	110	0.0219	[0.0205,0.0234]	
Last Generation	110	0.7119	[0.6859,0.7378]	3144%

generation queries is compared to the evolved ones. This comparison shows that NSGA-II 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.

Aggregative MOEA. To complete the first stage of our evaluation, we monitored the performance of the aggregative MOEA by analyzing the *Precision@10*, *Recall* and $F^{\star}$ values achieved throughout 300 generations on a training index containing $\frac{2}{3}$ of the corpus. The charts in figure 5 show the average performance of the evolved queries at each generation for the topic CONSULTING while table 2 summarizes the statistics for the 110 topics considered in our evaluation.

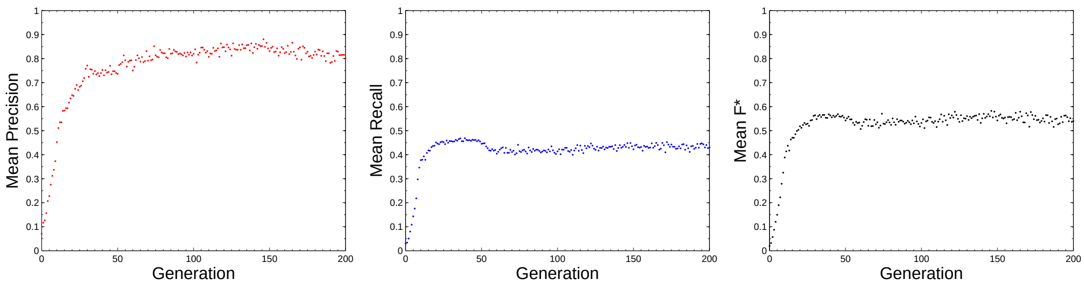

Figure 5. The evolution of *Precision@10* (left), *Recall* (center) and $F^{\star}$ (right) for the topic CONSULTING when applying the aggregative multi-objective algorithm based on $F^{\star}$.

Table 2: First generation vs. last generation of queries evolved with the aggregative MOEA: mean, confidence intervals and improvement for average query quality based on 110 topics.

Average *Precision@10*				
	N	mean	95% C.I.	improvement
First Generation	110	0.0609	[0.0566,0.0652]	
Last Generation	110	0.9099	[0.8833,0.9365]	1394%

Average *Recall*				
	N	mean	95% C.I.	improvement
First Generation	110	0.0459	[0.0429,0.0488]	
Last Generation	110	0.5641	[0.5331,0.5951]	1130%

Average $F^\star$				
	N	mean	95% C.I.	improvement
First Generation	110	0.0219	[0.0205,0.0233]	
Last Generation	110	0.6804	[0.6525,0.7084]	3005%

We have observed 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 (high *Precision@10* and high *Recall*) the results of applying an aggregative approach to rank and evolve queries are comparable to those obtained by a non-aggregative, more computationally expensive approach. At this stage, however, the conclusions are partial since we haven't analyzed yet whether the evolved topical queries have a good performance on a corpus different from the one used for training the MOEAs.

Evaluating Query Performance on the Test Set. In order to determine if the topical queries evolved by the MOEAs are effective when used on a new corpus we computed the *Precision@10* and *Recall* of the evolved queries for each of the 110 topics on the test set. For this analysis we used training indices of two different sizes and an independent test set. The first training index contained $\frac{2}{3}$ of the corpus while the second contained $\frac{1}{2}$ of the corpus. The remaining $\frac{1}{3}$ of the corpus was used as the test set.

The question addressed here is whether the evolved queries are superior to those generated directly from the initial topic description and to those generated by the query-refinement techniques reviewed earlier in this article. Note that the topic description is a reliable instrument for baseline evaluation given the fact that it consists of a summary provided by human editors familiar with the topic. On the other hand, Bo1 is considered to be one of the most highly effective successor of Rocchio's query-refinement method (Amati et al., 2004), while Bo1* is an even more challenging basis for comparison as it uses relevance information.

The charts presented on figures 6, 7 and 8 depict the average query performance for the individual topics using *Precision@10*, *Recall* and $F^\star$ respectively. In this case $\frac{2}{3}$ of the corpus was used for training and the remaining $\frac{1}{3}$ of the corpus was used for testing. Each of the 110 topics corresponds to a trial and is represented by a point. The point's vertical coordinate (z) corresponds to the average performance of NSGA-II (chart on the left-hand side of the figure) or the aggregative MOEA (chart on the right-hand side of the figure), while the point's other two coordinates (x and y) correspond to the baseline and the Bo1* method. Note that different markers are used to illustrate the cases in which each of the tested methods performs better than the other two. In addition we can observe the projection of each point on the x-y, x-z and y-z planes. For the case of *Precision@10* (figure 6) we observe that NSGA-II outperforms both the baseline and Bo1* for 100 of the tested topic, while the aggregative MOEA is the best method for 105 topics. Figure 7 shows that for the case of *Recall* NSGA-II outperforms both the baseline and Bo1* for 96 of the tested topics, while the aggregative MOEA is the best method for 91 topics. This can be verified by observing that for the x-z and y-z plane most of the points appear above the diagonal. Finally, for the case of the $F^\star$ measure (figure 8) NSGA-II is superior to the baseline and the Bo1* method for 101 topics while the aggregative MOEA is the best method for 105 of the tested topics.

These charts show that in fact the evolved queries are much more effective than the baseline queries and the Bo1* queries on the test set. An important finding is that we can confirm that the tested MOEAs are not overfitting the training data.

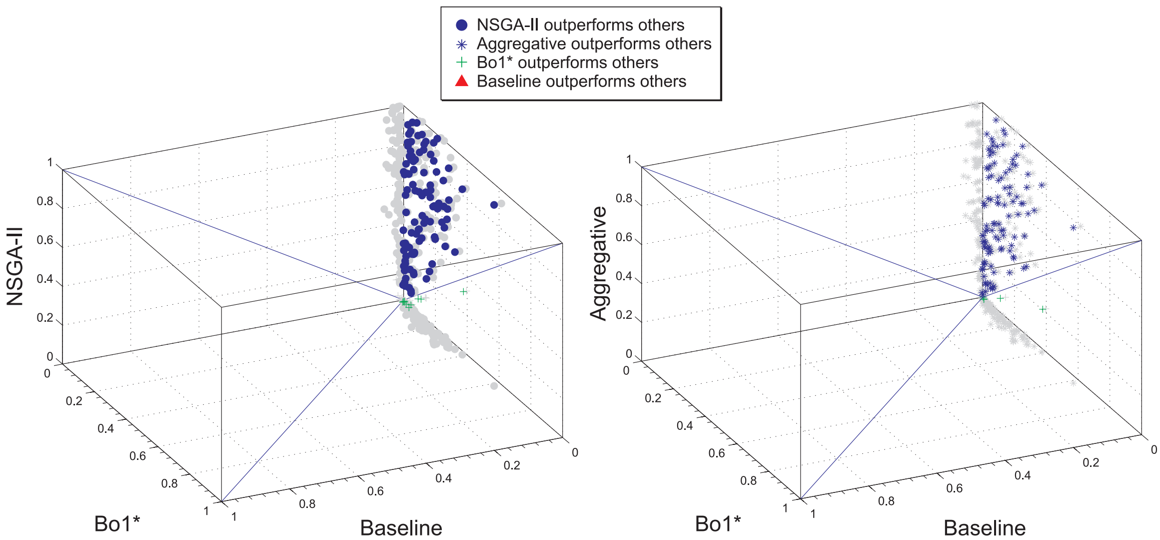

Figure 6. A comparison of the baseline, Bo1* and NSGA-II (left) and a comparison of the the baseline, Bo1* and the aggregative MOEA (right) for 110 topics based on *Precision@10*.

Table 3 presents the statistics comparing the performance of the baseline queries against the performance of the queries obtained by the Bo1 and Bo1* methods, and the queries evolved with NSGA-II and the aggregative MOEA. This comparison shows that the queries evolved by NSGA-II and the aggregative MOEA outperform those generated by applying the baseline, Bo1 and Bo1* techniques. In addition, the improvements are

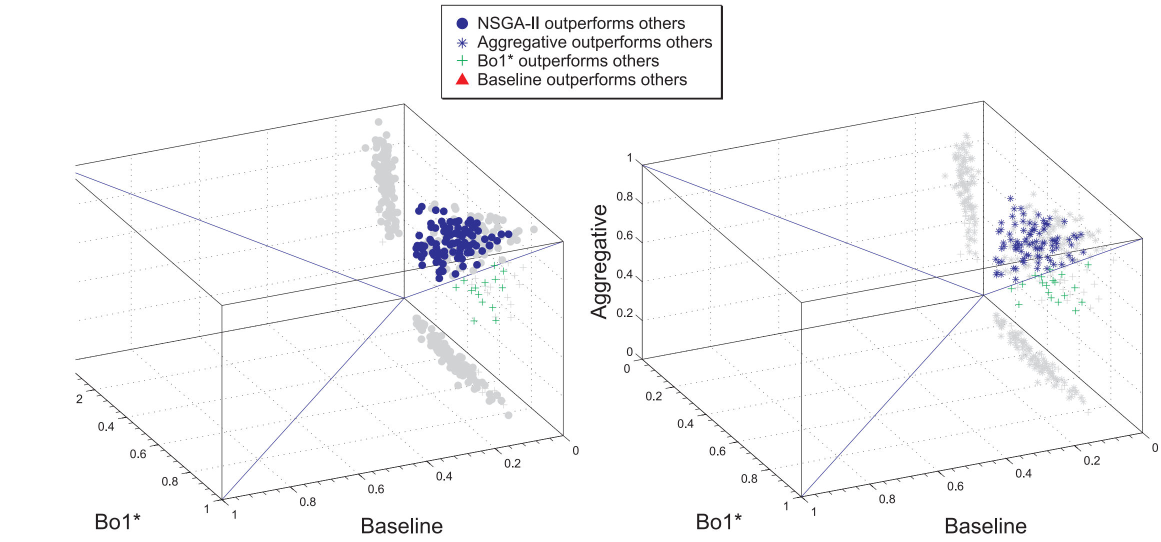

Figure 7. A comparison of the baseline, Bo1⋆ and NSGA-II (left) and a comparison of the the baseline, Bo1⋆ and the aggregative MOEA (right) for 110 topics based on *Recall*.

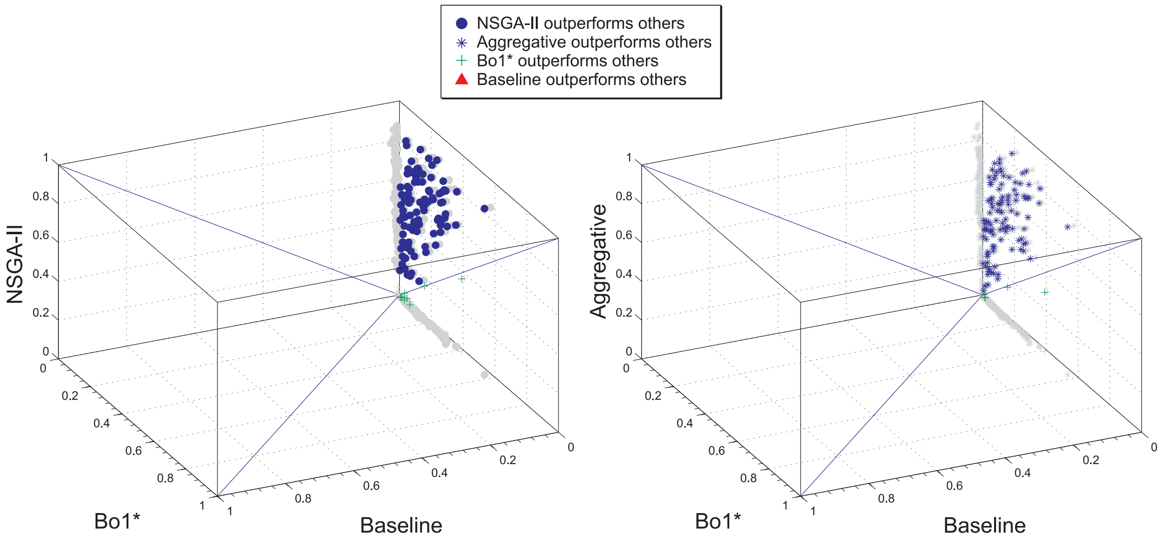

Figure 8. A comparison of the baseline, Bo1⋆ and NSGA-II (left) and a comparison of the the baseline, Bo1⋆ and the aggregative MOEA (right) for 110 topics based on the $F^{\star}$ measure.

statistically significant. On the other hand, this comparison allows us to conclude that NSGA-II and the aggregative MOEA have similar performance on the test set.

The above analysis was performed using training sets of different sizes. We observe that changing the size of the training set from $\frac{1}{2}$ to $\frac{2}{3}$ of the corpus does not significantly affect the overall performance of the evolved queries on the test set. This may be due to the fact that despite having significantly reduced topic size, each topic preserves a sufficient

Table 3: Baseline vs. queries refined with the Bo1 and Bo1* methods, queries evolved with NSGA-II and queries evolved with the aggregative MOEA: mean, confidence intervals and improvement over baseline for average query quality based on 110 topics. TrS refers to the size of the index used for training.

Average *Precision@10*					
	TrS	N	mean	95% C.I.	improvement
Baseline	–	110	0.0153	[0.0131,0.0174]	–
Bo1	–	110	0.0157	[0.0134,0.0181]	3%
Bo1*	$\frac{1}{2}$	110	0.1149	[0.0983,0.1314]	651%
	$\frac{2}{3}$	110	0.1379	[0.1171,0.1587]	802%
NSGA-II	$\frac{1}{2}$	110	0.5530	[0.4970,0.6089]	3516%
	$\frac{2}{3}$	110	0.5359	[0.4791,0.5928]	3405%
Aggregative MOEA	$\frac{1}{2}$	110	0.5307	[0.4759,0.5855]	3370%
	$\frac{2}{3}$	110	0.4958	[0.4423,0.5493]	3142%

Average *Recall*					
	TrS	N	mean	95% C.I.	improvement
Baseline	–	110	0.0512	[0.0480,0.0545]	–
Bo1	–	110	0.1283	[0.1203,0.1364]	150%
Bo1*	$\frac{1}{2}$	110	0.4295	[0.4011,0.4579]	738%
	$\frac{2}{3}$	110	0.4404	[0.4108,0.4700]	768%
NSGA-II	$\frac{1}{2}$	110	0.6007	[0.5723,0.6292]	1073%
	$\frac{2}{3}$	110	0.5860	[0.5582,0.6138]	1044%
Aggregative MOEA	$\frac{1}{2}$	110	0.5646	[0.5347,0.5945]	1002%
	$\frac{2}{3}$	110	0.5587	[0.5296,0.5877]	990%

Average $F^\star$					
	TrS	N	mean	95% C.I.	improvement
Baseline	–	110	0.0076	[0.0069,0.0085]	–
Bo1	–	110	0.0156	[0.0134,0.0178]	105%
Bo1*	$\frac{1}{2}$	110	0.1254	[0.1079,0.1430]	1549%
	$\frac{2}{3}$	110	0.1410	[0.1208,0.1613]	1753%
NSGA-II	$\frac{1}{2}$	110	0.5255	[0.4819,0.5690]	6805%
	$\frac{2}{3}$	110	0.5037	[0.4603,0.5471]	6519%
Aggregative MOEA	$\frac{1}{2}$	110	0.4978	[0.4553,0.5403]	6442%
	$\frac{2}{3}$	110	0.4771	[0.4360,0.5183]	6170%

Table 4: Pearson correlation between topic size and *Precision@10*. TrS refers to the size of the index used for training.

	TrS	correlation	p-value
NSGA-II	$\frac{1}{2}$	0.3130	0.0009
	$\frac{2}{3}$	0.3011	0.0014
Aggregative MOEA	$\frac{1}{2}$	0.3557	0.0001
	$\frac{2}{3}$	0.2449	0.0099

amount of information to guide the EAs towards good solutions. As part of our future work we expect to run additional tests in order to gain a deeper understanding of the relation existing between the size of the training data and query performance.

The documents in our corpus are not uniformly distributed among topics. We have noticed that the most populated topics typically achieved higher *Precision@10* than the less populated ones. The correlation analysis shown in table 4 reveals a positive relationship between the topic size and the *Precision@10* achieved by the queries evolved for the topics. A similar analysis for the other performance measures used in this work is not informative because, by definition, topic size (i.e., number of relevant documents) negatively affects recall. Finally, we should mention that although the most populated topics tend to yield very high *Precision@10* values, we also noticed that some smaller topics performed very well too. This brings to light the important fact that the ability to learn good queries for a topic depends on many factors, which are not only quantitative (e.g., topic size) but also qualitative (e.g., topic vocabulary). An analysis of how different quantitative and qualitative aspects of a corpus affect query performance would be instructive but is beyond the scope of this work.

Discussion

This section discusses several aspects of the proposed approach and explores possible future directions.

Efficiency

In an analysis of the time efficiency of the method, we should consider (1) the time needed to submit each query, as well as to retrieve the results, (2) the time needed to compute the fitness functions associated with each query, and (3) the time needed to compare individuals and select the most promising ones. Other aspects, such as the applications of the crossover or mutation operators do not have a significant impact on the time efficiency of the method. The cost of submitting a query and retrieving its search results will highly depend on whether the search is performed on a local or remote corpus. For the study reported in this work, we have used a set of topical documents indexed locally and therefore the access time was considerably lower than in previous studies (Cecchini et al., 2008). Each document can be accessed in constant time.

The implemented fitness functions can be computed by identifying the portion of the top ten retrieved documents that are relevant (for the case of precision at ten) and the portion of all the relevant documents that have been retrieved (for the case of recall). For this computation it is sufficient to have access to relevancy information while the analysis of the content of each retrieved document is unnecessary. Other fitness functions than can be applied in this scenario, such as those based on similarity discussed in (Cecchini et al., 2008), are more computationally intensive as they require representing the retrieved documents in vector space and using cosine similarity to compare them to the user context.

Finally, the time complexity of the Pareto strategy for identifying the best individuals at each generation is $O(m * n^2)$ where m is the number objectives ($m = 2$ in our tests) and n is the size of the population (Deb, 2001). On the other hand, the time complexity of the aggregative strategy at each generation is $O(n)$. It is worth mentioning that by applying a more efficient algorithm for non-dominated sorting the overall run-time complexity of NSGA-II can be significantly reduced (Jensen, 2003).

Although we have run each of the tested MOEAs for 300 generations, we have observed that 30 generations are usually sufficient to identify individuals with high performance. Each generation involves the submission of as many queries as individuals are in the population, combined with the retrieval and analysis of the returned results. As a consequence the cost for the entire process is high and hence it is important to note that the proposed approach is not intended to provide a real-time search mechanism. In this sense, the goal of the proposed system is similar to that of a thematic Web crawler except that instead of crawling the Web, it will crawl the results returned by a search engine for a sequence of evolved queries.

Diversity Preservation

As discussed earlier, niching techniques help the population converge not to a single point, but to the entire Pareto front. Since our analysis for Pareto-based MOEAs was performed using NSGA-II, the niching strategy adopted was based on the use of crowding distances. In particular, the distances between solutions were computed in the objective space. However, alternative approaches could be adopted to maintain diversity. Typically, a measure of similarity (or distance) among individuals needs to be computed, which can be based on either the individuals' genotype or phenotype. Our problem domain is suitable for computing similarity among individuals by either approach. For example, an approach based on genotypic similarity could favor diversity by looking at the vector representations of the individual queries, computing the cosine similarity among them and penalizing those queries that are similar to many other queries. On the other hand, an approach based on phenotypic similarity could take into consideration information associated with the retrieval results of each query. A simple implementation of query phenotypic similarity could consist in counting the number of common results returned by two queries.

A main issue with niching techniques has to do with the computational cost associated with the estimation of some measure of similarity among individuals. An appealing and less expensive technique that favors the exploration of diverse solutions is local selection (Menczer, Degeratu, & Street, 2000). This approach is similar to, but more efficient than fitness sharing. It has the advantage of allowing parallel implementation for distributed tasks, an important pro in the information retrieval scenario. This is achieved by evaluating the fitness function by an external environment that provides appropriate data structures for maintaining shared resources. This technique is amenable to the information retrieval task, where each retrieved document can be marked so that the same document does not yield payoff multiple times. In the future we plan to test adaptations of these niching methods as well as others found in the literature.

Convergence

In a theoretical analysis of the convergence of canonical single-objective Genetic Algorithms (CGAs), Rudolph (1994) showed that "convergence to the global optimum is not an inherent property of the CGA but rather is a consequence of the algorithmic trick of keeping track of the best solution found over time." Later he proposed a series of multi-objective algorithms (Rudolph, 2001) for which the convergence to the Pareto-optimal set is guaranteed. These algorithms, however, do not guarantee maintaining diversity.

In the case of NSGA-II, on the other hand, the application of the crowding approach to niching guarantees diversity while convergence to the true Pareto-optimal front is not ensured (Laumanns, Thiele, Deb, & Zitzler, 2002). However, the lack of a guaranteed convergence property is not a serious limitation for our problem because (1) although NSGA-II does not ensure true convergence to the Pareto-optimal set it does well in progressing closer to it, and (2) a query can be considered effective even if it is not an optimal one. As mentioned earlier, we may be interested in finding many near optimal queries rather than a single optimal one. In scenarios in which both properties of converging to the true Pareto-optimal front and maintaining diversity are important, a class of MOEAs based on the concept of ϵ-dominance can be applied (see (Laumanns et al., 2002) for details).

Favoring Exploration with the Mutation Pool

A powerful aspect of the proposed technique 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 (Cecchini, Lorenzetti, Maguitman, & Brignole, 2007).

We looked at novel terms introduced by the search process and found out that many of them were very good descriptors of the topic at hand, even when those terms were not in the initial thematic context description. For example, the topic *SHOPPING/PETS/BIRDS* has the following terms in its initial topic description: "*birds, canaries, considered, doves, finches, include, kept, parrot, pets, pigeons, poultry, softbills, species, wild*". One of the best individuals evolved by NSGA-II for this topic was the query **q**="*administr, aviari, belli,* **birds**, *cabela,* **canaries**, *chat, cleanest, cockatoo, harrison, hitag2, keepac, lamp, purpl, realtim, rib, tasti, toi, twinleath*", with *Precision@10* = 1 and *Recall* = 0.8968. Note that all stems, except for "**birds**" and "**canaries"** are not part of the initial topic description. This highlights the importance of exploring new vocabularies through the use of a mutation pool.

Aggregative vs. Multi-objective Strategies

The aggregative multi-objective approach is less computationally expensive than the non-aggregative one. In addition, differently from the Pareto-based optimization methods, the aggregative techniques allow to define ranking functions that favor one objective over the other. For example, a generalization of the F_1 measure can be defined as follows (Rijsbergen, 1979):

$$F_\beta = \frac{(1+\beta^2)\cdot(\mathit{Precision}\cdot\mathit{Recall})}{(\beta^2\cdot\mathit{Precision}+\mathit{Recall})}$$

This measure allows to attach β times as much importance to recall as precision.

On the other hand, it is a theoretical fact that most aggregative techniques are not able to generate concave portions of the Pareto front (Coello Coello et al., 2007) and therefore they may not be applicable to certain optimization scenarios. Therefore, non-aggregative strategies will be preferred over aggregative ones for these scenarios. The exploration of this problem is part of our future research work.

Conclusions

This article proposes to look at the problem of context-based search as a multi-objective optimization problem and discusses EA-based strategies to learn increasingly better queries. We have analyzed three different settings to deal with multiple objectives: (1) performing independent runs for each objective using single-objective EAs, (2) simultaneously optimizing multiple objectives in the same run by applying a Pareto-based ranking method, and (3) optimizing an aggregative objective function that combines multiple objectives. We have adopted two classical information retrieval performance measures to establish the objectives to be optimized: precision at ten and recall. After evaluating the different settings, we have noted the limitations of the single-objective EA strategy and observed that the Pareto-based and aggregative techniques have comparable performance.

The proposed method is fully automatic as long as a training corpus is available and the objective functions have been defined. It's applicable to any domain for which it is possible to generate term-based characterizations of a context. An important result derived from our evaluations is that the evolved queries did not overfit the training data. The implication of this result is that a corpus for which relevance assessments are available (such as ODP) can be used to evolve effective queries for the topics at hand. 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.

A number of other machine learning techniques could be used for generating topical queries. For example, the training data could be used to create a classifier that discriminates between relevant and irrelevant documents. Drucker, Shahraray, and Gibbon (2001) present an analysis of relevance feedback based on support vector machines and show that it performs better than classical techniques in specific scenarios, such as topics with low visibility. However, support vector machine as well as other successful techniques for classifier construction typically takes the full training set as input. If the training set is too large this may be unfeasible. Our proposal, on the other hand, takes an incremental approach. Therefore, instead of taking the full training set to build a classifier, it incrementally improves a set of queries based on the retrieved material.

In (Cecchini et al., 2008) we proposed to apply single-objective genetic algorithms to evolve topical 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, yielding a semi-supervised evolution of the queries. 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.

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. Our proposal shares insights and motivations with the study presented by López-Herrera et al. (2009). However, this study differs from ours in several respects: 1) it uses a boolean model rather than a vector space model for document retrieval; 2) it uses a set of documents and the IQBE paradigm to train the algorithms while we use a topic description and a topic ontology; 3) it is run on corpora of much smaller sizes than ours; 4) it compares non-aggregative MOEA strategies among themselves while we include an aggregative strategy as well as a comparison to a baseline, the Bo1 and the Bo1* methods. By taking precision at ten and recall as measures of performance effectiveness we attempt to maximize the ability of the system to retrieve relevant documents while at the same time holding back non-relevant ones. However, many other performance measures exist in the literature and new measures for evaluation are constantly being proposed. In the future we expect to run additional experiments applying other performance measures coming from the information retrieval and Web search communities as well as ad-hoc ones. In addition, we plan to test different parameter settings for the EAs and other mutation, crossover, selection and niching techniques.

References

Amati, G. (2003). *Probabilistics Models for Information Retrieval based on Divergence from Randomness.* Unpublished doctoral dissertation, Department of Computing Science, University of Glasgow, UK.

Amati, G., Carpineto, C., & Romano, G. (2004). Query Difficulty, Robustness and Selective Application of Query Expansion. In *Advances in Information Retrieval, 26th European Conference on IR research* (pp. 127–137). Berlin / Heidelberg: Springer.

Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995, March). WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments* (pp. 6–12). Palo Alto, CA, USA: AAAI Press.

Baeza-Yates, R. A., & Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval.* Reading, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Baldonado, M. Q. W., & Winograd, T. (1997, March). SenseMaker: an information-exploration interface supporting the contextual evolution of a user's interests. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 11–18). Atlanta, Georgia: ACM Press.

Billerbeck, B., Scholer, F., Williams, H. E., & Zobel, J. (2003). Query expansion using associated queries. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (pp. 2–9). New York, NY, USA: ACM Press.

Bleuler, S., Laumanns, M., Thiele, L., & Zitzler, E. (2003). PISA – A Platform and Programming Language Independent Interface for Search Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *Evolutionary Multi-criterion Optimization* (Vol. 2632, pp. 494–508). Berlin: Springer-Verlag.

Buckley, C., Singhal, A., & Mitra, M. (1995). New Retrieval Approaches Using SMART. In D. K. Harman (Ed.), *Proceedings of the Fourth Text REtrieval Conference (TREC-4)* (Vol. Special Publication 500-236). Gaithersburg, MD, USA: National Institute of Standards and Technology (NIST).

Budzik, J., Hammond, K. J., & Birnbaum, L. (2001). Information Access in Context. *Knowledge Based Systems*, *14*(1–2), 37–53.

Caramia, M., Felici, G., & Pezzoli, A. (2004). Improving search results with data mining in a thematic search engine. *Computers & Operations Research*, *31*(14), 2387–2404.

Cecchini, R. L., Lorenzetti, C. M., Maguitman, A. G., & Brignole, N. B. (2007, October). 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: Universidad Nacional del Nordeste.

Cecchini, R. L., Lorenzetti, C. M., Maguitman, A. G., & Brignole, N. B. (2008). Using genetic algorithms to evolve a population of topical queries. *Information Processing and Management*, *44*(6), 1863–1878.

Chakrabarti, S., Berg, M. van den, & Dom, B. (1999). Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, *31*(11–16), 1623–1640.

Chen, H., Shankaranarayanan, G., & She, L. (1998). A Machine Learning Approach to Inductive Query by Examples: An Experiment Using Relevance Feedback, ID3, Genetic Algorithms, and Simulated Annealing. *Journal of the American Society for Information Science*, *49*(8), 693–705.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd ed.). Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Cooper, W. S. (1968, January). Expected search length: A single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*, *19*(1), 30–41.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to Algorithms* (1st ed.). Cambridge, MA, USA: MIT Press.

Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.* London, UK: John Murray.

Deb, K. (2001). *Multi–Objective Optimization Using Evolutionary Algorithms.* Chichester, W Sussex, UK: John Wiley & Sons, Ltd.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions Evolutionary Computation*, *6*(2), 182–197.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems.* Unpublished doctoral dissertation, Ann Arbor, MI, USA.

Drucker, H., Shahraray, B., & Gibbon, D. C. (2001). Relevance Feedback using Support Vector Machines. In *Proceedings of the 18th International Conference on Machine Learning* (pp. 122–129). Williamstown, MA, USA: Morgan Kaufmann.

Eiben, A. E., & Smith, J. E. (2003). *Introduction to Evolutionary Computing* (1st ed.). Berlin / Heidelberg: Springer.

Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 416–423). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Frieder, O., & Siegelmann, H. T. (1991). 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* (pp. 230–239). New York, NY, USA: ACM.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the second international conference on genetic algorithms on genetic algorithms and their application* (pp. 41–49). Hillsdale, NJ, USA: L. Erlbaum Associates Inc.

Gordon, M. (1988). Probabilistic and genetic algorithms in document retrieval. *Communications of the ACM*, *31*(10), 1208–1218.

Hawking, D., & Craswell, N. (2001). Overview of the TREC-2001 Web track. In E. M. Voorhees & D. K. Harman (Eds.), *Proceedings of the Tenth Text REtrieval Conference TREC-2001* (Vol. Special Publication 500-250, pp. 61–67). Gaithersburg, MD, USA: National Institute of Standards and Technology (NIST).

He, H., Meng, W., Lu, Y., Yu, C., & Wu, Z. (2007). Towards Deeper Understanding of the Search Interfaces of the Deep Web. *World Wide Web*, *10*(2), 133–155.

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor, MI, USA: The University of Michigan Press.

Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In *International Conference on Evolutionary Computation* (pp. 82–87). Orlando, Florida, USA: IEEE.

Hsinchun, C., Yi-Ming, C., Ramsey, M., & Yang, C. C. (1998). An intelligent personal spider (agent) for dynamic Internet/intranet searching. *Decision Support Systems*, *23*(1), 41–58.

Jensen, M. T. (2003). Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, *7*(5), 503–515.

Jordan, C., & Watters, C. R. (2004). 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)* (Vol. 3034, pp. 135–144). Berlin / Heidelberg: Springer.

Kautz, H., Selman, B., & Shah, M. (1997). The Hidden Web. *AI Magazine*, *18*(2), 27–36.

Kraft, R., Chang, C. C., Maghoul, F., & Kumar, R. (2006). Searching with context. In *WWW '06: Proceedings of the 15th International Conference on World Wide Web* (pp. 477–486). New York, NY, USA: ACM.

Kushchu, I. (2005, April). Web-based evolutionary and adaptive information retrieval. *IEEE Transactions on Evolutionary Computation*, *9*(2), 117–125.

Kwok, K. L., & Chan, M. S. (1998). 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* (pp. 250–256). New York, NY, USA: ACM.

Laumanns, M., Thiele, L., Deb, K., & Zitzler, E. (2002). Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation*, *10*(3), 263–282.

Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., et al. (2003). Aiding Knowledge Capture by Searching for Extensions of Knowledge Models. In *KCAP-2003 – Proceedings of the 2nd International Conference on Knowledge Capture* (pp. 44–53). Sanibel Island, FL, USA: ACM Press.

Leake, D. B., Bauer, T., Maguitman, A., & Wilson, D. C. (2000). 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.* (pp. 33–37). Austin, Texas: AAAI Press.

Leroy, G., Lally, A. M., & Chen, H. (2003). The use of dynamic contexts to improve casual internet searching. *ACM Transactions on Information Systems (TOIS)*, *21*(3), 229–253.

Lieberman, H. (1995). Letizia: An Agent That Assists Web Browsing. In C. S. Mellish (Ed.), *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. IJCAI-95* (pp. 924–929). Montreal, Quebec, Canada: Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

Lin, F., & He, G. (2005). An Improved Genetic Algorithm for Multi-Objective Optimization. In *Proceedings of the Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT)* (pp. 938–940). Los Alamitos, CA, USA: IEEE Computer Society.

López-Herrera, A. G., Herrera-Viedma, E., & Herrera, F. (2009). A study of the use of multi-objective evolutionary algorithms to learn Boolean queries: A comparative study. *Journal of the American Society for Information Science and Technology*, *60*(6), 1192–1207.

Maglio, P. P., Barrett, R., Campbell, C. S., & Selker, T. (2000). SUITOR: an attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces* (pp. 169–176). New Orleans, Louisiana, United States: ACM Press.

Maguitman, A., Leake, D., & Reichherzer, T. (2005). Suggesting novel but related topics: towards context-based support for knowledge model extension. In *IUI '05: Proceedings of the 10th International Conference on Intelligent User Interfaces* (pp. 207–214). New York, NY, USA: ACM Press.

Maguitman, A., Leake, D., Reichherzer, T., & Menczer, F. (2004, November). Dynamic Extraction of Topic Descriptors and Discriminators: Towards Automatic Context-Based Topic Search. In *Proceedings of the Thirteenth Conference on Information and Knowledge Management (CIKM).* Washington, DC: ACM Press.

Martin-Bautista, M. J., Vila, M.-A., & Larsen, H. L. (1999). A fuzzy genetic algorithm approach to an adaptive information retrieval agent. *Journal of the American Society for Information Science and Technology*, *50*(9), 760–771.

Menczer, F., Degeratu, M., & Street, W. N. (2000). Efficient and Scalable Pareto Optimization by Evolutionary Local Selection Algorithms. *Evolutionary Computation*, *8*(2), 223–247.

Menczer, F., Pant, G., & Srinivasan, P. (2004, November). Topical web crawlers: Evaluating adaptive algorithms. *ACM Transactions on Internet Technology (TOIT)*, *4*(4), 378–419.

Mitchell, M. (1996). *An introduction to genetic algorithms.* Cambridge, MA, USA: MIT Press.

Nick, Z. Z., & Themis, P. (2001). Web Search Using a Genetic Algorithm. *IEEE Internet Computing*, *5*(2), 18–26.

Ounis, I., Lioma, C., Macdonald, C., & Plachouras, V. (2007, February). 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.

Pareto, V. (1896). *Cours d'économie politique.* Lausanne, Vaud, Switzerland: Rouge.

Petry, F. E., Buckles, B. P., Prabhu, D., & Kraft, D. H. (1993, October). Fuzzy information retrieval using genetic algorithms and relevance feedback. In S. Bonzi (Ed.), *Proceedings of the 56th Annual Meeting of the American Society for Information Science (ASIS)* (Vol. 30, pp. 122–125). Medford, NJ, USA: Learned Information.

Pulido, G. T. (2001). *Optimización Multiobjetivo Usando un Micro Algoritmo Genético.* Unpublished master's thesis, Maestría en Inteligencia Artificial, Universidad Veracruzana, Xalapa, Veracruz, México.

Raghavan, V., & Agarwal, B. (1987). 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* (pp. 241–246). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Rhodes, B., & Starner, T. (1996, April). The Remembrance Agent: A continuously running automated information retrieval system. In *Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)* (pp. 487–495). London, UK: AAAI Press.

Rijsbergen, C. J. v. (1979). *Information Retrieval.* Newton, MA, USA: Butterworth-Heinemann.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The Smart retrieval system - experiments in automatic document processing* (pp. 313–323). Englewood Cliffs, NJ, USA: Prentice-Hall.

Rudolph, G. (1994). Convergence Analysis of Canonical Genetic Algorithms. *IEEE Transactions on Neural Networks*, *5*, 96–101.

Rudolph, G. (2001). Evolutionary Search under Partially Ordered Fitness Sets. In *Proceedings of the International Symposium on Information Science Innovations in Engineering of Natural and Artificial Intelligent systems (ISI 2001)* (pp. 818–822). Dubai, United Arab Emirates: ICSC Academic Press.

Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms* (pp. 93–100). Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc.

Scholer, F., & Williams, H. E. (2002). Query Association for Effective Retrieval. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management* (pp. 324–331). New York, NY, USA: ACM Press.

Somlo, G., & Howe, A. E. (2004). QueryTracker: An Agent for Tracking Persistent Information Needs. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems* (pp. 488–495). Los Alamitos, CA, USA: IEEE Computer Society.

Srinivas, M., & Patnaik, L. M. (1994). Genetic algorithms: A Survey. *Computer*, *27*(6), 17–26.

Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, *2*(3), 221–248.

Yang, J.-J., & Korfhage, R. (1993). Query Optimization in Information Retrieval using Genetic Algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 603–613). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Zitzler, E., Laumanns, M., & Thiele, L. (2001, May). *SPEA2: Improving the Strength Pareto Evolutionary Algorithm* (Tech. Rep. No. 103). Gloriastrasse 35, CH-8092 Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.

Zitzler, E., & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271.