Using Genetic Algorithms to Evolve a Population of Topical Queries

Rocío L. Cecchini [a,c] Carlos M. Lorenzetti [b,c]
Ana G. Maguitman [b,c] Nélida Beatríz Brignole [a,c,d]

[a] *LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica*
[b] *LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial*
[c] *Departamento de Ciencias e Ingeniería de la Computación*
Universidad Nacional del Sur, Av. Alem 1253, (8000) Bahía Blanca, Argentina
phone: 54-291-4595135 fax: 54-291-4595136
[d] *Planta Piloto de Ingeniería Química (UNS-CONICET)*
Cno la Carrindanga km 7, (8000) Bahía Blanca, Argentina
Email: {cr,cml,agm,nbb}@cs.uns.edu.ar

Abstract

Systems for searching the Web based on thematic contexts can be built on top of a conventional search engine and benefit from the huge amount of content as well as from the functionality available through the search engine interface. The quality of the material collected by such systems is highly dependant on the vocabulary used to generate the search queries. In this scenario, selecting good query terms can be seen as an optimization problem where the objective function to be optimized is based on the effectiveness of a query to retrieve relevant material. Some characteristics of this optimization problem are (1) the high-dimensionality of the search space, where candidate solutions are queries and each term corresponds to a different dimension, (2) the existence of acceptable suboptimal solutions, (3) the possibility of finding multiple solutions, and in many cases (4) the quest for novelty. This article describes optimization techniques based on Genetic Algorithms to evolve "good query terms" in the context of a given topic. The proposed techniques place emphasis on searching for novel material that is related to the search context. We discuss the use of a mutation pool to allow the generation of queries with new terms, study the effect of different mutation rates on the exploration of query-space, and discuss the use of a especially developed fitness function that favors the construction of queries containing novel but related terms.

Key words: Web search, context, genetic algorithms, query formulation, novelty

1 Introduction

As search continues to grow in popularity, context-based methods to search the Web are progressively becoming a key component of intelligent information access systems. Taking advantage of the resources available through major search engines can dramatically simplify the process of information access and delivery. However, accessing topical information through existing search engines requires the formulation of appropriate queries, which is highly challenging. In current search engines, there are limits on query length, or if long queries are allowed, they may become too specific, returning very few or no results. This makes it difficult to provide appropriate queries to describe rich thematic contexts. Even if special syntaxes are used to formulate context-based queries, there is no guarantee that the vocabulary used to describe the context will match the vocabulary by which the relevant resources are indexed. The goal of our research work is to design intelligent techniques to automatically refine search queries and to accumulate resources relevant to a thematic context as a whole. In particular, we are interested in collecting novel but related material.

This article describes a framework based on Genetic Algorithms (GAs) that addresses the problem of reflecting topical information when formulating search queries. The proposed framework takes an incremental approach to evolve high-quality queries for retrieving context-relevant textual resources (such as html pages, pdf files, Word files, etc.). It starts by generating an initial population of queries using terms extracted from a thematic context and incrementally evolves those queries based on their ability to retrieve relevant results when presented to a search engine.

Developing methods to evolve high-quality queries and collect context-relevant resources can have important impacts on today's information society. These methods can help build systems for a range of information services:

- **Task-Based Search.** Task-based search systems exploit user interaction with computer applications to determine the user's current task and contextualize information needs [Leake et al., 2000,Budzik et al., 2001]. Basic keyword searches could very easily miss task-relevant pages. By evolving high-quality queries, a task-based search system can automatically generate suggestions that are richly contextualized within the user's task.
- **Resource Harvest 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 et al., 1999,Menczer et al., 2004]. As and 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.

- **Deep Web Search.** 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 et al., 1997,Ntoulas et al., 2005]. 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, the formulation of high-quality queries is of utmost importance at the moment of accessing deep Web sources. For that reason, searching the deep Web in context is an important area of application for the proposed techniques.
- **Support for Knowledge Management.** Effective knowledge management may require going beyond initial knowledge capture, to support decisions about how to extend previously-captured knowledge [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 and starting from an in-progress knowledge model, the techniques discussed here can facilitate the process of capturing knowledge to help extend organizational memories.

After reviewing work related to query adaptation and context-based search, the next section presents a discussion of GAs and their suitability for their application to our research problem. It then reviews other proposals that have applied evolutionary algorithms to solve similar problems. This is followed by a presentation of the main contribution of our work, a GA approach for evolving high-quality queries, followed by an evaluation of the proposal. The article closes with a summary of our conclusions and a discussion of future work.

2 Background

2.1 Query Adaptation and Context-Based Search

Automatic query expansion or reformulation are Information Retrieval (IR) techniques based on the use of a set of documents from which additional terms can be obtained [Attar and Fraenkel, 1977,Gauch and Smith, 1991]. In recent proposals, query reformulation has been used to automatically augment user queries with other terms selected from the user context. A variety of systems pursuing this approach have obtained encouraging results. 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 such system is the Remembrance Agent [Rhodes and 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 et al., 1995] use contextual information compiled from past browsing behavior to provide suggestions on related Web pages or links to explore next. SenseMaker [Baldonado and 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 et al., 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 et al., 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.

2.2 An Overview of Genetic Algorithms

GAs [Holland, 1975] 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 GAs 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 GAs 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 GA 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,Srinivas and Patnaik, 1994,Mitchell, 1996].

2.3 Genetic Algorithms for Context-Based IR

There are a number of reasons why GAs are appropriate to deal with the problem of context-based Web search:

- **Context-Based Web Search as an Optimization Problem.** Generating high-quality queries for context-based search on the Web can be regarded as an optimization problem. The search space of the problem is defined as the set of possible queries that can be presented to a search engine. The objective function to be optimized is based on the effectiveness of a query to retrieve relevant material when presented to a search engine. Depending on the system goals, a measure of query effectiveness can be defined using traditional IR notions such as precision and recall, or other customized performance evaluation metrics.
- **High-Dimensional Space.** Query space is a high-dimensional space, where each possible term accounts for a new dimension. This kind of problems cannot be effectively solved using analytical methods but are natural for GAs.
- **Suboptimal Solution.** Successful Web search requires the formulation of high-quality queries even if the formulated queries are not the optimal ones. GAs do not guarantee the identification of optimal solutions but are usually successful in finding near optimal ones.
- **Multiple Solutions.** Each one of multiple sets of Web pages can represent a satisfactory result for a context-based search. Therefore, we may be interested in finding many high-quality queries rather than a single one. GAs

can be naturally used for multimodal relevance optimization.

- **Exploration and Exploitation.** Finding good combinations of query terms requires exploring different direction of the thematic-context space. This exploration must be independent of the initial population of queries and it may require going beyond the initial set of terms by incorporating novel terms. Such a search process can be effectively performed by applying the genetic operators of crossover and mutation. In addition, the exploitation of the most promising combination of terms is naturally induced by the selection mechanism.

Initial attempts to use evolutionary computation in IR dates back to the late eighties. The focus at that time was on the use of GA techniques to derive better document descriptions to aid indexing or clustering [Gordon, 1988,Raghavan and Agarwal, 1987]. GA techniques have also been applied to term-weight reinforcement in query optimization [Frieder and Siegelmann, 1991,Yang and Korfhage, 1993,Petry et al., 1993]. Nick and Themis (2001) propose to use GAs for Web search by describing an intelligent Web assistant that uses GAs 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 et al., 2004] and on extending the initial set of results by means of improved queries [Leroy et al., 2003]. A related research area deals with the development of evolving agents that crawl the Web to search for topical material [Hsinchun et al., 1998,Martin-Bautista et al., 1999,Menczer et al., 2004]. The ultimate goal of a topical crawler is similar to the goal of the methods proposed here, 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 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].

3 A Genetic Approach for Evolving High-Quality Queries

The goal of this research work is to evolve queries that have the capability of retrieving material similar to the user context when presented to a search interface. In order to accomplish this goal we start with a population of queries composed of terms extracted from the user context and rate each query according to the quality of the search results. 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.

3.1 Population and Representation of Chromosomes

The search space Q is constituted by all the possible queries that can be formulated to a search engine. Thus the population of chromosomes is a subset of such queries. Consequently, each chromosome is represented as a list of terms, where 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 context. The number of terms in each of the initial queries will be random, with a constant upper bound on the query size. While all terms in the initial population of queries come from the initial thematic context, 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.

3.2 Fitness Function

We associate with the search space Q a fitness function $\text{Fitness} : Q \rightarrow [0 \ldots 1]$ which can numerically evaluate individual queries. 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 similar to the thematic context c when submitted to a search engine. The function we propose to measure fitness is

$$\text{Fitness}(\mathbf{q}) = \max_{d_i \in \mathbf{A_q}} (\sigma(c, d_i))$$

where $\mathbf{A_q}$ is the answer set for query $\mathbf{q}$ (set of documents returned by a search engine when $\mathbf{q}$ is used as a query) and $\sigma : D \times D \rightarrow [0 \ldots 1]$ is the similarity measure for a pair of documents (note that the context c can be regarded as a document in D).

Different similarity measures, such as the standard cosine similarity or Jaccard similarity [Baeza-Yates and Ribeiro-Neto, 1999], can be used in the implementation of the fitness function. Besides the standard cosine similarity, we use a specialized similarity measure that favors novel content. This measure will be presented in section 4.4.

One pragmatic difficulty is the use of the complete answer set $\mathbf{A_q}$ in our definition of fitness. Looking at the entire set of pages returned by a search engine is too expensive for practical purposes. Therefore, we only look at the top ten results and only the "snippets" returned by the search engine are used for computing similarity. (The snippet is a text excerpt from the page summarizing the context where the search terms occur.)

3.3 Genetic Operators

A new generation in our GA 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. The probability that a query **q** will be selected is proportional to its own fitness and is inversely proportional to the fitness of the other queries in the current population. This method is known as the roulette-wheel selection.
- **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).

3.4 Mutation Pool

The mutation pool is a set of terms 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 snippets returned by the search engine. This procedure brings new terms to the scene, allowing a broader exploration of the search space.

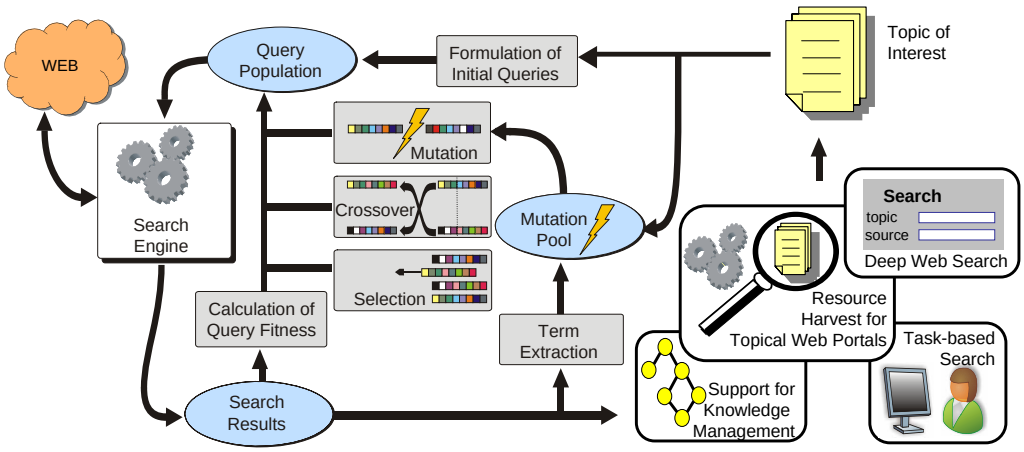

Fig. 1. Architecture for a contextualized Web-search system based on GAs. The application domains can be seen in the bottom right of the figure.

3.5 Proposed System Architecture

Figure 1 depicts the proposed system architecture for different information services that can benefit from Web-search system based on GAs. In the proposed prototype, the system maintains an internal representation of the thematic context. In addition it maintains a population of queries which is incrementally refined as the system evolves. The basic mechanisms that enable the system to evolve queries and retrieve context-based results are the following:

- **Formulation of Initial Queries**. It selects terms from the thematic context and forms suitable queries, which are submitted to a standard search engine (e.g, Google) or entered into individual search forms (e.g., Amazon or PubMed). Initial queries will be automatically formed using a random selection of terms from the thematic context. The sizes of the initial queries are never more than a predefined constant.
- **Calculation of Query Fitness**. This mechanism estimates the relevance of the results returned by a search engine after submitting a query. Based on the estimated relevance it will associate a fitness value with the query. One way the relevance of a search result can be approximated is by computing the similarity between the collected material and the thematic context, but other approaches can be taken.
- **Term Extraction**. This component uses the content returned by a search engine to extract new terms, which are used to update the mutation pool.
- **Selection, Crossover and Mutation**. These mechanisms, described in section 3.3, are in charge of selecting, recombining and mutating the queries of the current population.

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. Notice 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). Interestingly, the eventual increase of query size beyond this limit captures, in a rough sense, the phenomenon of recessive inheritance: some terms that are ignored in a generation (because they occur beyond the query size limit) may be taken into account in subsequent generations when these terms become part of an offspring query after crossover takes place.

4 Evaluation

4.1 Evaluation Criteria

To evaluate the performance of context-based retrieval based on GAs we first had to establish evaluation criteria suitable for this task. As is the case with most systems that access the Web to collect relevant material, existing evaluation schemes face a serious limitation because the set of relevant documents cannot be identified in advance. In our case, we adopted evaluation criteria based on the quality of the best queries at each generation, and the performance improvement is measured as the increase in the quality value as the generations pass. We consider that a proposed context-based-search GA technique is successful if the query quality significantly outperforms that of the initial generations. Notice that 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 thematic context (no evolutionary mechanisms have yet been applied at that point).

In order to propose a measure of query quality we first give a precise definition of similarity between a thematic context and a retrieved result. Assume c is a thematic context and $\mathbf{q}$ a query associated with c. Let $\mathbf{A_q} = \{a_1, \ldots, a_n\}$ be the set of retrieved resources (answer set) for $\mathbf{q}$. A measure of similarity between c and a_i can be computed using the *cosine similarity* defined as:

$$\sigma(c, a_i) = \frac{\overrightarrow{c} \cdot \overrightarrow{a}_i}{\|\overrightarrow{c}\| \cdot \|\overrightarrow{a}_i\|}$$

where $\overrightarrow{c}$ is the vector representation of the thematic context based on the terms in c, and $\overrightarrow{a}_i$ is the vector representation of a_i based on the terms occurring in the corresponding snippet returned by a search engine.

We use σ to define *query quality based on maximum similarity* as follows:

$$\text{Quality\_Max}(\mathbf{q}) = \max_{a_i \in \mathbf{A_q}} (\sigma(c, a_i)).$$

Analogously, we define *query quality based on mean similarity* as:

$$\text{Quality\_Mean}(\mathbf{q}) = \frac{\sum_{a_i \in \mathbf{A_q}} (\sigma(c, a_i))}{|A_q|}$$

Notice that the function Quality_Max is defined exactly as the fitness function presented in section 3.2. On the other hand, Quality_Mean is computed as the average similarity over all pairs (c, a_i). Depending on the task at hand, one notion of query quality may be preferred over the other. For instance, Quality_Max is more appropriate if the goal is to retrieve a unique highly relevant result. Alternatively, Quality_Mean combines the relevance of a set of results and therefore is more appropriate if several results are expected to be useful.

4.2 The Performance Evaluation

A performance test based on our criterion functions requires access to a thematic context c. We generated six thematic contexts to conduct six tests by selecting three topics from the DMOZ directory (dmoz.org) and two webpages from each topic. The topics selected for our tests are *Business*, *Recreation* and *Society*. Each of our initial tests consisted in running the GA five times. Each run consisted in 20 generations, with a population of 60 queries, a crossover probability of 0.7 and a mutation probability of 0.03. The population of queries was randomly initialized using the thematic context. The size of each query was a random number between 1 and 32.

Figures 6, 7 and 8 show the performance of the GA for the selected topics. For each generation, we plotted the average quality of the best query (using both Quality_Max and Quality_Mean) and error bars (at 95% C.I.) resulting from the five runs. In all the tests, the comparison of the query quality obtained through a small number of generations shows that the GA results in statistically significant improvements over the initial generations. (Notice that in all our tests the error bar corresponding to the first generation does not overlap with the error bar at some later generation.) In other words, the GA is able to evolve queries with quality considerably superior to that of the queries generated directly from the thematic context.

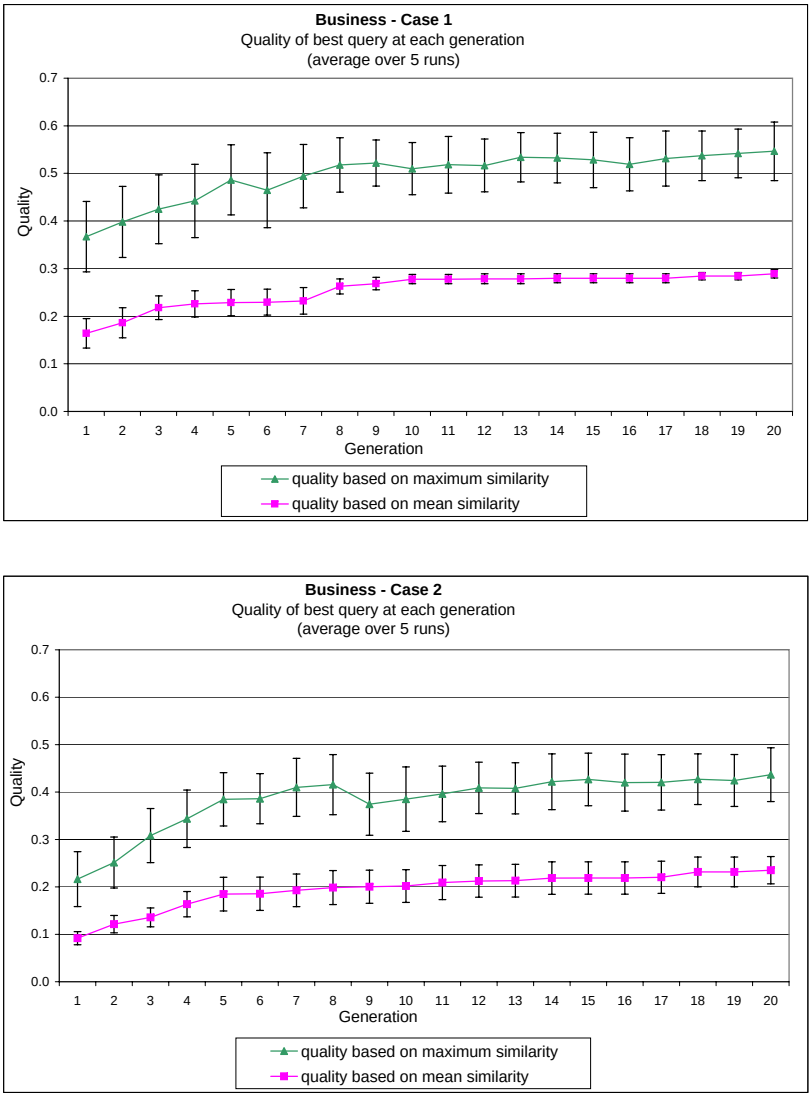

Fig. 2. Two tests showing the average query quality over five independent runs for the topic *Business*.

4.3 The Effect of Different Mutation Rates

An important question is how mutation affects performance and diversity. In this section we address this question by running a number of tests using different mutation rates. Once again, the topics used for these tests were *Business*, *Recreation* and *Society* and the mechanisms implemented for initializing the GA as well as the parameter values were the same as the ones described earlier. However, to analyze the effect of different mutation rates we tested three different settings for the mutation probability: Pm=0 (no mutation), Pm=0.03

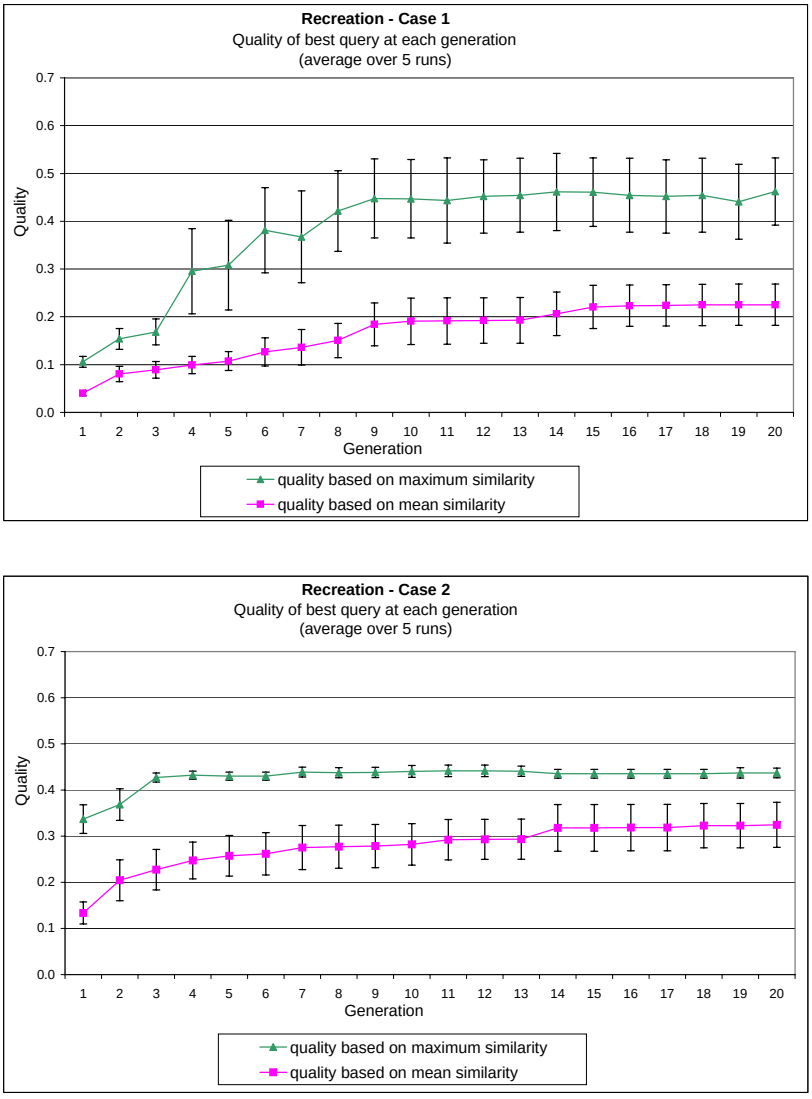

Fig. 3. Two tests showing the average query quality over five independent runs for the topic *Recreation*.

(classical mutation) and Pm=0.3 (hypermutation). The results reported in this section measure query quality based on maximum similarity.

In figure 5 we present three plots that show the evolution of a population of 60 queries across 20 generations for the topic *Business* using different mutation rates. An interesting observation is that the higher the mutation rate, the earlier the algorithm starts to achieve higher similarity scores as well as more diversity. This is consistent with our intuitions, and highlights the importance of mutation at the moment of exploring the query space.

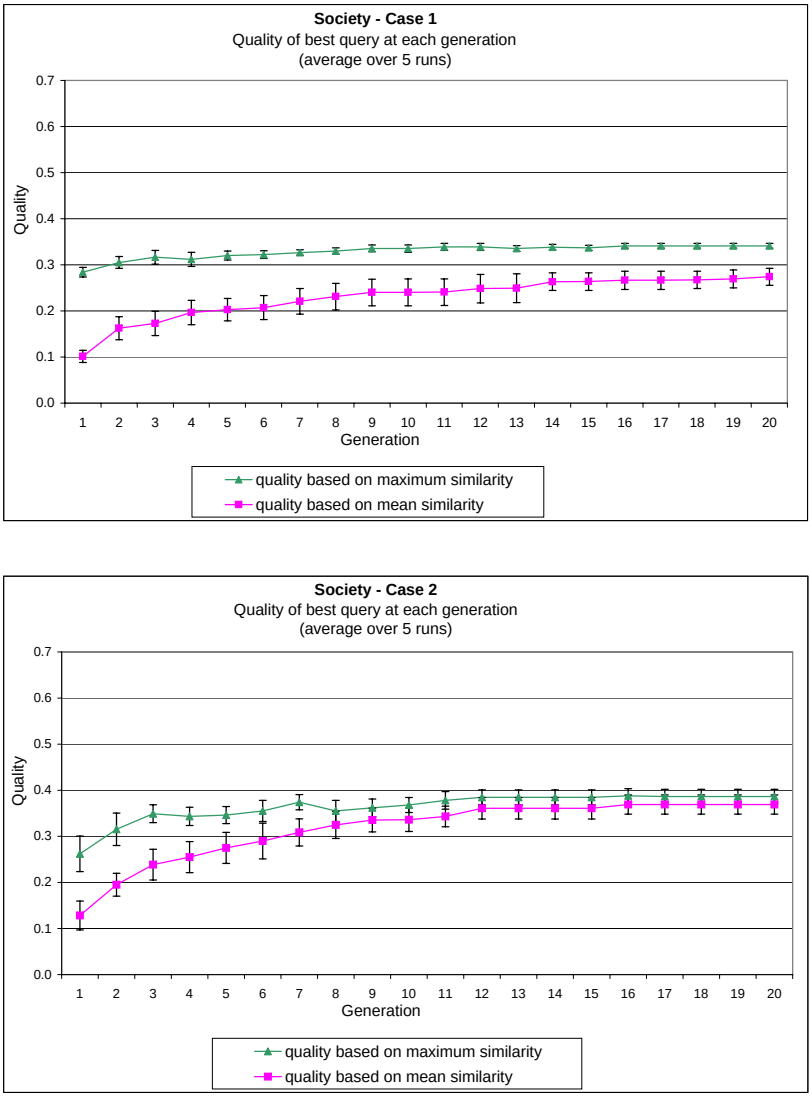

Fig. 4. Two tests showing the average query quality over five independent runs for the topic *Society*.

Figures 6, 7 and 8 show the performance of the GA for one example of each of the topics *Business*, *Recreation* and *Society*, respectively. For each topic we analyzed the effect of running the GAs without mutation, with classical mutation and with hypermutation. In these figures, we plotted the quality of the best query at each generation, averaged over five runs. An interesting observation is that in all the tests, the case with Pm=0 (no mutation) results in the one with the slowest convergence rate towards high-quality queries.

We performed a statistical analysis to compare the query quality obtained at generation 1 with that obtained at generation 20. Tables 1, 2 and 3 show

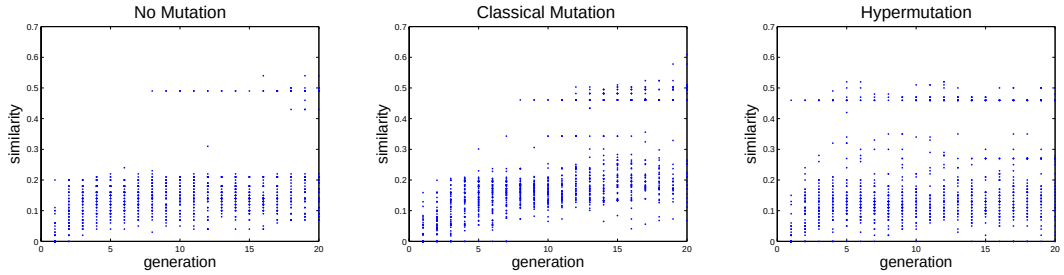

Fig. 5. Scatter plots showing the distribution of similarity values for the best results associated with the individuals at each generation with Pm=0 (left), Pm=0.03 (center) and Pm =0.3 (right) for the topic *Business*.

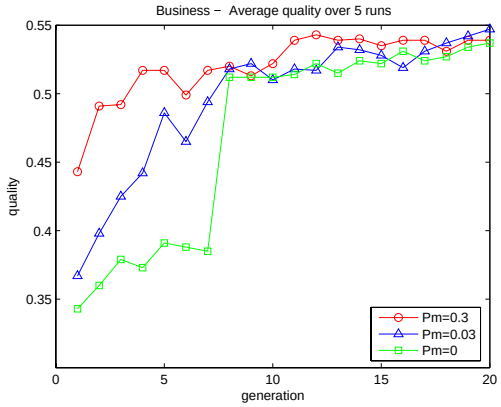

Fig. 6. A test showing the average query quality over five independent runs for the topic *Business* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

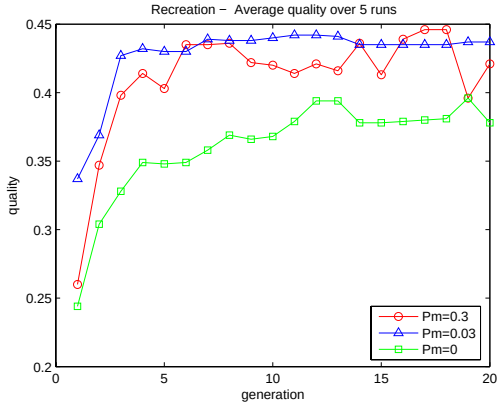

Fig. 7. A test showing the average query quality over five independent runs for the topic *Recreation* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

that for all tests performed there is an important improvement in query quality after 20 generations, and in most cases the improvement is statistically significant (C.I. highlighted in the tables).

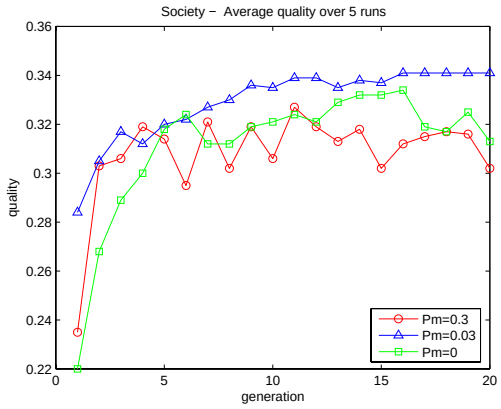

Fig. 8. A test showing the average query quality over five independent runs for the topic *Society* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

	MEAN	95% C.I.
g=1	0.343	**(0.264,0.421)**
g=20	0.537	**(0.500,0.574)**

Pm=0

	MEAN	95% C.I
g=1	0.367	**(0.305,0.429)**
g=20	0.547	**(0.530,0.564)**

Pm=0.03

	MEAN	95% C.I.
g=1	0.443	(0.375,0.511)
g=20	0.539	(0.404,0.673)

Pm=0.3

Table 1
First Generation vs. Last Generation: confidence intervals for average query quality for topic *Business*.

	MEAN	95% C.I.
g=1	0.244	**(0.225,0.264)**
g=20	0.378	**(0.336,0.420)**

Pm=0

	MEAN	95% C.I.
g=1	0.337	**(0.289,0.385)**
g=20	0.437	**(0.395,0.479)**

Pm=0.03

	MEAN	95% C.I.
g=1	0.260	**(0.219,0.300)**
g=20	0.421	**(0.380,0.463)**

Pm=0.3

Table 2
First Generation vs. Last Generation: confidence intervals for average query quality for topic *Recreation*.

	MEAN	95% C.I.
g=1	0.220	**(0.202,0.237)**
g=20	0.313	**(0.243,0.383)**

Pm=0

	MEAN	95% C.I.
g=1	0.284	(0.258,0.311)
g=20	0.341	(0.304,0.378)

Pm=0.03

	MEAN	95% C.I.
g=1	0.235	(0.204,0.267)
g=20	0.302	(0.222,0.381)

Pm=0.3

Table 3
First Generation vs. Last Generation: confidence intervals for average query quality for topic *Society*.

4.4 The Effect of Elitism and Novelty-Driven Fitness

Some of the figures shown in section 4.2 indicate that the quality of some queries can decay from one generation to the next one. This is because the roulette selection mechanism, which is the one we have adopted so far, does not guarantee that the best queries will survive across generations.

Elitism is a mechanism used by some GAs which ensures that the fittest individuals are passed on to the next generation without being altered by the mutation or crossover operators. The elitist strategy ensures that the fitness of

the best individual in the population can never reduce from one generation to the next. Some of the results obtained using the elitist strategy are reported at the end of this section.

The results reported so far assume that the higher the similarity between the thematic context and retrieved material, the higher the quality of the query. However, a few IR approaches take a different position [Budzik et al., 2000,Smyth and McClave, 2001,Maguitman et al., 2005] and postulate that in some circumstances conventional notions of similarity may not be the best criteria for retrieval. In certain scenarios, attaining novelty and diversity may be as important, or even more important, than attaining similarity.

As an alternative to the conventional notion of similarity we propose a new measure of similarity $\sigma^N : Q \times D \times D \rightarrow [0 \ldots 1]$ defined as follows:

$$\sigma^N(\mathbf{q}, c, a_i) = \frac{\overrightarrow{c - \mathbf{q}} \cdot \overrightarrow{a_i - \mathbf{q}}}{\|\overrightarrow{c - \mathbf{q}}\| \cdot \|\overrightarrow{a_i - \mathbf{q}}\|}$$

The notation $\overrightarrow{d - \mathbf{q}}$ stands for the vector representation of the document d in term space with all the values corresponding to the terms from query q set to zero (the same applies to the cases where $d = a_i$ and $d = c$).

This measure of similarity gives rise to a novelty-driven fitness function $\text{Fitness}^N : Q \rightarrow [0 \ldots 1]$ defined as follows:

$$\text{Fitness}^N(\mathbf{q}) = \max_{d_i \in \mathbf{A_q}} (\sigma^N(q, c, d_i))$$

where, as before, $\mathbf{A_q}$ is the answer set for query$\mathbf{q}$.

Analogously, we define two new measures of query quality as:

$$\text{Quality\_Max}^N(\mathbf{q}) = \max_{a_i \in \mathbf{A_q}} (\sigma^N(\mathbf{q}, c, a_i))$$

and

$$\text{Quality\_Mean}^N(\mathbf{q}) = \frac{\sum_{a_i \in \mathbf{A_q}} (\sigma^N(\mathbf{q}, c, a_i))}{|A_q|}.$$

The purpose of defining such functions was twofold:

(1) To ensure that query quality improvement throughout the successive generations was not due only to the fact that we were taking an increasing

number of terms from the originating context. Note that if the original definition of similarity is used, taking many terms from the original context to formulate queries would guarantee high similarity between context and retrieved results. However, by disregarding the query terms when computing σ^N we avoid introducing this bias. This approach allowed us to evaluate whether the performance improvement was simply due to the use of longer queries or to the actual evolution of the queries in a more qualitative way.

(2) We wanted to promote the introduction of novel terms in the queries to bias the search towards novel but related material.

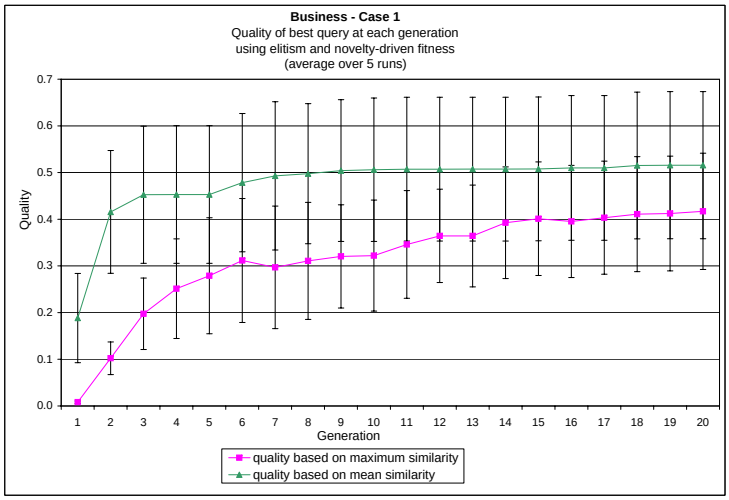

Fig. 9. A test showing the average query quality (based on σ^N) over five independent runs for the topic *Business*.

An evaluation to address the first point was performed with positive results. To illustrate this, figure 9 shows the performance of the GA that applies both elitism and the novelty-driven fitness function for the topic *Business* (other parameters are set as in figure 6). In this case we plotted the average quality of the best query using Quality_Max^N and Quality_Mean^N and error bars (at 95% C.I.) resulting from five runs. Once again, the comparison of the query quality of the first and last generations shows that the GA results in statistically significant improvements. In addition, the elitism mechanism ensured the quality based on maxium similarity function is non-decreasing.

Finally, 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, for a thematic context related to *marketing companies that provide solutions to other companies planning to enter the North-American marketplace* our system discovered novel terms such as: *grow*, *collaboration*, and *financing*, among many others. One of the best individuals evolved by the GA was the query **q**="*planning Chicago compared competitive focus USA pres-*

ence North USA requiring marketing uniquely marketing uniquely North", with $\text{Quality\_Max}^N(\mathbf{q}) = 0.594$

5 Conclusions and Future Work

This article proposes a novel GA approach to Web search based on thematic contexts. Differently from most of the existing GA 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 and does not require relevance feedback from the users. A novel 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. To further investigate the effect of using new terms to form search queries, we have proposed an alternative fitness function that favors those queries containing novel terms over those in which most terms are extracted from the original context. We have observed that many high-quality queries are composed of terms that are not part of the initial context.

The techniques presented in this article are applicable to any domain for which it is possible to generate term-based characterizations of a context. However, query adaptation involves submitting each new query to a search engine to calculate its fitness, which is a time consuming process. Therefore, we expect the proposed techniques to have potential applicability to exploiting thematic context for non-real time systems, where slow response times are acceptable.

Our evaluations show the effectiveness of GA techniques for query generation and refinement. More work, however, needs to be done in this area to make the results richer. We plan to test additional settings for the GA parameters. We have used roulette-wheel selection with and without elitism in the implementation of our methods. However, it is known that other selection methods such as tournament selection are better than roulette-wheel at maintaining diversity. Therefore, we plan to study the impact that other selection methods have on the overall performance of our techniques.

Many search engines allow the formulation of queries with special syntaxes that help results get more specific [Calishain and Dornfest, 2003]. Consequently, an interesting followup study concerns applying genetic programming to evolve queries that take advantage of these special syntaxes. In such methods not only terms will be important at the moment of formulating queries, but boolean operators and other special commands will be considered as well.

There is also much to investigate regarding the fitness function. Choosing a good fitness function is one of the most important aspects in the development of GAs. We based our definition of fitness on two forms of similarity. This was done to keep in line with classical IR systems that typically attempt to match requests with the most similar documents. However, alternative fitness functions can be defined depending on the task at hand.

In the future we expect to run additional experiments applying evaluation metrics coming from the information retrieval community. We expect to adapt these metrics to the Web scenario and to perform human-subject experiments to compare our algorithms to existing ones.

Acknowledgement

We wish to thank anonymous reviewers for helpful suggestions. This research work is supported by 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

[Armstrong et al., 1995] Armstrong, R., Freitag, D., Joachims, T., and Mitchell, T. (1995). WebWatcher: A learning apprentice for the World Wide Web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12.

[Attar and Fraenkel, 1977] Attar, R. and Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *J. ACM*, 24(3):397–417.

[Baeza-Yates and Ribeiro-Neto, 1999] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.

[Baldonado and Winograd, 1997] Baldonado, M. Q. W. and Winograd, T. (1997). 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*, pages 11–18. ACM Press.

[Budzik et al., 2001] Budzik, J., Hammond, K. J., and Birnbaum, L. (2001). Information access in context. *Knowledge based systems*, 14(1–2):37–53.

[Budzik et al., 2000] Budzik, J., Hammond, K. J., Birnbaum, L., and Krema, M. (2000). Beyond similarity. In *Proceedings of the 2000 Workshop on Artificial Intelligence and Web Search*. AAAI Press.

[Calishain and Dornfest, 2003] Calishain, T. and Dornfest, R. (2003). *Google Hacks. 100 Industrial-Strengths Tips and Tools*. O'Reilly.

[Caramia et al., 2004] Caramia, M., Felici, G., and Pezzoli, A. (2004). Improving search results with data mining in a thematic search engine. *Comput. Oper. Res.*, 31(14):2387–2404.

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

[Frieder and Siegelmann, 1991] Frieder, O. and 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*, pages 230–239.

[Gauch and Smith, 1991] Gauch, S. and Smith, J. B. (1991). Search improvement via automatic query reformulation. *ACM Trans. Inf. Syst.*, 9(3):249–280.

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

[Gordon, 1988] Gordon, M. (1988). Probabilistic and genetic algorithms in document retrieval. *Commun. ACM*, 31(10):1208–1218.

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

[Hsinchun et al., 1998] Hsinchun, C., Yi-Ming, C., Ramsey, M., and Yang, C. C. (1998). An intelligent personal spider (agent) for dynamic internet/intranet searching. *Decis. Support Syst.*, 23(1):41–58.

[Kautz et al., 1997] Kautz, H., Selman, B., and Shah, M. (1997). The hidden Web. *AI Magazine*, 18(2):27–36.

[Kushchu, 2005] Kushchu, I. (2005). Web-based evolutionary and adaptive information retrieval. *IEEE Transactions on Evolutionary Computation*, 9.

[Leake et al., 2003] Leake, D., Maguitman, A., Reichherzer, T., Cañas, A., Carvalho, M., Arguedas, M., Brenes, S., and Eskridge, T. (2003). Aiding knowledge capture by searching for extensions of knowledge models. In *Proceedings of KCAP-2003*. ACM Press.

[Leake et al., 2000] Leake, D. B., Bauer, T., Maguitman, A., and 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. Austin, Texas*, pages 33–37. AAAI Press.

[Leroy et al., 2003] Leroy, G., Lally, A. M., and Chen, H. (2003). The use of dynamic contexts to improve casual internet searching. *ACM Trans. Inf. Syst.*, 21(3):229–253.

[Lieberman, 1995] Lieberman, H. (1995). Letizia: An agent that assists Web browsing. In Mellish, C. S., editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. IJCAI-95*, pages 924–929, Montreal, Quebec, Canada. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[Maglio et al., 2000] Maglio, P. P., Barrett, R., Campbell, C. S., and Selker, T. (2000). SUITOR: an attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces*, pages 169–176. ACM Press.

[Maguitman et al., 2005] Maguitman, A., Leake, D., and 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*, pages 207–214, New York, NY, USA. ACM Press.

[Maguitman et al., 2004] Maguitman, A., Leake, D., Reichherzer, T., and Menczer, F. (2004). 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 et al., 1999] Martin-Bautista, M. J., Vila, M.-A., and Larsen, H. L. (1999). A fuzzy genetic algorithm approach to an adaptive information retrieval agent. *J. Am. Soc. Inf. Sci.*, 50(9):760–771.

[Menczer et al., 2004] Menczer, F., Pant, G., and Srinivasan, P. (2004). Topical web crawlers: Evaluating adaptive algorithms. *ACM Trans. Inter. Tech.*, 4(4):378–419.

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

[Ntoulas et al., 2005] Ntoulas, A., Zerfos, P., and Cho, J. (2005). Downloading textual hidden web content through keyword queries. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 100–109, New York, NY, USA. ACM Press.

[Petry et al., 1993] Petry, F., Buckles, B., Prabhu, D., , and Kraft, D. (1993). Fuzzy information retrieval using genetic algorithms and relevance feedback. In *Proceedings of the ASIS Annual Meeting*, pages 122–125.

[Raghavan and Agarwal, 1987] Raghavan, V. and 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*, pages 241–246, Mahwah, NJ, USA. Lawrence Erlbaum Associates, Inc.

[Rhodes and Starner, 1996] Rhodes, B. and Starner, T. (1996). The remembrance agent: A continuously running automated information retrieval system. In *The Proceedings of The First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pages 487–495, London, UK.

[Smyth and McClave, 2001] Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, Canada.*

[Srinivas and Patnaik, 1994] Srinivas, M. and Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, 27(6):17–26.

[Yang and Korfhage, 1993] Yang, J.-J. and Korfhage, R. (1993). 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. Morgan Kaufmann Publishers Inc.