# Genetic Algorithms for Topical Web Search: A Study of Different Mutation Rates [*]

Rocío L. Cecchini[†], Carlos M. Lorenzetti[‡],
Ana G. Maguitman[‡] and Nélida Beatriz Brignole[†§]

{cr,cml,agm,nbb}@cs.uns.edu.ar

[†] LIDeCC - Laboratorio de Investigación y Desarrollo en Computación Científica
[‡] LIDIA - Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
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

[§]Planta Piloto de Ingeniería Química (UNS-CONICET)
Cno la Carrindanga km 7, (8000) Bahía Blanca, Argentina

## Abstract

Harvesting topical content is a process that can be done by formulating topic-relevant queries and submitting them to a search engine. The quality of the material collected through this process 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. Three 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, and (3) the possibility of finding multiple solutions. This paper describes optimization techniques based on Genetic Algorithms to evolve "good query terms" in the context of a given topic. We discuss the use of a mutation pool to allow the generation of queries with novel terms, and study the effect of different mutation rates on the exploration of query-space.

**Keywords:** topical web search, genetic algorithms, query formulation, diversity, query optimization

## 1 INTRODUCTION

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 [6, 14]. As an alternative to focused crawlers, the process of harvesting topical content can be done by formulating topical queries to a search engine and selecting from the answer set those resources that are related to the topic at hand.

[*]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).

To access topical information, appropriate queries must be formed. Finding good combinations of query terms requires exploring different direction of the space of potential queries. This exploration may require going beyond the initial set of terms by incorporating novel terms, which may prove to be useful at the moment of retrieving relevant material.

Refining search queries for topical search can be seen as an optimization problem, in which the search space 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 quality can be defined using traditional information retrieval notions such as precision and recall, or other customized performance evaluation metrics.

Solving this optimization problem is challenging because the query space has a huge number of dimensions, where each possible term accounts for a different dimension. However, for successful topical Web search high-quality queries are useful even if these queries are not the optimal ones. Therefore, it is usually sufficient to identify suboptimal solutions, a characteristic that helps alleviate our optimization problem. Another aspect of the optimization problem at hand is that we may be interested in finding many high-quality queries rather than a single one.

These characteristics make Genetic Algorithms (GAs) good candidates to tackle the problem of finding high-quality queries. This paper describes a framework based on GAs that addresses the problem of reflecting topical information when formulating search queries. The framework, discussed in detail in [5], 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. The contribution of this paper is a study of the effect that different mutation rates have on search results' diversity and quality.

In the next section we review the fundamental concepts of GAs. Then, in section 3 we overview our GA approach for evolving high-quality queries. Section 4 presents a study of the effect of different mutation rates (no mutation, classical mutation and hypermutation) on the diversity and quality of search results. The paper closes with a summary of our conclusions and a discussion of future work.

## 2 GENETIC ALGORITHMS

GAs [8] 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.

# 3 GENETIC ALGORITHMS FOR EXPLORING QUERY SPACE

The goal of our research work is to evolve queries that have the capability of retrieving topic-relevant content when presented to a search interface. In order to accomplish this goal we start with a population of queries composed of terms extracted from a topic description 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 topic description, 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 description of the topic at hand. 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 topic 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 topic description.

## 3.2 Fitness Function

We associate with the search space $Q$ a fitness function 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 topic of interest $t$ when submitted to a search engine. The function we propose to measure fitness is

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

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 topic $t$ can be regarded as a document in $D$).

Different similarity measures, such as the standard cosine similarity or Jaccard similarity [1], can be used in the implementation of the fitness function. 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 $\mathbf{q}$ 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 description of the topic 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.

### 3.5 System Architecture

Figure 1 depicts the proposed system architecture for a topic-based search system based on GAs, which goal is to harvest resources for a topical portal. In the proposed prototype, the system maintains an internal representation of the topic at hand. 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 topic-relevant results are the following:

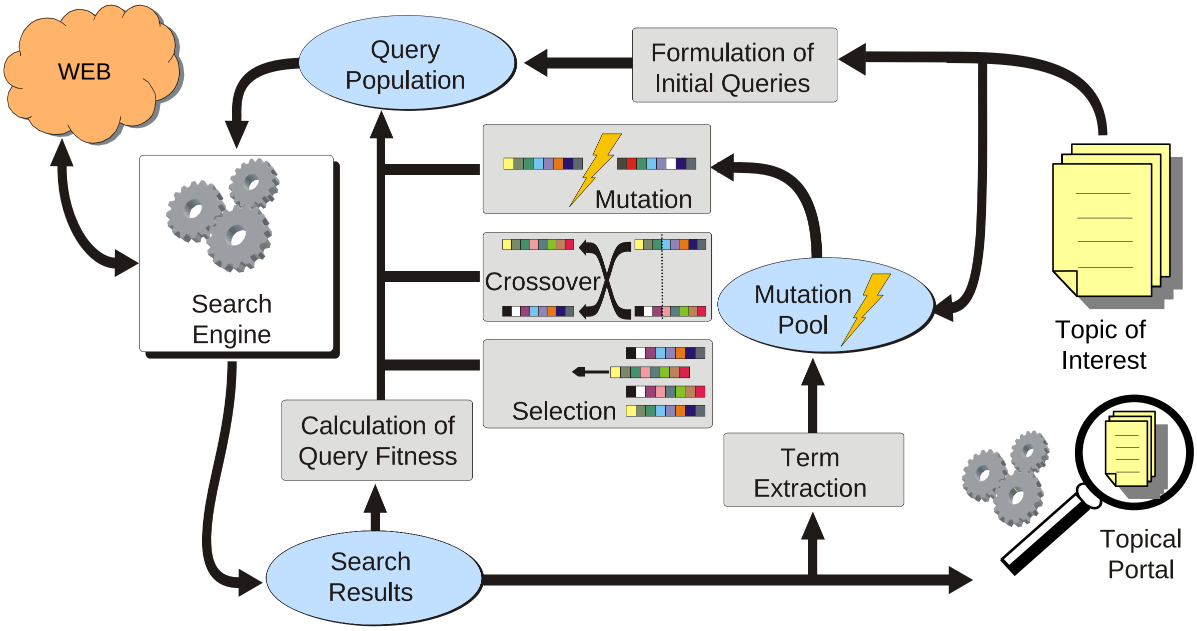

Figure 1: Architecture for a topical search system based on GAs. The collected results are used to populate the indices of a topical portal

- **Formulation of Initial Queries**. It selects terms from the topic of interest (consisting of a document or a set of related documents) 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 topic of interest. 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 topic of interest, 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 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 THE EFFECT OF DIFFERENT MUTATION RATES

In this section we analyze the effect of different mutation rates used by the implemented GAs for addressing our optimization problem. This analysis required access to different topic descriptions. We generated three topic descriptions using webpages from topics selected from the DMOZ directory (dmoz.org). The topics selected for our tests are *Business*, *Recreation* and *Society*. For each of our tests we run the GA five times. Each run consisted in 20 generations, with a population of 60 queries. The population of queries was randomly initialized using the corresponding topic description. The size of each query was a random number between 1 and 32. In all our tests the crossover probability was set to 0.7. To analyze the effect of different mutation rates we tested three different settings for the mutation probability: Pm=0 (no mutation), Pm=0.03 (classical mutation) and Pm=0.3 (hypermutation).

To evaluate the performance of topic-based retrieval based on GAs we adopted evaluation criteria based on the quality of the best queries at each generation. In order to propose a measure of query quality we first give a precise definition of similarity between a topic description and a retrieved result. Assume $t$ is a topic description and $\mathbf{q}$ a query associated with $t$. 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 $t$ and $a_i$ can be computed using the *cosine similarity* defined as:

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

where $\overrightarrow{t}$ is the vector representation of the topic description based on the terms in $t$, 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 $\sigma$ to define *query quality based on maximum similarity* as follows:

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

Notice that the function Quality_Max is defined exactly as the fitness function presented in section 3.2.

## 4.1 The Effect of Mutation on Diversity

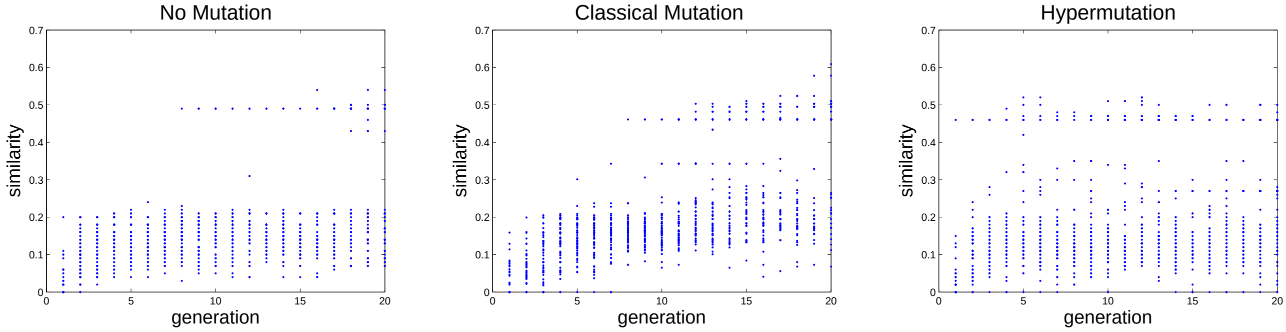

Figure 2: Scatter plots showing the distribution of similarity values for the best results associated with the individuals at each generation with Pm=0 (left), Pm=0.03 (center) and Pm =0.3 (right) for the topic *Business*.

In figure 2 we present three plots that allow us to visualize the evolution of a population of 60 queries across 20 generations for a description of 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.

## 4.2 Evaluation of Query Quality

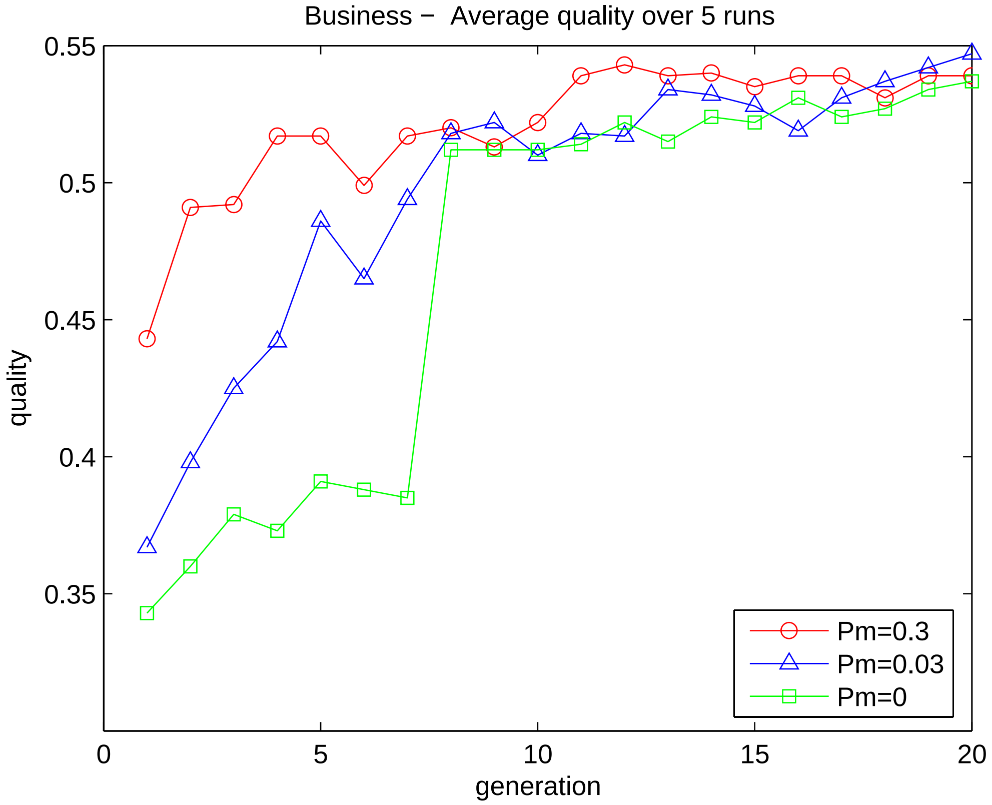

Figure 3: A test showing the average query quality over five independent runs for the topic *Business* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

Figures 3, 4 and 5 show the performance of the GA for 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 the 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.

Finally, 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 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). This allows us to conclude that, in general, the GA is able to evolve queries with quality considerably superior to that of the queries generated directly from the topic description.

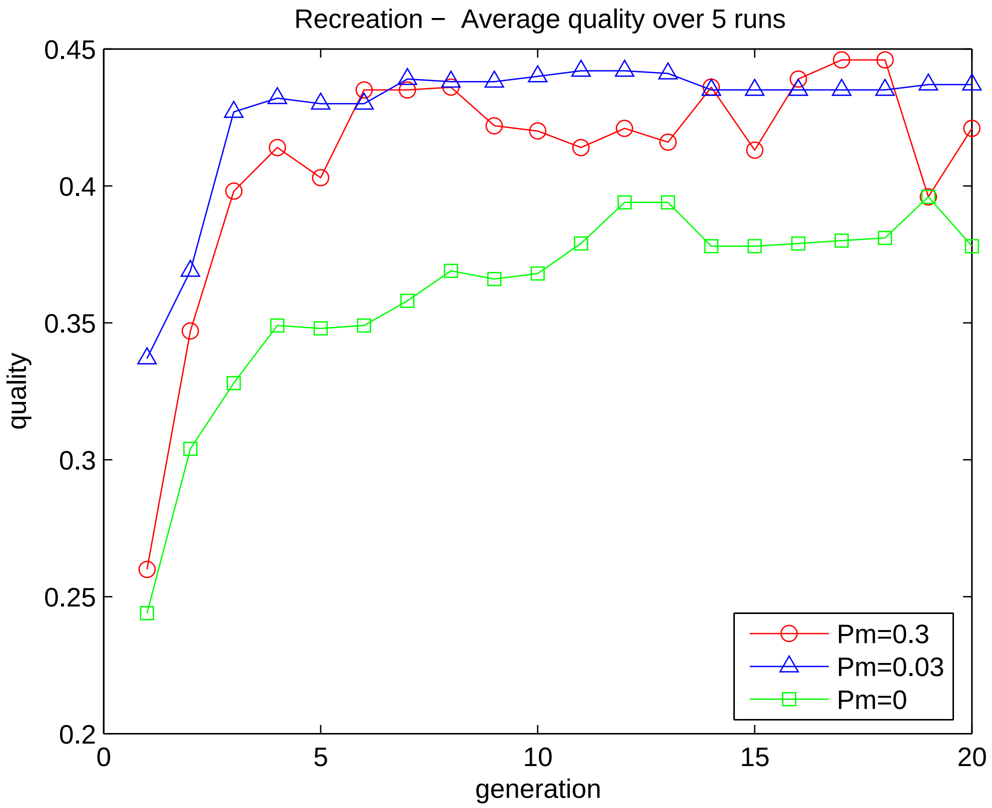

Figure 4: A test showing the average query quality over five independent runs for the topic *Recreation* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

| | 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*.

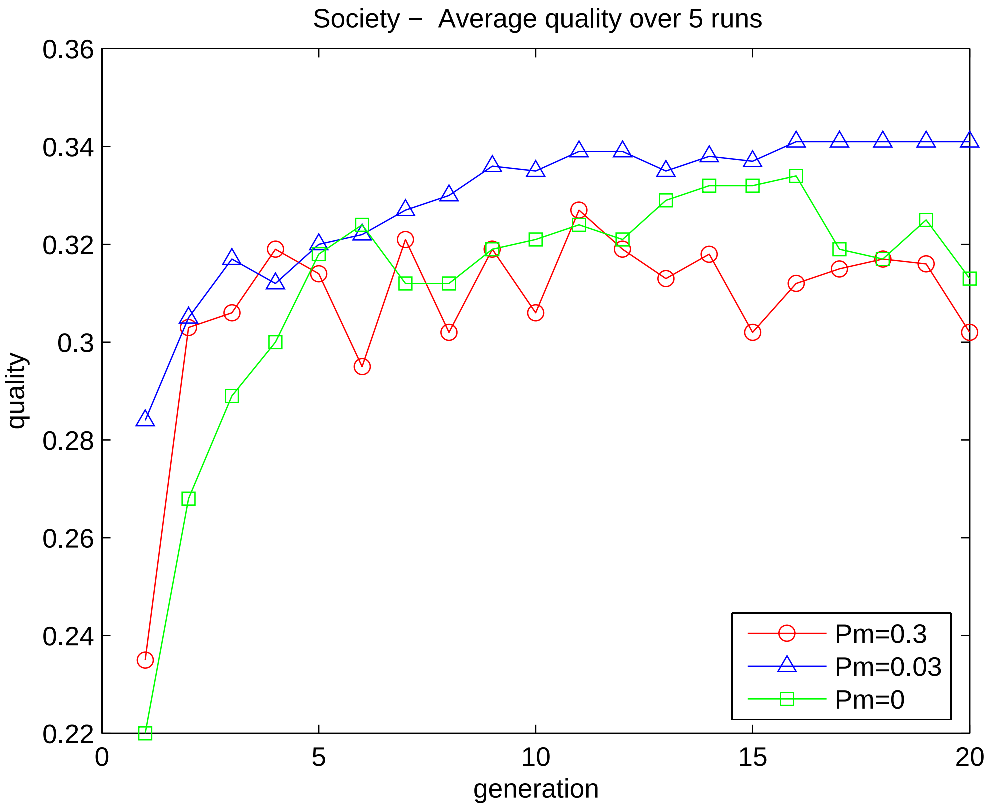

Figure 5: A test showing the average query quality over five independent runs for the topic *Society* using no mutation (Pm=0), classical mutation (Pm=0.03) and hypermutation (Pm=0.3).

## 5 CONCLUDING REMARKS

This paper presented an overview of techniques based on GAs for topical search. We have studied the effect of different mutation probabilities on the behavior of the proposed methods. As expected, we observed that higher mutations rates induce a more thorough exploration of the search space.

The techniques presented in this paper are applicable to any domain for which it is possible to generate term-based characterizations of a topic. Besides the application of resource harvesting for topical Web portals, these techniques can help build systems for a range of information services, including task-based search [11, 3], deep web search [9, 15] and support for knowledge management [10, 12]. However, we should remark that 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 for non-real time systems, where slow response times are acceptable.

There have been some previous proposals to apply GA techniques to deal with problems in the area of information retrieval. Among the existing proposals we can mention the application of GA techniques to derive better document descriptions [7] and for term weight reinforcement in query optimization [18, 2]. These proposals differ from ours in attempting to tune the weights of individual terms rather than evolving queries. In addition, while our approach is fully automatic, others require relevance feedback from the users.

As part of our future work we expect to continue testing different settings for the GA parameters (population size, crossover probability, mutation probability) as well as other selection methods such as tournament selection. We also plan to implement elitism, which will ensure the preservation of the best queries across generations. Other future directions include the application of genetic programming to evolve queries with special syntaxes [4] and the investigation of alternative fitness functions.

## REFERENCES

[1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley, 1999.

[2] Mohand Boughanem, Claude Chrisment, and Lynda Tamine. On Using Genetic Algorithms for Multimodal Relevance Optimization in Information Retrieval. *J. Am. Soc. Inf. Sci. Technol.*, 53(11):934–942, 2002.

[3] Jay Budzik, Kristian J. Hammond, and Larry Birnbaum. Information Access in Context. *Knowledge Based Systems*, 14(1–2):37–53, 2001.

[4] Tara Calishain and Rael Dornfest. *Google Hacks. 100 Industrial-Strengths Tips and Tools.* O'Reilly, 2003.

[5] Rocío L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman, and Nélida B. Brignole. Searching the Web in Context: Genetic Algorithms for Exploring Query Space. To appear in *Proceedings of the Symposium of Information Society (SSI-JAIIO), Mar del Plata, Argentina.* SADIO, 2007.

[6] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused Crawling: a New Approach to Topic-Specific Web Resource Discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640, 1999. 1999a.

[7] M. Gordon. Probabilistic and Genetic Algorithms in Document Retrieval. *Commun. ACM*, 31(10):1208–1218, 1988.

[8] John H. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor: The University of Michigan Press, 1975.

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

[10] David Leake, Ana Maguitman, Thomas Reichherzer, Alberto Cañas, Marco Carvalho, Marco Arguedas, Sofia Brenes, and Tom Eskridge. Aiding Knowledge Capture by Searching for Extensions of Knowledge Models. In *Proceedings of KCAP-2003.* ACM Press, 2003.

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

[12] Ana Maguitman, David Leake, and Thomas Reichherzer. Suggesting Novel but Related Topics: Towards Context-Based Support for Knowledge Model Extension. In *Proceedings of IUI-2005*, pages 207–214, New York, NY, USA, 2005. ACM Press.

[13] Ana Maguitman, David Leake, Thomas Reichherzer, and Filippo Menczer. Dynamic Extraction of Topic Descriptors and Discriminators: Towards Automatic Context-Based Topic Search. In *Proceedings of CIKM-2004*, Washington, DC, November 2004. ACM Press.

[14] Filippo Menczer, Gautam Pant, and Padmini Srinivasan. Topical Web Crawlers: Evaluating Adaptive Algorithms. *ACM Trans. Inter. Tech.*, 4(4):378–419, 2004.

[15] Alexandros Ntoulas, Petros Zerfos, and Junghoo Cho. Downloading Textual Hidden Web Content through Keyword Queries. In *Proceedings of JCDL-2005*, pages 100–109, New York, NY, USA, 2005. ACM Press.

[16] Bradley Rhodes and Thad Starner. The Remembrance Agent: A Continuously Running Automated Information Retrieval System. In *Proceedings of PAAM-1996*, pages 487–495, London, UK, April 1996.

[17] Barry Smyth and Paul McClave. Similarity vs. Diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning. Vancouver, Canada*, 2001.

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