Effective Information Retrieval using Genetic Algorithms based Matching Functions Adaptation

Praveen Pathak Michael Gordon Weiguo Fan

Purdue University University of Michigan

pathakp@mgmt.purdue.edu mdgordon@umich.edu wfan@umich.edu

Abstract

Knowledge intensive organizations have vast array of information contained in large document repositories. With the advent of E-commerce and corporate intranets/extranets, these repositories are expected to grow at a fast pace. This explosive growth has led to huge, fragmented, and unstructured document collections. Although it has become easier to collect and store information in document collections, it has become increasingly difficult to retrieve relevant information from these large document collections. This paper addresses the issue of improving retrieval performance (in terms of precision and recall) for retrieval from document collections.

There are three important paradigms of research in the area of information retrieval (IR): Probabilistic IR, Knowledge-based IR, and, Artificial Intelligence based techniques like neural networks and symbolic learning. Very few researchers have tried to use evolutionary algorithms like genetic algorithms (GA's). Previous attempts at using GA's have concentrated on modifying document representations or modifying query representations. This work looks at the possibility of applying GA's to adapt various matching functions. It is hoped that such an adaptation of the matching functions will lead to a better retrieval performance than that obtained by using a single matching function. An overall matching function is treated as a weighted combination of scores produced by individual matching functions. This overall score is used to rank and retrieve documents. Weights associated with individual functions are searched using Genetic Algorithm.

The idea is tested on a real document collection called the Cranfield collection. The results look very encouraging

1. Introduction

As the cost of storage devices continues to decrease there is tremendous growth in databases of all sorts (relational, graphical, and textual). Knowledge intensive organizations have vast array of information contained in large document repositories. With the advent of E-commerce and corporate intranets/extranets, these repositories are expected to grow at a fast pace. This explosive growth has led to huge, fragmented, and unstructured document collections. Although it has become easier to collect and store information in document collections, it has become increasingly difficult to retrieve relevant information from these large document collections. Various techniques have been used by researchers to address the issue of improving retrieval performance. This paper looks at how genetic algorithms (GA’s) can be used in the field of information retrieval (IR) and specifically how matching functions, used to match documents descriptions with query descriptions, can be adapted using GA's. The technique is tested on an actual document collection and the results look promising.

In the next section we present a basic architecture of an IR system. Section 3 reviews various paradigms used in IR and describes where this work fits in. Section 4 describes the details of the algorithm used in the research. It also discusses the hypotheses used, and the methodology followed. Section 5 describes some results obtained when testing the methodology on an actual document collection. Section 6 talks about possible future directions in this field of research and concludes the paper.

0-7695-0493-0/00 $10.00 (c) 2000 IEEE

2. Basic Information Retrieval System

A document based IR system typically consists of three main subsystems: document representation, representation of users' requirements (queries), and the algorithms used to match user requirements (queries) with document representations. The basic architecture is as shown in figure 1.

A document collection consists of many documents containing information about various subjects or topics of interests. Document contents are transformed into a document representation (either manually or automatically). Document representations are done in a way such that matching these with queries is easy. Another consideration in document representation is that such a representation should correctly reflect the author's intention. The primary concern in representation is how to select proper index terms. Typically representation proceeds by extracting keywords that are considered as content identifiers and organizing them into a given format.

Queries transform the user's information need into a form that correctly represents the user's underlying information requirement and is suitable for the matching process. Query formatting depends on the underlying model of retrieval used (Boolean models [Bookstein, 1985], vector space models [Salton & McGill, 1983], probabilistic models [Maron & Kuhns, 1960; Robertson, 1977], fuzzy retrieval models [Borgodna & Pasi, 1993], models based on artificial intelligence techniques [Maaeng, 1992; Evans 1993]).

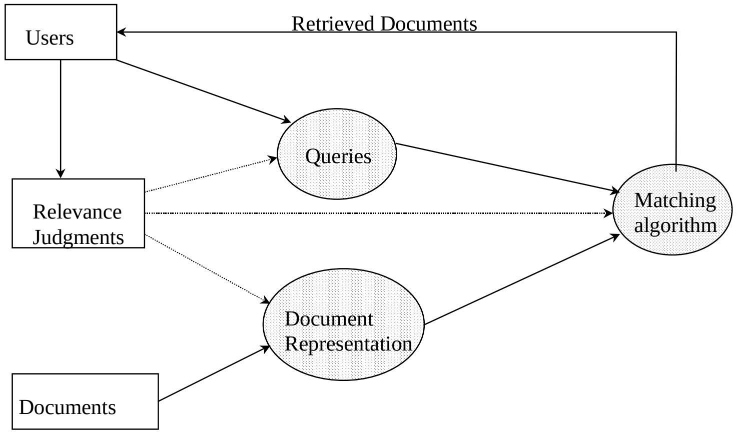

Figure 1 Information Retrieval System - Basic Architecture

A matching algorithm matches a user's requests (in terms of queries) with the document representations and retrieves documents that are most likely to be relevant to the user. A matching algorithm addresses two issues: 1. How to decide how well documents match a user's information request. Blair & Maron [1985] showed that it is very difficult for users to predict the exact words or phrases used by authors in desired documents. Hence if a document term does not match search terms then a relevant document may not be retrieved. 2. Another issue involved in matching is how to decide the order in which the documents are to be shown to the user. Typically the matching algorithms calculate a matching number for each document and retrieve the documents in the decreasing order of this number.

The user rates documents presented as either relevant or non-relevant to his/her information need. The basic problem facing any IR system is how to retrieve only the relevant documents for the user's information requirements, while not retrieving non-relevant ones. Various system performance criteria like *precision* and *recall* have been used to gauge the effectiveness of the system in meeting users' information requirements. *Recall* is the ratio of the number of relevant retrieved documents to the total number of relevant documents available in the document collection. *Precision* is defined as the ratio of the number of relevant retrieved documents to the total number of retrieved documents. Relevance feedback is typically used by the system (dotted arrows in figure 1) to improve document descriptions [Gordon, 1988], or queries [Salton & Buckley, 1990] with the expectation that the overall performance of the system will improve after such a feedback. In this paper we look at how relevance feedback can be used to improve retrieval performance by adapting the matching algorithms used.

3. IR Paradigms

This section briefly describes various research paradigms prevalent in IR and where our work fits in. At a broad level, research in IR can be categorized [Chen, 1995] into three categories: 1. Probabilistic IR, 2. Knowledge based IR, and 3. IR based on machine learning techniques.

1. *Probabilistic IR*: Probabilistic retrieval is based on estimating a probability of relevance of a document to the user for the given user query. Typically relevance feedback from a few documents is used to establish the probability of relevance for other documents in the collection [Fuhr et al., 1991; Gordon, 1988]. There are three different learning strategies used in probabilistic retrieval. Estimation of probabilities of relevance is done for a set of sample documents [Robertson & Sparck Jones, 1976], or a set of sample queries [Maron & Kuhns, 1960] and extended to all the documents or queries. Inference networks [Turtle & Croft, 1990] use a document and query network that capture probabilistic dependencies among the nodes in the network.

2. *Knowledge based IR*: This approach focuses on modeling two areas. First, it tries to model the knowledge of an expert retriever in terms of the expert's domain knowledge, that is, his or her search strategies and feedback heuristics. An example of such an approach is the Unified Medical Language System. Another area that has been modeled is the user of the system. This typically follows the way the librarian develops a client profile. Although knowledge based approaches might be effective in certain domains, it may not be applicable in all domains [Chen et al., 1991].

3. *Learning systems based IR*: This approach is based on algorithmic extraction of knowledge or identifying patterns in the data. There are three broad areas within this approach: Symbolic learning, Neural networks, and Evolution based algorithms.

   In the symbolic learning approach knowledge discovery is done typically by inductive learning by creating a hierarchical arrangement of concepts and producing IF-THEN type production rules. ID3 decision-making algorithm [Quinlan, 1986] is one such popular algorithm.

   Neural networks are connectionist learning algorithms that typically simulate the way human brain learns and remembers knowledge. In these algorithms knowledge is captured and remembered in terms of the weights on synapses, the interconnections of the neurons, and the thresholds on logic units. Belew [1989] used a neural network of authors, index terms, and documents to produce new connections between documents and index terms. Other instances of use of neural networks in IR have been documented by Doszkocs et al. [1990].

   Evolutionary algorithms are based on the Darwinian principles of natural selection. These algorithms can be further divided into: GA's, evolutionary strategies, and evolutionary programming. While evolutionary programming utilizes changes at the level of species, the evolutionary strategies exploit changes at individual behavioral level. GA's [Holland, 1975] are based on genetic operators of selection, crossover, and mutation. There are a few studies in IR literature that use GA's. Gordon [1988] presented an approach for redescribing document descriptions and subsequently adopted a similar approach to document clustering [Gordon, 1991]. Raghavan et al. [1987] have also used GA's for modifying document clustering. Yang [1993] used GA's to improve queries using relevance feedback. Chen [1995] used GA's to optimize keywords that were used to suggest relevant documents. Our work fits well in this paradigm. We will be using GA's to adapt matching functions that are used to match document descriptions with queries.

4. Algorithm Details

As stated earlier GA's have been used to modify document descriptions or queries. In this section we describe how we can use GA's to modify the matching functions used and the experimental design to test our algorithm. We use the vector space model [Salton, 1971] as the underlying model in this research. In this model, documents and queries are located in a multi-dimensional vector space. Retrieval is accomplished by searching for documents that are close to the query vector. Typically a single such matching function is used to match document vector with the query vector. Although a large number of matching functions have been tried in literature [Jones et al., 1987], no single matching function has been proved to be the best. Characteristics of retrieval environment such as the size of the database, the type of the database, and the nature of the user community affects which matching function will perform better [Jones et al., 1987]. Harman

[1986] showed that by switching between different normalized inner product measures as matching functions it is possible to get a 12% improvement in average precision. These factors also suggest the need for learning optimal matching functions. Although important, very little research has been done in adaptation of matching functions.

Bartell, Cottrell, and Belew [1998] have used numerical methods to optimize the parameters of a matching function. But they have chosen to optimize only the parameters involved in a standard inner product measure. Hence their adapted matching function is limited to variations of standard inner product measures. As an example, their adaptation leads to the use of one of the following matching functions: inner product, cosine, or pseudo-cosine. By contrast, our research looks at adaptation of various different forms of matching functions and is not restricted to a particular form of the matching function. Bartell et al. [1998] have assumed that the IR model have criteria (like ordering of documents) that are differentiable in nature. This assumption leads them to use numerical methods. This assumption of existence of differentiable criteria may not hold for discreet criteria like precision and recall. Hence numerical methods may not always be useful. Our research uses genetic algorithms that do not suffer from such a limitation in assumptions regarding the nature of the criteria used.

We treat an overall matching function as a weighted sum of the scores returned by different matching functions. Thus,

Overall matching function $(d_j,q) = \Sigma\ (wt_i * Mf_i(d_j,q))$

where i ranges from 1 to the total number of matching functions used; MF_1, MF_2 etc. are the scores produced by individual matching functions; and wt_1, wt_2, etc. are the weights associated with these scores. The (d_j,q) signifies that this matching function is utilized to calculate scores for the document d_j (j varying from 1 to the total number of documents) for the given query 'q'. The weights wt_1, wt_2, etc. range from 0.0 to 1.0. A higher weight signifies that the associated matching function is more important than that which is associated with lower weights. Thus a matching function with a weight of 0.6 is doubly as important as that with a weight of 0.3. A matching function with an associated weight of 0.0 is completely insignificant. It is hypothesized that by proper combination of these different weights (a weighted combination of scores produced by individual matching function) it should be possible to achieve retrieval results that are superior compared to that produced by any single matching function. Hence the task now reduces to finding appropriate weights to be used for each matching function. This essentially is searching a multidimensional space for optimum combination of weights.

GA's are robust in searching a multidimensional space to find optimal or near optimal solutions [Goldberg, 1989; Holland, 1975]. This motivated the use of GA in this research to search for such an optimal or near optimal combination of weights.

Fitness function in GA is the function that is optimized using the genetic process. Choosing an appropriate fitness function is very important. In IR *recall* and *precision* (defined earlier) are the two most widely used measures of retrieval performance. GA's typically require a single valued measure to evaluate fitness of an individual in the population. van Rijsbergen [1979] suggested a single point measure which combines precision and recall measures. It is:

$$E = 1 - \frac{1}{\left[\frac{\alpha}{P} + \frac{(1-\alpha)}{R}\right]} \qquad \ldots\ (1)$$

where $\propto$ is a parameter that is used to express the degree of user preference for precision (P) or recall (R) component. A higher value of $\propto$ characterizes a user with less preference for recall, while a lower value of $\propto$ characterizes one with a less preference for precision. We decided to use (1-E) as our fitness function so that higher values of our fitness function correspond with better performance.

<u>Hypotheses</u>: The effectiveness of the GA based solution was tested using two hypotheses, and also by graphical analysis. Performance was measured in terms of average precision. The hypotheses tested were:

<u>Hypothesis 1</u>: GA based matching function adaptation improves the average retrieval performance in the final generation as compared to the performance achieved by the individual matching function without genetic modifications.

H_0: Average overall performance$_{\text{final generation}}$ - Average performance$_{\text{individual matching function}} \leq 0$

(for every individual matching function)

H1: Average overall performance$_{\text{final generation}}$ - Average performance$_{\text{individual matching function}} > 0$

(for every individual matching function)

This hypothesis tests for improvement in performance when an overall matching function is used for retrieval instead of individual matching functions. This is done by comparing the average overall performance obtained in the final generation (last generation of the GA process) with that produced by any individual matching function used without any adaptation or modification. Average performance here refers to the averaging of performance over all queries issued to the system. Average performance$_{\text{individual matching function}}$ indicates the average performance obtained when only an individual matching function without any adaptation is used in the retrieval process. The weights associated with an overall matching function (weighted sum of scores of individual matching functions, as described earlier) are adapted using GA's. Over generations the retrieval performance is expected to improve. At the final generation we test for significant performance improvement over the performance achieved by any individual matching function. If the null hypothesis is false then one can establish the importance of using a weighted matching function instead of an individual matching function, given the weights are adapted using GA's as described.

Hypothesis 2: GA based matching function adaptation improves the average retrieval performance in the final generation as compared to the values achieved at initial generation.

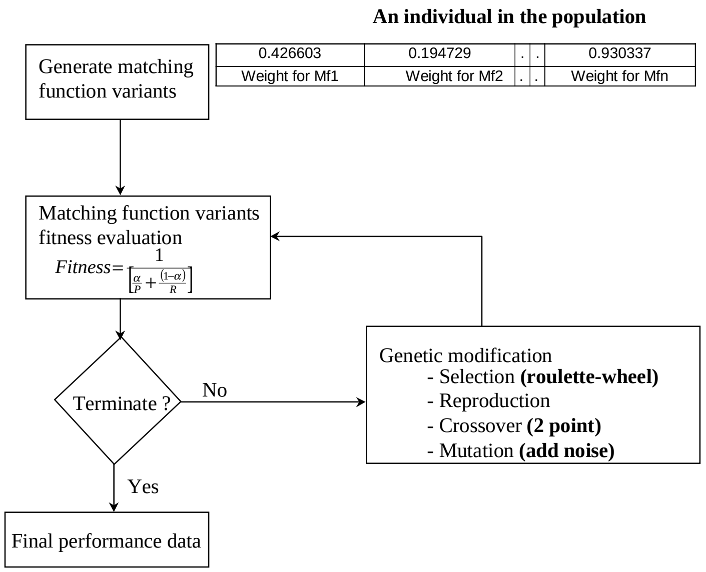

Figure 2 Genetic Procedure Followed

H_0: Average overall performance$_{\text{final generation}}$ - Average overall performance$_{\text{initial generation}}$ ≤ 0

H1: Average overall performance$_{\text{final generation}}$ - Average overall performance$_{\text{initial generation}}$ > 0

This hypothesis tests for the usefulness of the GA process in adapting the weights associated with individual matching functions. Here Average overall performance$_{\text{initial generation}}$ indicates the average performance of the matching function variants (to be discussed) in the initial generation i.e. at the beginning of the process. Similarly, Average overall performance$_{\text{final generation}}$ indicates the average performance of the matching function variants at the final generation. If the null hypothesis is false, then that establishes the fact that a GA based adaptation of weights improves retrieval performance over successive generations.

The Genetic Process: The following process was followed to implement GA (refer to figure 2).

Generate matching function variants: For each individual matching function we assigned a randomly chosen weight (in the range 0.0 to 1.0. The overall matching function is a weighted combination of the individual function scores (individual matching function scores are normalized to be in the range of 0 to 1). Weights are encoded using the actual real numbers between 0.0 and 1.0 (inclusive). The initial population consisted of 50 (population size) such randomly chosen individuals.

Matching function variants fitness evaluation: For each individual in the population an overall matching score is calculated for each document and documents in the collection are arranged in the decreasing order of this score. Based on the parameter for document cut-off value (DCV is number of documents the user is willing to see) the top DCV number of documents are retrieved. Based on the relevance judgments for this set of documents, precision and recall are calculated. These values are used to calculate fitness of the individual.

Genetic Modification: In this step, genetic operators are applied to the individuals in the previous generation to generate the next generation of individuals. It involves four stages.

1. *Selection and reproduction*: All individuals in the previous generation were made available for reproduction in the next generation. The roulette-wheel reproduction process [Goldberg, 1989] was used to select individuals for reproduction.
2. *Crossover*: A two-point crossover was followed (exchanging information between two randomly selected points on the individual string). A parameter 'cross-over rate' determined the number of individuals that actually mate.
3. *Mutation*: Mutation was accomplished by introducing gaussian noise. This was accomplished by adding to the original number, a number generated from a normal population with a mean of 0.0 and a standard deviation of 0.2. If the resultant was capped to be between 0.0 and 1.0 (in case the result overshot in either direction it was capped at the boundaries).
4. *Process termination*: The process of genetic modification was terminated after a preset number of generations (75) or after convergence when no improvement in performance was observed for 10 generations.

5. Results

We tested the algorithm using two document databases: A simulated document collection and Cranfield document collection. The simulated document collection was primarily used to understand the characteristics and behavior of the retrieval process and the genetic algorithm. Details of the simulation and the results on the simulated collection are available elsewhere [Pathak, 1998]. Another document collection that we used was the Cranfield database. It is a widely used database by the IR community. It contains documents in the natural language format regarding the experiments in the field of aeronautical engineering. It has 225 user queries with relevance judgements for each document in the collection for each of these queries. These documents were first parsed to extract the token and document frequencies of different tokens. This information was used in the matching process. A standard tf*idf weight [Salton et al., 1983] was used for each token.

We used four different matching functions in our experiments. These were: Cosine, Jaccard, Dice, and Overlap [Rijsbergen, 1979]. They were chosen as they are

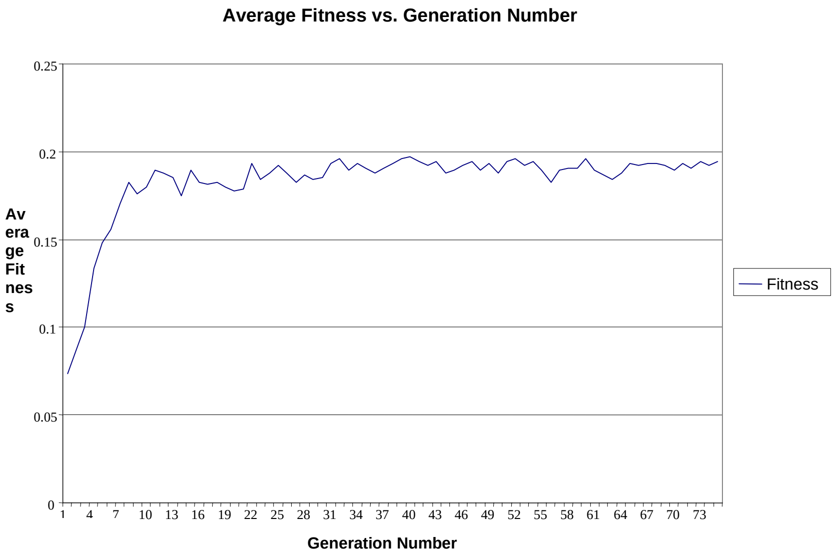

Figure 3 Average Fitness vs. Generation Number

the most commonly used functions.

Experiments were run for 75 generations with 50 individuals in each generation. ∝ was set to 1 and the document cutoff was set to 15 documents. Crossover rate was 0.6, while mutation was set to 0.1. These parameters were selected based on experiments that used various combinations of these parameters. This particular combination worked best with a set of training queries. The genetic process was tested for 41 queries form the Cranfiled database.

Hypothesis 1 was tested by comparing average precision values, across all queries, after the last generation with the maximum average precision values obtained by any individual matching function without any modification. A paired T-test was used. T-statistic was 3.35 and hence the null hypothesis was rejected at 0.05 significance, indicating that performance after genetic adaptation is statistically significantly different from the performance achieved without any adaptation of the matching functions.

Hypothesis 2 was tested by comparing, across the 41 queries, average fitness obtained in the last generation with the average fitness obtained in the first generation. T-statistic was 3.52 and hence the null hypothesis was rejected at 0.05 significance, indicating that average fitness of individuals does increase over generations.

Graphical analysis was done by plotting average fitness obtained in a generation against the generation number (figure 3). This graph, plotted for a typical query, gives an idea about how adaptation over generations improves fitness, and thereby precision and recall. It can be seen that the fitness increases rapidly initially suggesting a fast convergence.

6. Discussion & Future Directions

- This work has introduced a new perspective to the area of matching function adaptation in IR. Prior research in IR has focussed primarily on document and query adaptation. We have shown here that genetic adaptation of matching functions can lead to improved retrieval performance. More work, however, needs to be done in this area. We have included only four matching functions in these experiments. We need to include more matching functions to make the results richer.
- Our algorithm seems to work well with the simulated document collection and also the Cranfield document collection. It is also necessary to test this algorithm on different document collections to see how it performs with scaling both in size of the database and in the features available.
- It is to be noted that we do not assume any specific underlying model of retrieval (although the experiments were run using the vector-space model). Our model requires only the retrieval value associated with a document. Hence our approach is generalizable to a variety of retrieval techniques.
- Our selection of fitness function lets us handle various user preferences. By appropriate setting of the parameter value '∝' we can fine-tune our approach for users varying from those who need high recall to those who need high precision.
- Current research in IR focuses on adaptation of an individual subsystem (document, or query). In future it should be possible to combine the ideas in this work with previous GA based work on document adaptation and query adaptation. It is to be noted that document, query, and matching function adaptation approaches are complementary to each other. We do not necessarily have to choose any one of these approaches over the other. All three can coexist in an IR system. From a practical perspective, matching function adaptation and query adaptation can be done during the user's query session. Document adaptation involves changing document descriptions for thousands of documents, which is a time consuming process. Hence document descriptions can be adapted over a longer time frame.
- Our research combines various matching functions available by combining them. Another promising area could be to see, if rather than using a set of existing matching functions could we evolve completely novel matching functions. This evolution of novel matching functions could be done using genetic programming type of techniques by appropriately combining various features (e.g. token frequency, document frequency, paragraph lengths, availability of tokens in the titles etc.) utilized in retrieval.

7. Conclusion

In this paper we described a method of utilizing genetic algorithms in the field of information retrieval and specifically how the GA's can be used to adapt the matching functions used. This algorithm was tested on the Cranfield document collection and the results look promising. We see the need to pursue more research in this promising area.

References

[1] Bartell, B.T., Cottrell, G.W., & Belew, R.K., "Optimizing similarity using multi-query relevance feedback", Journal of the American Society for Information Science, 49(8), 1998, pp: 742-761

[2] Bordogna, G & Pasi, G. “A fuzzy linguistic approach generalizing Boolean information retrieval: a model and its evaluation”, Journal of the American Society for Information Science, 44(2), 1993, pp: 70-82

[3] Belew, R., "Adaptive information retrieval", Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval, 1989, pp: 11-20

[4] Blair, D.C. & Maron, M.E. “An evaluation of retrieval effectiveness for a full text document-retrieval system”, Communications of the ACM, 28(3), 1985, pp: 289-299

[5] Bookstein, A. “Probability and fuzzy-set applications to information retrieval”, Annual Review of Information Science and Technology, 20, 1985, pp: 117-151

[6] Chen, H., & Dhar, V., "Cognitive process as a basis for intelligent retrieval systems design", Information Processing and Management, 27, 1991, pp: 405-432

[7] Chen, H., “Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms", Journal of the American Society for Information Science, 46(3), 1995, pp: 194-216

[8] Doszkocs, T., Reggia, J., & Lin, X., "Connectionist models and information retrieval", Annual Review of Information Science and Technology, 25, 1990, pp: 209-260

[9] Evans, D., “TREC experiments of the CLARIT project”, in The First Text Retrieval Conference (TREC1), 1993

[10] Fuhr, N. & Buckley, C, "A Probabilistic Learning Approach for Document Indexing", ACM Transactions on Information Systems, 9, 1991, pp: 223-248

[11] Goldberg, D.E. “Genetic Algorithms in Search, Optimization and Machine Learning”, Reading M.A.: Addison-Wesley, 1989

[12] Gordon, M.D. “Probabilistic and genetic algorithms for document retrieval”, Communications of the ACM, 31(10), 1988, pp: 1208-1218

[13] Gordon, M.D. “User-based document clustering by redescribing subject descriptions with a genetic algorithm”, Journal of the American Society for Information Science, 42, 1991, pp: 311-322

[14] Harman, D., "An experimental study of factors important in document ranking", in Proceedings of the ACM SIGIR, 1986, pp: 186-193

[15] Holland, J.H. “Adaptation in Natural and Artificial Systems”, Ann Arbor: The University of Michigan Press, 1975

[16] Jones, W.P., & Furnas, G.W., "Pictures of relevance: A geometric analysis of similarity measures", Journal of the American Society for Information Science, 38, pp: 420-442

[17] Maron, M., & Kuhns, J., "On relevance, probabilistic indexing and information retrieval", Journal of the ACM, 7, 1960, pp: 216-243

[18] Myaeng, S.H., “Using conceptual graphs for information retrieval: a framework for adequate representation and flexible inferencing”, Proceedings of the Symposium on Document Analysis and Information Retrieval, 1992, pp: 102-116

[19] Pathak, P., "A simulation model of document information retrieval system with relevance feedback", Proceedings of the America Conference of the Association for Information Systems, 1998, pp: 194-196

[20] Pathak, P., "Relevance Feedback in Information Retrieval Using Genetic Algorithms: A Test on Simulated Documents", Proceedings of the Eighth Annual Workshop on Information Technologies and Systems, WITS'98, 1998, pp: 65-74

[21] Quinlan, J., "Induction of decision trees", Machine Learning, 1, 1986, 1993, pp: 81-106

[22] Raghavan, V., & Agarwal., B., "Optimal determination of user-oriented clusters: An application for the reproductive plan", Proceedings of the Second International Conference on Genetic Algorithms and their Applications, Hillsdale, NJ: Lawrence Erlbaum Associates, 1987, pp: 241-246

[23] Robertson, S. & Sparck Jones, K., "Relevance weighting of search terms", Journal of the American Society for Information Sciences, 27, 1976, pp: 129-146

[24] Robertson, S.E., “The probabilistic character of relevance”, Information Processing & Management, 13, 1977, pp: 247-251

[25] Salton, G., "The SMART Retrieval System: experiments in automatic document processing", New Jersey: Prentice Hall, (1971)

[26] Salton, G. & McGill, M., "Introduction to Modern Information Retrieval", New York: McGraw-Hill, (1983)

[27] Salton, G. & Buckley C. “Improving retrieval performance by relevance feedback”, Journal of the American Society for Information Science, 41(4), 1990, pp: 288-297

[28] Turtle, H. & Croft, W., "Inference networks for document retrieval", Proceedings of the 13th Annual International ACM/SIGIR Conference in Research and Development in Information Retrieval, ACM Press, 1990, pp: 1-24

[29] van Rijsbergen, C.J. “Information Retrieval”, Butterworth, 1979

[30] Yang, J., & Korfhage, R.R., “Query Optimization in Information Retrieval Using Genetic Algorithms”, Proceedings of the fifth International Conference on Genetic Algorithms, 1993, pp: 603-613