[Project MINERVAEUROPE] Project MINERVAEUROPE: Ministerial Network for Valorising Activities in digitalisation - <http://www.minervaeurope.org>

[Rainie, 2006] Rainie, L. Life Online: Teens and technology and the world to come (Annual conference of Public Library Association, Boston, March 23, 2006) - <http://www.pewinternet.org/ppt/Teens and technology.pdf>

Authors' Information

Detelin Luchev - Ethnographic Institute with Museum, BAS; Bulgaria, 1000 Sofia, Moskovska str. 6A; e-mail: luchev_detelin@abv.bg

APPLYING GENETIC ALGORITHM IN QUERY IMPROVEMENT PROBLEM

Abdelmgeid A. Aly

Abstract: This paper presents an adaptive method using genetic algorithm to modify user's queries, based on relevance judgments. This algorithm was adapted for the three well-known documents collections (CISI, NLP and CACM). The method is shown to be applicable to large text collections, where more relevant documents are presented to users in the genetic modification. The algorithm shows the effects of applying GA to improve the effectiveness of queries in IR systems. Further studies are planned to adjust the system parameters to improve its effectiveness. The goal is to retrieve most relevant documents with less number of non-relevant documents with respect to user's query in information retrieval system using genetic algorithm.

1. Introduction

Several researchers have used the GA in IR and their results seem to indicate that this algorithm could be efficient. In this vein, the main directions concern modifying the document indexing [1] and [2] and the clustering problem [3].

It is not surprising therefore that there have recently appeared many applications of GAs in information retrieval. Most of them use the vector space model, which also seems to be one of the most widely, used models in general [4]. These applications implement learning of the terms and/ or weights of the queries.

Information Retrieval systems are used to retrieve documents that depend on or relevant to the user input query. The growth in the number of documents made it necessary to use the best knowledge or methods in retrieving the most relevant documents to the user query.

Information Retrieval systems deal with data bases which are composed of information items documents that may consist of textual, pictorial or vocal information. Such systems process user queries trying to allow the user to access the relevant information in an appropriate time interval. The art of searching will be in the databases or hypertext networked databases such as internet or intranet for text, sound, images or data, [4]. Thus an information system has its heart a collection of data about reality [5].

Most of the information retrieval systems are based on the Boolean queries where the query terms are joined by the logical operators AND and OR. The similarity between a query and documents is measured by different retrieval strategies that are based on the more frequent terms found in both the document and the query. The more relevant document is deemed to be the query request. The most frequently used measures of retrieval effectiveness are **precision**, *the percentage of the retrieval documents that are relevant* and **recall**, *the percentage of the relevant documents that are retrieved*.

Information retrieval is concerned with collection and organization of texts, responding to the requests of users for the information seeking texts, retrieving the most relevant documents from a collection of documents; and with retrieving some of non-relevant as possible. Information retrieval is involved in:

- Representation,
- Storage,
- Searching,

- Finding documents or texts which are relevant to some requirements for the information desired by a user.

The input of GAs [6] is a population of individuals called chromosomes, which represent the possible solutions to the problem. These are either generated at random or, if one has some knowledge can be used to create part of the initial set of potential solutions [7]. These individuals change (evolve) over successive iterations called generations, via processes of selection, crossover, and mutation. The iterations end when the system no longer improves or when a pre-set maximum number of generations has been reached. The output of the GA will be the best individual of the end population, or a combination of the best chromosomes.

To solve each problem, one has to provide a fitness function *f*. Its choice is crucial for the proper functioning of the algorithm. Given a chromosome, the fitness function must return a numeric value that represents its utility. This score will be used in the process of selection of the parents, so that the fittest individuals will have a greater chance of being selected.

This paper presents a GA based on Cosine fitness function and Classical IR in query optimization problems, and some applications of GA in information retrieval system. This algorithm has been applied on three well-known test collections (CISI, CACM and NPL). The goal is to retrieve most relevant documents with less number of non-relevant documents with respect to user query in information retrieval system using genetic algorithm.

2. Antecedents

2.1. Information retrieval models

Several retrieval models have been studied and developed in the IR area, we analyze some of these models, which are:

Boolean model. In the Boolean retrieval model, the indexer module performs a binary indexing in the sense that a term in a document representation is either significant (appears at least once in it) or not. User queries in this model are expressed using a query language that is based on these terms and allows combinations of simple user requirements with the logical operators AND, OR and NOT. The result obtained from the processing of a query is a set of documents that totally match with it, i.e., only two Possibilities are considered for each document: to be or not to be relevant for the user's needs, represented by the user query [8][9].

Vector space model. In this model, a document is viewed as a vector in an n-dimensional document space (where n is the number of distinguishing terms

used to describe contents of the documents in a collection) and each term represents one dimension in the document space. A query is also treated in the same way and constructed from the terms and weights provided in the user request. Document retrieval is based on the measurement of the similarity between the query and the documents. This means that documents with a higher similarity to the query are judged to be more relevant to it and should be retrieved by the IRS in a higher position in the list of retrieved documents. In This method, the retrieved documents can be orderly presented to the user with respect to their relevance to the query [8].

Probabilistic model. This model tries to use the probability theory to build the search function and its operation mode. The information used to compose the search function is obtained from the distribution of the index terms throughout the collection of documents or a subset of it. This information is used to set the values of some parameters of the search function, which is composed of a set of weights associated to the index terms [10][11].

2.2. Evaluation of information retrieval systems

There are several ways to measure the quality of an IRS, such as the system of efficiency and effectiveness, and several subjective aspects related to the user satisfaction. Traditionally, the retrieval effectiveness (usually based on the document relevance with respect to the user's needs) is the most considered. There are different criteria to measure this aspect, with the precision and the recall being the most used.

Precision (P) is the rate between the relevant documents retrieved by the IRS in response to a query and the total number of documents retrieved, whilst Recall (R) is the rate between the number of relevant documents retrieved and the total number of relevant documents to the query existing in the data base [9]. The mathematical expression of each of them is showed as follows:

$$P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \quad , \quad R = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d} \longrightarrow \qquad (1)$$

with $r_d \in \{0,1\}$ being the relevance of document d for the user and $f_d \in \{0,1\}$ being the retrieval of document d in the processing of the current query. Notice that both measures are defined in [0,1], being the optimal value.

3. Query Definition

This is the most extended group of applications of GAs in IR. Every proposal in this group uses GAs either as a relevance feedback technique or as an Inductive Query By Example (IQBE) algorithm.

The basis of relevance feedback lies in the fact that either users normally formulate queries composed of terms, which do not match the terms (used to index the relevant documents to their needs) or they do not provide the appropriate weights for the query terms. The operation mode involving and modifying the previous query (adding and removing terms or changing the weights of the existing query terms) which taking into account the relevance judgments of the documents retrieved by it, constitutes a good way to solve the latter two problems and to improve the precision, and especially the recall, of the previous query [9].

IQBE was proposed in [12] as ‘‘a process in which searchers provide sample documents (examples) and the algorithms induce (or learn) the key concepts in order to find other relevant documents’’. This method is a process for assisting the users in the query formulation process performed by machine learning methods. It works by taking a set of relevant (and optionally, non-relevant documents) provided by a user and applying an off-line learning process to automatically generate a query describing the user’s needs.

Smith and Smith [13] propose a GA for learning queries for Boolean IRSs. Although they introduce it as a relevance feedback algorithm, the experimentation is actually closer to the IQBE framework. The algorithm components are described as follows:

- The Boolean queries are encoded in expression trees, whose terminal nodes are query terms and whose inner nodes are the Boolean operators AND, OR and NOT.
- Each generation is based on selecting two parents, with the best fitted having a larger chance to be chosen, and generating two offspring from them. Both offspring are added to the current population which increments its size in this way.
- The usual GA crossover is considered [14]. No mutation operator is applied.
- The initial population is generated by randomly selecting the terms included in the set of relevant documents provided by the user, having those present in more documents a higher probability of being selected.
- The fitness function gives a composite retrieval evaluation encompassing the two main retrieval parameters (precision and recall).

Yang and Korfaghe [15] propose a similar GA to that of Robertson and Willet’s [16]. They use a real coding with the two-point crossover and random mutation operators (besides, crossover and mutation probabilities are changed throughout the GA run). The selection is based on a classic generational scheme where the chromosomes with a fitness value below the average of the population are eliminated, and the reproduction is performed by Baker’s mechanism.

4. System Framework

4.1. Building an IR System

The proposed system is based on Vector Space Model (VSM) in which both documents and queries are represented as vectors. Firstly, to determine documents terms, the following procedure is used:

- Extraction of all the words from each document.
- Elimination of the stop-words from a stop-word list generated with the frequency dictionary of Kucera and Francis [17].
- Stemming the remaining words using the porter stemmer, which is the most commonly used stemmer in English [4][18].

After using this procedure, the final number of terms was 6385 for the CISI collection, 7126 for CACM and 7772 for NPL. After determining the terms that described all documents of the collection, the weights were assigned by using the formula proposed by Salton and Buckley [19]:

$$a_{ij} = \frac{\left(0.5 + 0.5\frac{tf_{ij}}{\max tf}\right) \times \log\frac{N}{n_i}}{\sqrt{\left(0.5 + 0.5\frac{tf_{ij}}{\max tf}\right)^2 \times \left(\log\frac{N}{n_i}\right)^2}} \longrightarrow \tag{2}$$

Where a_{ij} is the weight assigned to the term t_j in document D_i, tf_{ij} is the number of times that term t_j appears in document D_i, n_j is the number of documents indexed by the term t_j and finally, N is the total number of documents in the database. Finally, the vectors are normalize by dividing them by their Euclidean norm. This is according to the study of Noreault *et al.*[17], of the best similarity measures which make angle comparisons between vectors. A similar procedure is carried out with the collection of queries, thereby obtaining the normalized query vectors. Then, the following steps are applied:

- For each collection, each query is compared with all the documents, using the cosine similarity measure. This yields a list giving the similarities of each query with all documents of the collection.
- This list is ranked in decreasing order of similarity degree.
- Make a training data consists of the top 15 document of the list with a corresponding query.
- Automatically, the keywords (terms) are retrieved from the training data and the terms which are used to form a binary query vector.
- Adapt the query vector using the genetic approach.

4.2 The Genetic Approach

Once significant keywords are extracted from training data (relevant and irrelevant documents) including weights are assigned to the keywords. The binary weights of the keywords are formed as a query vector, and the adapting of the query vector as a chromosome. Then, the GA is applied to get an optimal or near optimal query vector, and the result of GA approach is compared with the classical IRS without using GA. Ga's are characterized by 5 basic components as follows:

- Chromosome representation for the feasible solutions to the optimization problem.
- Initial population of the feasible solutions.
- A fitness function that evaluates each solution.
- Genetic operators that generate a new population from the existing population.
- Control parameters such as population size, probability of genetic operators, number of generation.

4.2.1. Representation of the chromosomes

These chromosomes use a binary representation, and are converted to a real representation by using a random function. We will have the same number of genes (components) as the query and the feedback documents have terms with non-zero weights. The set of terms contained in these documents and the query is calculated. The size of the chromosomes will be equal to the number of terms of that set, we get the query vector as a binary representation and applying the random function to modify the terms weights to real representation. Our GA approach receives an initial population chromosomes corresponding to the top 15 documents retrieved from classical IR with respect to that query.

4.2.2. Fitness Function

Fitness function is a performance measure or reward function, which evaluates how each solution, is good. In our work, we use the cosine similarity as fitness function with equation:

$$\frac{\sum_{i=1}^{t} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{t} x_i^{2} \cdot \sum_{i=1}^{t} y_i^{2}}} \longrightarrow \tag{3}$$

4.2.3. Selection

As the selection mechanism, the GA uses "simple random sampling" [20][6]. This consists of constructing a roulette with the same number of slots as there are individuals in the population, and in which the size of each slot is directly related to the individual's fitness value. Hence, the best chromosomes will on average achieve more copies, and the worst fewer copies. Also, uses the "elitism" strategy [21], as a complement to the selection mechanism. If after generating the new population, the best chromosome of the preceding generation is by chance absent, the worst individual of the new population is withdrawn and replaced by that chromosome.

4.2.4. Operators

In our GA approaches, we use two GA operators to produce offspring chromosomes, which are:

Crossover is the genetic operator that mixes two chromosomes together to form new offspring. Crossover occurs only with crossover probability P_c. Chromosomes are not subjected to crossover remain unmodified. The intuition behind crossover is exploration of a new solutions and exploitation of old solutions. GAs construct a better solution by mixture good characteristic of chromosome together. Higher fitness chromosome has an opportunity to be selected more than lower ones, so good solution always alive to the next generation. We use a single point crossover, exchanges the weights of sub-vector between two chromosomes, which are candidate for this process.

Mutation is the second operator uses in our GA systems. Mutation involves the modification of the gene values of a solution with some probability P_m. In accordance with changing some bit values of chromosomes give the different breeds. Chromosome may be better or poorer than old chromosome. If they are poorer than old chromosome they are eliminated in selection step. The objective of mutation is restoring lost and exploring variety of data.

5. Experiments and Results

To perform the trial, it was first necessary to generate test databases. We created these from three of the best known test collections: the CISI collection (1460 documents on information science), the CACM collection (3204 documents on Communications), and finally the NPL collection (11,429 documents on electronic engineering). One of the principal reasons for choosing more than one test collection is to emphasize and generalize our results in all alternative test documents collections. The experiments are applied on 100 queries chosen according to each query which does not retrieve 15 relevant documents for our IR system. From our experimental observation, the best values for this test documents collections at crossover probability P_c = 0.8 and mutation rate is P_m= 0.7 for GA. The following are the results of applying GA for 100 generations.

5.1. The CISI Documents Collection

The results for the GA are shown in table (1) and figure 1, using non-interpolated average Recall – Precision relationship. From this table and the corresponding figure we notice that GA gives a high improvement than Classical IR system with 11.9%. Also, the average number of terms of query vector before applying GA is 509.61 terms; these terms are reduced after applying GA to 358.84 terms as average number.

Table (1): Shows the experimental results of applying GA on CISI Collection

Average nine-point Recall Precision for 100 query in CISI Collection			
Recall	Precision		GA Improvement %
	Classic IR	GA	
0.1	0.679345	0.877	29.09493703
0.2	0.557805	0.658205	17.99912156
0.3	0.461991	0.584501	26.5178326
0.4	0.400701	0.444153	10.8439959
0.5	0.349373	0.403625	15.52838943
0.6	0.303939	0.310678	2.217221219
0.7	0.25167	0.264587	5.132514801
0.8	0.198868	0.192231	-3.337389625
0.9	0.149076	0.153811	3.176232257
Average	0.37253	0.432088	11.90809502

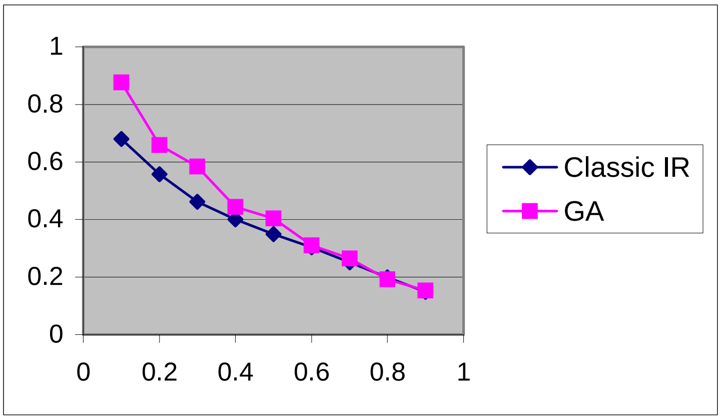

Figure 1. Represents the relationship between average recall-precision for 100 queries on CISI

5.2. The NPL Documents Collection

The results for this experiment are shown in table (2) and figure 2, using non-interpolated average Recall-Precision relationship. From this table and the corresponding figure, we find that the GA gives a higher improvement than classic IR system with 11.5% as average values. Also, the average number of terms of query vector before applying GA is 134.14 terms; these terms are reduced after applying GA to 16.8 terms.

Table (2): Shows the experimental results of applying GA on NPL Collection

Average nine-point Recall Precision for 100 query in NPL Collection			
Recall	Precision		GA Improvement %
	Classic IR	GA	
0.1	0.732924	0.802594	9.505760488
0.2	0.503371	0.550861	9.43439332
0.3	0.435145	0.47663	9.533603741
0.4	0.340466	0.385739	13.29736303
0.5	0.313329	0.363884	16.13479761
0.6	0.239987	0.274637	14.43828207
0.7	0.215388	0.238706	10.82604416
0.8	0.174277	0.193571	11.07088141
0.9	0.145552	0.159113	9.316945147
Average	0.344493	0.382859	11.50645233

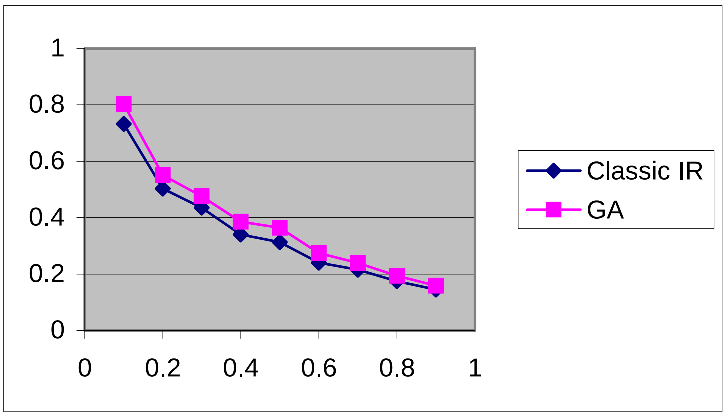

Figure 2. Represents the relationship between average recall-precision for 100 queries on NPL

5.3. The CACM Documents Collection

The results for this experiment are shown in table (3) and figure 3, using non-interpolated average Recall–Precision relationship. From this table and the corresponding figure, we notice that GA gives a higher improvement than that with classic IR system 5.13%, as average values. Also, the average number of terms of query vector before applying GA is 160.7 terms; these terms are reduced after applying GA to 16.83 terms.

Table (3): Shows the experimental results of applying GA on CACM Collection

Average nine-point Recall Precision for 100 query in CACM Collection			
Recall	Precision		GA Improvement %
	Classic IR	GA	
0.1	0.722666	0.788457	9.103929063
0.2	0.416461	0.473779	13.76311347
0.3	0.369358	0.419403	13.54918534
0.4	0.246728	0.273892	11.00969489
0.5	0.212679	0.224543	5.578359876
0.6	0.158008	0.162339	2.741000456
0.7	0.142905	0.139693	-2.247647038
0.8	0.107276	0.104937	-2.180357209
0.9	0.089651	0.085076	-5.103122107
Average	0.27397	0.296902	5.134906305

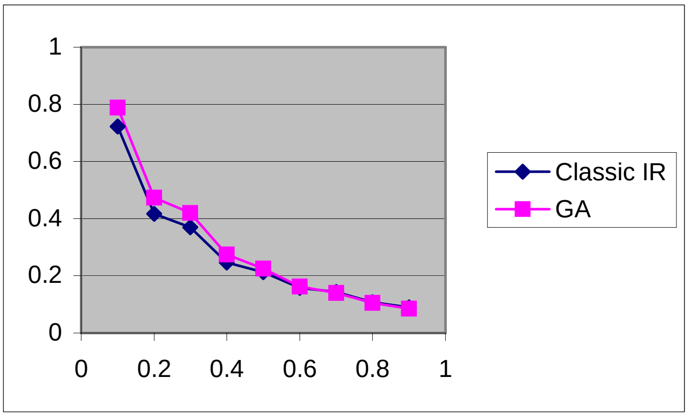

Figure 2. Represents the relationship between average recall-precision for 100 queries on CACM

6. Conclusion

From the previous results, it can be concluded that our GA approach gives more sophisticated results than classical IR system in our test collections. Also, the average number of terms is reduced after applying GA. The experiments developed use three of the relative document collections (CACM, CISI and NPL), and compare the results of two variant systems (Classical IR and GA). The latter algorithm achieves the best performance and it obtains better precision than the first approach. Our GA approach used to adapt the weights of query terms is to get high precision results.

References

[1] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor (1975).

[2] K. A. DeJong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", Ph.D. Thesis, University of Michigan (1975).

[3] V. V. Raghavan and B. Agrwal. "Optimal determination of user – oriented clusters: An application for the reproductive plan". In *Proceedings of the second conference on genetic algorithms and their applications*, Hillsdale, NJ (pp. 241-246), 1987.

[4] Baeza-Yates, R. and Ribeiro-Neto, B., *Modern Information Retrieval*, Adisson, 1999.

[5] R. Korfhage Robert. *Information storage and Retrieval*. John Wiley & Sons, Inc. 1997.

[6] Goldberg, D. E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA. (1989).

[7] Z. Michalewicz. *Genetic algorithms + data structures= evolution programs*. Berlin: Springer- verlag, 1995.

[8] Salton, G. and McGill, M. H., *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[9] Van Rijsbergen, C. J., *Information Retrieval*, second ed., Butterworth, 1979.

[10] A. Bookstein. "Outline of a general probabilistic retrieval model", *Journal of Documentation* 39 (2) 63–72 (1983).

[11] N. Fuhr. "Probabilistic models in information retrieval", *Computer Journal* 35 (3) 243–255 (1992).

[12] H. Chen et al., "A machine learning approach to inductive query by examples: an experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing", *Journal of the American Society for Information Science* 49 (8) 693–705 (1998).

[13] M.P. Smith, M. Smith. "The use of genetic programming to build Boolean queries for text retrieval through relevance feedback", *Journal of Information Science* 23 (6) 423–431 (1997).

[14] J. Koza. "Genetic Programming". *On the Programming of Computers by means of Natural Selection*, The MIT Press (1992).

[15] J. Yang and R. Korfhage. "Query modifications using genetic algorithms in vector space models", *International Journal of Expert Systems* 7 (2) 165–191 (1994).

[16] A.M. Robertson and P. Willet. "Generation of equifrequent groups of words using a genetic algorithm", *Journal of Documentation* 50 (3) 213–232 (1994).

[17] T. Noreault, M. McGill and M. B. Koll. "A performance evaluation of similarity measures, document term weighting schemes and representation in a Boolean environment". Information retrieval research. London: Butterworths (1981).

[18] M. F. Porter. "An algorithm for suffix stripping. Program", 14(3), 130–137 (1980).

[19] G. Salton and C. Buckley. "Improving retrieval performance by relevance feedback". *Journal of the American Society for Information Science*, 41(4), 288–297 (1990).

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

[21] D. C. Blair. "Language and Representation in information Retrieval". Amsterdam: Elsevier, 1990.

Authors' Information

A. A. Aly – Dept. of Computer Science, Faculty of Science, Minia University, El-Minia, Egypt; Email: abdelmgeid@yahoo.com