RETRIEVING DOCUMENT WITH COMPACT GENETIC ALGORITHM(cGA)

Sarab M. Hameed, Maisaa I. Abdul-Hussain, Zayneb R. Ahmed
Department of Computer, College of Science, University of Baghdad, Iraq-Baghdad.

Abstract

Information retrieval is the task, given a set of documents and a user query, of finding the relevant documents. Information retrieval applications require speed, consistency, accuracy and ease of use in retrieving relevant texts to satisfy user queries. This paper presents an automatic tool to retrieve documents based on Compact Genetic Algorithm (cGA). The similarity between queries and documents is computed with cosine coefficient, dice's coefficient, and Jaccard coefficient that are used as the fitness functions. Experimental results show that cGA can be successfully applied to information retrieval.

استرجاع الوثائق باستخدام الخوارزمية الوراثية المضغوطة (cGA)

سراب مجيد حميد ، ميساء ابراهيم عبدالحسين ، زينب رائد احمد
قسم الحاسبات, كلية العلوم, جامعة بغداد , بغداد - العراق.

الخلاصه

استرجاع المعلومات هي المهمة التي تعطى مجموعة من الوثائق وطلبات المستخدم لإيجاد الوثائق ذات العلاقة . تطبيق استرجاع المعلومات يحتاج للسرعة ,أتساق ,دقه وسهولة الاستعمال في استرجاع المعلومات ذات العلاقة لتلبية طلبات المستخدم . يقدم هذا البحث أداة آلية لاسترجاع الوثائق بالاعتماد على الخوارزمية الوراثية المضغوطة (cGA). إن التشابه بين الطلبات والوثائق محسوبة بمعامل جيب التمام (cosine coefficient) ,معامل التردد (dice's coefficient) ومعامل (Jaccard) الذي استخدم ال(fitness functions). تشير النتائج التجريبية أن الخوارزمية الوراثية المضغوطة (cGA) يمكن أن تطبق بنجاح في استرجاع المعلومات.

Introduction

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [1]. An information retrieval system basically constituted by three main components as shown in figure [1]. The documentary data base, this component stores the documents and the representation of their information contents. It is associated with the indexer module, which automatically generates a representation for each document by extracting the document contents. Textual document representation is typically based on index terms (that can be either single terms or sequences), which are the content identifiers of the documents. The query subsystem allows the users to formulate their queries and presents the relevant documents retrieved by the system to them. To do so, it includes a query language that collects the rules to generate legitimate queries and procedures to select the relevant documents. The matching mechanism evaluates the degree to which the document representations satisfy the requirements expressed in the query, the retrieval status value (RSV), and retrieves those documents that are judged to be relevant to it.

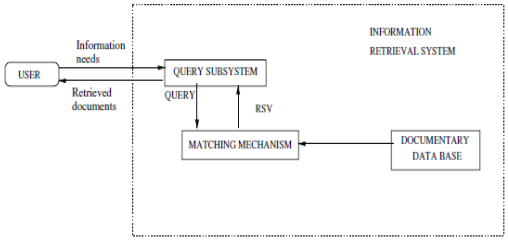

Figure 1: Typical IR System.

An IR model is a formal background defining internal document representation, query language and document-query matching mechanism.

Several retrieval models have been studied and developed in the IR area. In this paper two influential IR models: Boolean IR model and vector space IR model are presented.

Boolean model [2] [3] performs a binary indexing in the sense that a term in a document representations either significant (appears at least once in it) or not (it does not appear in it at all). 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.

Vector Space Model (VSM) [2][4] 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. This way, the retrieved documents can be orderly presented to the user with respect to their relevance to the query.

Many artificial intelligence techniques are used in IR including Klabbankoh [5] used a Genetic Algorithms (GA's) to increase information retrieval efficiency. GA's are probabilistic search methods which applied natural selection and natural genetics in artificial intelligence to find the globally optimal solution to the optimization problem from the feasible solutions. Belew [6] 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. [7].

Compact GA (cGA) was introduced to solve the general optimization problem which represents the population as a probability distribution over the set of solutions and is operationally equivalent to the order-one behavior of the simple GA

As the matter of design, the cGA shows an interesting way of getting more information out of a finite set of evaluations [8].This paper uses cGA to retrieve the documents under VSM.

VSM is based on interpretation of both, documents and queries, as points in a multidimensional document space.

The dimension of the document space is given by the number of indexed terms in the documentary collection.

The reminders of this paper are organized as follows. Section two illustrates the application of cGA in IR and retrieves the required documents. Section three demonstrates the obtained results from the cGA. Finally, section four presents the final remarks of cGA in IR.

cGA BASED DOCUMENTS RETRIEVAL

This section illustrates the application of cGA in IR based on VSM. In VSM model the document is represented by vector of terms and a particular query is represented by vector of query terms. The cGA is used to match the query with document representations and finds the optimized query which retrieves the required documents. Figure 2 shows the flowchart of the cGA.

Following steps illustrates the cGA for retrieving documents

Step1: Initialize the probability vector (p) with value (0.5). The length of p is set to the number of total terms in the documents.

For i=1 to n (n equal to total terms in the documents)

P[i] =0.5

Step2: Generate two individuals a and b. (i.e. generation procedure that compute the new individual based on the probability vector p)
a= generate(p)
b= generate(p)
The individual is coded as a binary string in which each individual gene represents the existing of the corresponding term in the documents.

Step3: Compute the fitness value for a and b. Then, compete a and b and keep the maximum fitness value as the winner, and keep the minimum fitness value represented as the loser. Equations 1, 2, and 3 demonstrate cosine coefficient, Dice's coefficient**,** and Jaccard coefficient used as the fitness to measure the similarity between query and document [7].

$$Sim(D,Q) = \frac{D.Q}{|Q||D|} \quad (1)$$

$$Sim(D,Q) = \frac{2|D \cap Q|}{|Q||D|} \quad (2)$$

$$Sim(D,Q) = \frac{|D \cap Q|}{|Q \cup D|} \quad (3)$$

Where *D* and *Q* represent the document term vector and query vector respectively.

Step 4: Update the probability vector *p* from the winner and loser.
for i=1 to n do
if winner[i] ≠ loser[i]
  if winner[i] = 1 then p[i] = p[i] + (1/n)
  else p[i] = p[i] – (1/n)
Where n is the population size.

Step 5: Check if *p* has converged
for i=1 to n do
If (p[i] >0 and p[i]<1) or (p[i] >threshold$_1$ and (p[i]< threshold$_2$) then return to step 2.

Step 6 If the solution is converged, *p* is the optimal solution which represent the optimized query, and then decode the query to retrieve the documents from the database.

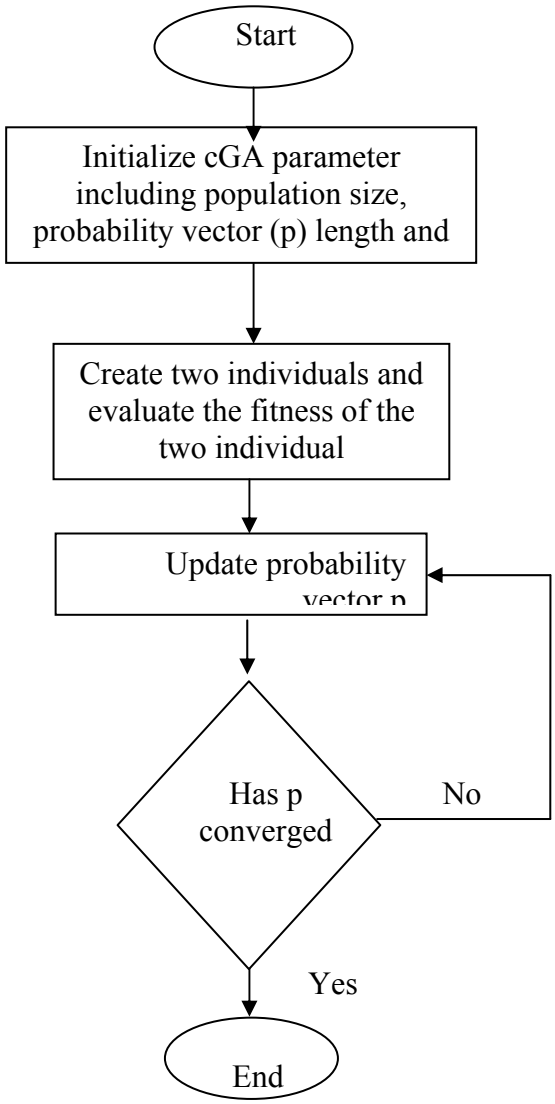

Figure 2: Flowchart of cGA.

EXPERIMENTAL RESULT

cGA document retrieval is tested with 100 documents stored in the database. The cGA document retrieval tested with documents containing sixteen terms. These terms are artificial intelligence, computer networks, data retrieval, databases, DBMS, expert system, fuzzy logic, indexing, information retrieval system, internet, multimedia, natural language processing, neural network, object-oriented, query, relational databases (i.e. *p* length is set to 16) .

Additionally, cGA document retrieval tested with documents contains thirty two terms including relational databases, query, data retrieval, computer networks, DBMS, artificial

intelligence, internet, indexing, natural language, processing, databases, expert system, information retrieval system, multimedia, fuzzy logic, neural network, lexical analysis, object-oriented, syntax and semantics, HTML, search engines, website, page rank, world wide web, computer architecture, genetic algorithms, crossover probability, introduction, operating system, visual basic, parser, SQL, oracl (i.e. *p* length is set to32).

Precision and recall criteria are used to measure the effectiveness of cGA document retrieval. Precision is the fraction of the documents retrieved that are relevant to the user's information need. Recall is the fraction of the documents that are relevant to the query that are successfully retrieved [7]. Equations 4 and 5 illustrate the formula for precision and recall respectively [1].

$$precision = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} \quad (4)$$

$$recall = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items in collection})} \quad (5)$$

In this paper, the threshold_1 was set to 0.01 and threshold_2 was set to 0.99 and the population size (n) is set to 10. Table 1 and 2 demonstrate the precision and recall of cGA document retrieval when the number of terms equals to 16 and 32 respectively. The 1st column illustrates the query in natural language, the 2nd column illustrates the optimal representation of the query that we hope to be obtained from cGA document retrieval results (i.e. *p*). The 3rd column demonstrates the retrieved documents and relevant to user query. The 4th column represents the total number of documents that are retrieved. The 5th and 6th columns describe the evaluation measures of cGA document retrieve.

The results show that cGA document retrieval is able to retrieve documents with high precision when the query contains terms identical to the terms in the documents and gives high recall depends on the terms in the documents.

Table 1: cGA Document Retrieval Effectiveness with 16 Terms

Query	p with 16 Terms	Retrieve and Relevant	Retrieve	Recall	Precision
Internet	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	17	25	0.68	0.68
Internet indexing	0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0	17	34	0.5	0.5
Databases search	0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1	14	40	1	0.37
Search engine	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	2	26	0.2	0.21
information retrieval system	0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0	34	39	1	0.98
Fuzzy logic system	0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 0	21	37	0.9	0.94
Fuzzy logic	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0	1	2	0.5	0.5
Operating system	0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0	36	36	1	1

Table 2: Effectiveness of cGA Document Retrieval with 32 Terms4

Query	P with 32 Terms	Retrieve and Relevant	Retrieve	Recall	Precision
Internet	00000000000000 10000000000000 000000	17	25	0.68	0.72
Internet indexing	00000000000101	12	34	1	0.35
Databases search	00000100000000 00000000000001 100000	11	40	0.6	.45
Search engine	00000000000000 00000000000001 00000	4	26	0.3	0.81
Information retrieval system	00001001000010 00000010000000 00000	26	39	0.97	0.91
Fuzzy logic system	00000001100010 00000001000000 000000	29	37	0.66	0.783
Fuzzy logic	0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 00000000000000 000	1	2	0.7	0.8
Operating system	00000001000010 00000010000000 000	34	36	0.97	0.94

The results show that cGA document retrieval is able to retrieve documents with high precision when the query contains terms identical to the terms in the documents and gives high recall depends on the terms in the documents.

CONCLUSION

This paper described a method of utilizing compact genetic algorithms in the field of information retrieval. This algorithm was tested on 100 documents collection and the number of terms in the query either 16 or 32. The results show that cGA document retrieval was acceptable with suitable precision and re call measure.

References

1. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, **2009**. *An Introduction To Information Retrieval.* Cambridge University Press, Cambridge, Englandm.
2. Cordon, O. A., Herrera-Viedma, E. A., Lopez-Pujalte, C. B.; Luque, A. M.; Zarco C. C. **2003**. A Review on the application of evolutionary computation to information retrieval. *Inter. J. of Appro. Reas*. **34**:241–264.
3. Van Rijsbergen, C. J. **1979**. *Information Retrieval.* Second ed., Butterworth.
4. Salton, G.; McGill, M. H. **1983.** *Introduction to Modern Information Retrieval*, McGraw-Hill.
5. Klabbankoh B., "Applied Genetic Algorithms in Information Retrieval", available at: http://www.journal.au.edu/ijcim/sep99/02-drouen.pdf.
6. Belew, R. **1989**. Adaptive information retrieval. Proceedings of the twelfth annual international Acm/Sigir conference on research and development in information retrieval, Pp: 11-20
7. Doszkocs, T.; Reggia, J. and Lin, X. **1990.** Connectionist models and information retrieval, *Annual Review Of Information Science And Technology*, **25**: 209-260.
8. Harik; Fernando G. Lobo and David E. Goldberg. November **1999**. The Compact Genetic Algorithm, *IEEE Trans. On Evolutionary Computation*, **3(**4).
9. Korfhage, R. R. **1997**. Information Storage and Retrieval. New York: Wiley Computer Publishing.