Available online at https://www.asjp.cerist.dz/en/PresentationRevue/134

RIST

Information Processing at the Digital Age Journal

Volume 27. No 1 (2023)

Using Genetic Algorithms to Improve Information Retrieval

F.Z. Bessai-Mechmache[1]*, Z. Alimazighi[2], K. Hammouche[1]

1 Research Centre on Scientific and Technical Information, CERIST, Ben Aknoun, 16028, Algiers, Algeria
2 Computer Science Department, University of Science and Technology Houari Boumediene, Algiers, Algeria

Abstract

Finding the valuable relevant information continues to be the major challenges of Information Retrieval Systems owing to the explosive growth of online web information. Among these challenges, we consider the XML Information Retrieval challenges as XML has become a de facto standard over the Web. In this paper, we tackle the issue of content-based XML information retrieval. We formulate the retrieval issue as a combinatorial optimization problem in order to generate the best set of relevant XML elements for a given keywords query. In our proposal, we define a genetic algorithm which maximizes similarity between a set of XML elements and the user query. The results based on the precision measure are very promising.

Keywords: Genetic Algorithm; Result Merging; Crossover Operator; Mutation Operator; Information Retrieval; XML Information; Retrieval.

1. Introduction

The development of electronic document and the Web have emerged and imposed structured data formats such as XML (eXtensible Markup Language) to represent information in a richer form adapted to specific needs. These formats allow representing jointly the textual information and the information of structure of a document. The logical structure or hierarchy of XML documents contains content and structural elements. The purpose of an XML information retrieval system is refined to retrieval strategies, which aim at returning document components, i.e. XML elements, instead of whole documents in response to a user, query Fuhr et al., 2004, Suma et al., 2014.

* F.Z. Bessai-Mechmache.*E-mail address: fatmazohrabm@gmail.com*

Content-based XML information retrieval considers just content-only queries. Content-only queries make use of content constraints only, i.e. they are made of keywords and are suitable for XML retrieval scenarios where users do not know, or are not concerned, with the document's logical structure when expressing their information needs.

Content-based XML information retrieval is a challenging task. In this researcher work, XML information retrieval is concerned with a refined strategy which involves searching a huge search space for the better selection and combination of relevant XML elements. As genetic algorithms (GA) are well suited for searching huge search spaces, in this paper, a content-based XML information retrieval model is proposed for efficient information retrieval systems using genetic algorithms, Abualigah and Anandeh, 2015, Ravi et al., 2012. In the proposed model, a genetic algorithm generates the best combination of XML elements from a set of selected XML elements.

The roadmap to the remaining part of the paper is as follows: Section 2 reviews some related work. Section 3 describes the proposed model with an approach for finding the optimal solution among all possible solutions. The experiment results are presented in Section 4. Finally, section 5 concludes this work and lists some perspectives.

2. Related Works

Content-based XML information retrieval addresses issues related to granularity, diversity and relevance of search results. So, authors in Lamlas 2014, Kopliku 2014, assume that search result is not necessarily a sorted list of independent XML elements; it could also be a meaningful aggregation of XML elements from various XML documents providing thereby more complete and less noisy information for users.

In literature, there are many works on generating meaningful query results for keyword XML search. Liu and Chen, 2015 address the query results display. Other works focus on how to identify keyword matches, Feng and Zhou, 2007, Sun et al., 2007. In XML information retrieval, Polyzotis and Garofalakis, 2006 were the first to propose a solution to merge search results by representing XML summaries. They grouped some XML elements and used small space to store their abstracts. Huang et al., 2009 focused on the problem of result snippet generation to generate meaningful small snippets. The model presented in Bessai-Mechmache and Alimazighi, 2011 and Bessai-Mechmache and Alimazighi, 2012 is close to our work. It is a content-based XML information retrieval model using Possibilistic networks to merge XML elements to provide a result including all relevant information for user query.

The approach we propose in this paper combines the search of relevant XML elements and their grouping in the same result to better answer the query all together. We address the issue in a context of a huge number of XML elements. The proposed approach is based on genetic algorithms which differentiate us from the frameworks used in previous approaches and allow us to select with the optimal possible way, the set of relevant XML elements that prevents the user from spending time searching the appropriate content in the ordered list of results.

3. The Proposed Genetic Model

In this paper, we focus on how to select the best set of XML elements from XML documents that are relevant and likely to better answer the query all together. We model this issue as a combinatorial optimization problem, which consists of finding the optimal combination (solution) in a search space containing all possible combinations (solutions). For this purpose, we propose an XML information retrieval model using genetic algorithms, which can contribute better performing information retrieval systems, Ahmed et al., 2008, Prajakta and Deipali, 2016, Mashagba et al., 2011. The proposed genetic algorithm merges XML

elements selected from several XML documents to build the optimal solution, which maximizes similarity between XML elements and user query.

The proposed model consists of two steps, namely XML element selection, and result merging according to user query. The XML element selection step extracts from XML documents, a set of relevant XML elements given the user's query. While the result merging combines outcomes (XML elements) of the previous step to find the optimal solution (combination) for the query.

The model architecture is shown in Figure1 below.

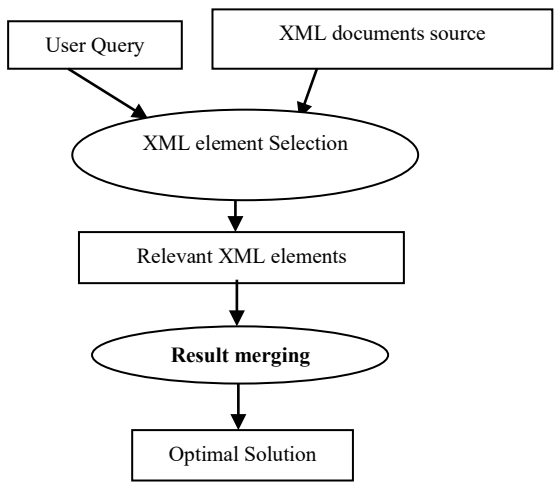

Fig. 1. Model architecture

3.1 XML Element Selection Process

The main task of XML element selection process is to select from an XML documents source the valuable relevant XML elements that better answer the user query formulated by keywords.

The relevance of an XML element is defined by an indexing process that uses the importance of terms of the element.

The importance or weight (W_{ij}) of term 't_i' in element 'e_j' is calculated as follows Trotman, 2005:

$$W_{ij} = tf_{ij} * ief * idf \quad (1)$$

Where 'tf_{ij}' is the frequency of term 't_i' in element 'e_j', '*ief*' is the inverse frequency of element 'e_j' for term 't_i', and '*idf*' is the inverse document frequency.

The relevance of an XML element with respect to query, denoted as *RSV (e, Q),* is equal to the sum of the weights of query terms with regard to this element Ogilvie and Callan, 2003.

$$RSV(e,Q) = \sum_{i=1}^{M} W_i^e \quad (2)$$

'W_i^e' is the weight of term 't_i' in element *'e'* (1), and *M* is the number of query terms.

3.2 Result Merging Process

The result merging process is considered as a combinatorial optimization problem, which consists of finding the optimal combination (solution) in a search space containing all possible XML element combinations. Thus, we address this problem by the use of intelligent methods in particular Genetic Algorithms, which are considered robust and efficient for large amount of data, Eiben and Smith, 2007.

Figure 2 shows the result merging process working.

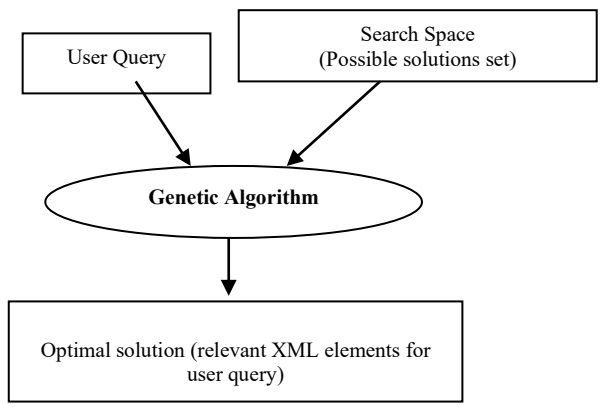

Fig. 2. Result merging process

3.3 Genetic Algorithms

When a genetic algorithm is applied to solve a problem, the first step is to define the problem and the search space:

- We define the problem of result merging with a pair (S, S'). S = {e_1, e_2, …, e_n} a set of n XML elements selected from an XML documents source according to the user query Q, and S' a subset of S such that the similarity between its elements and Q is maximal, where |S'| = k < |S| = n; k is the number of relevant XML elements for query Q.
- The search space includes all possible solutions of the problem. As a solution is a combination of k XML elements from n total XML elements, the size of the search space is expressed as follows:

$$C_n^k = \frac{n!}{k!(n-k)!} \tag{3}$$

When 'n' is very large, the number of possible combinations is enormous and NP-complete method is able to yield a solution of good quality. One method to deal with this issue is the use of Genetic Algorithms.

3.3.1 The proposed Genetic Algorithm

In genetic algorithms, the search space is a collection of candidate solutions to the problem; each represented by a string is termed as a chromosome. Each chromosome has an objective function value, called fitness function. A collection represented in set of chromosomes together with their associated fitness function is called the population. This population, at a given iteration of the genetic algorithm, is called a generation, Melanie, 1999, Sivanandam and Deepa, 2008.

When a genetic algorithm is applied to solve a problem, an initial population is then defined and three genetic operations that are selection, crossover and mutation are performed to generate the next generation. This process is repeated until the termination criterion is satisfied Gen and Cheng, 1997.

A solution to the result-merging problem is a set of k XML elements represented by a vector of length k. Thus, a possible solution is a vector of k XML elements belonging to the set $\{e_1, e_2 \ldots, e_n\}$. For example, if the number of XML elements 'n' is equal to 6 and the number of XML elements 'k' to be selected is equal to 4, a solution can be: *{*'e_1', 'e_2', 'e_4', 'e_6'*}*, *{*'e_2', 'e_3', 'e_5', 'e_6'*}* or *{*'e_3', 'e_4', 'e_5', 'e_6'*}* etc.

The fitness function is a performance measure function that evaluates the relevance of each solution. The relevance between a solution '*sol*' and a user query *Q*, denoted as *RSV (sol, Q),* is calculated by the average of the similarities between the XML elements of this solution and the user query.

$$RSV(sol, Q) = \frac{1}{|sol|} * \sum_{e_i \in sol} RSV(e_i, Q) \quad (4)$$

|*sol*| is the number of XML elements in the solution *'sol'*. *RSV (e_j, Q)* is the relevance of element 'e_j' with regard to user query *Q* (equation 2) .

The following algorithm outlines the proposed genetic algorithm to maximize similarity between XML elements composing a solution and the query. It works on two populations: the population of XML elements denoted 'Pe', and the population of solutions denoted 'Psol'.

Input: a user query *Q* and a set 'P_e' of 'n' XML elements generated in the XML element selection step.

Output: the optimal selection of XML elements for the query *Q*.

A) Genetic Algorithm:

1.Generate randomly an initial population 'P_{sol}' from the possible combinations of XML elements of set '*Pe*'. 'P_{sol}'= $\{sol_1, sol_2, \ldots, sol_m\}$.

In this step, the duplicate genes (XML elements) are avoided in the same chromosome (solution). We also avoid duplicating the same chromosome in the population.

2. Evaluate each solution in the initial population 'P_{sol}' using the fitness function given by (4).

3. Create a new population:

a. Select the best solutions (identified by their fitness value) for reproduction.

b. Apply crossover and mutation operators to 'P_{sol}' population to produce new solutions.

c. Add new solutions to 'P_{sol}'.

4. Evaluate the new population 'P_{sol}' using fitness function given by (4).

5. Control the stop criterion (maximum number of iterations to reach algorithm convergence): If the criterion is not reached, go to step 3.

In what follows, we describe crossover and mutation operators of the proposed genetic algorithm.

- Crossover operator: This operator combines two chromosomes together to form a new offspring. We use a single-point crossover.

B) Crossover Algorithm:

1. Select two parents used for crossover. Let Y = (y_1, y_2, ... y_n) and X = (x_1, x_2, ..., x_n) two solutions (chromosomes) to be crossed.

2. Randomly select any crossover point i (i =1 to n-1).

Two offspring X' and Y' are created by combining the parents at crossover point according to the following rules:

$$x'_k = \begin{cases} x_i & \text{if } k < i \\ y_i & \text{otherwise} \end{cases}$$

$$y'_k = \begin{cases} y_i & \text{if } k < i \\ x_i & \text{otherwise} \end{cases}$$

- Mutation Operator: Mutation is the process of randomly altering the genes in a particular chromosome. The objective of the mutation is restoring lost and exploring variety of data. We used a single-point mutation. A gene is changed with a certain probability 'Pm' while avoiding genes duplication. The proposed mutation algorithm is as follows.

C) Mutation Algorithm:

1**. For** each chromosome in 'P_{sol}**'** population **do**
2. Generate a random number k in interval [0, 1] (to select the chromosome to be mutated)
3. **if** (k < *Pm*) **then** apply the mutation operator to this chromosome
begin
a. Randomly select the XML element (gene) to be modified.
b. Choose the new value to be placed (it must be different from the values already existing in the chromosome).
c. Change the old value of the XML element by the new value.
d**.** Insert the new chromosome into the new population.
end
else the chromosome is inserted into the new population without change.
4. **end for**

4. Experiments

To evaluate the proposed model, a prototype has been implemented in Java environment.
An XML source provided by INEX'2005 (INitiative for the Evaluation of XML retrieval) collections has been used for assessment. INEX'2005 collection consists of scientific papers from the IEEE Computer Society, tagged in XML format. The collection is composed of articles from several review and journals and thousands of articles from Wikipedia encyclopedia. A medium sized article is composed of approximately 1500 elements. A document from the INEX'2005 collection consists of a single article which is composed of several sections (IEEE articles) which may belong to various volumes and dealing, at times, with different thematic not semantically related.

We also took a set of 20 test queries (composed of keywords) from the same collection. Table1 shows an example of test queries.

Table 1. Example of test queries

Query Identifier	Query
Q1	analysis and methodology
Q2	Software engineering
Q3	wireless multimedia
Q4	latent semantic analysis
Q5	machine translation approaches

4.1 Experiment results

For the evaluation, we calculate the precision factor Arguello, 2013 at the 5, 10, 15 and 20 top results. In order to show the influence of the use of genetic algorithms in the result, we compare the proposed model with the possibilistic model Bessai-Mechmache and Alimazighi, 2012.

We have computed two precision values of retrieval result returned respectively by the possibilistic model and the proposed model. Each precision value is computed for a top 5 XML elements, 10 XML elements, 15 XML elements and 20 XML elements.

We computed then two averages precisions over 20 test queries. Table 2 shows the average precision (at 5, 10, 15 and 20) values reached by each model.

Table 2. Average precision values of each model

Precision	Possibilistic model	Proposed model
Pr @ 5	0.82	0.85
Pr @ 10	0.74	0.79
Pr @ 15	0.63	0.76
Pr @ 20	0.59	0.72
Pr_avg	0.69	0.78

Figure 3 illustrates the precision values of the Possibilistic model and the proposed model.

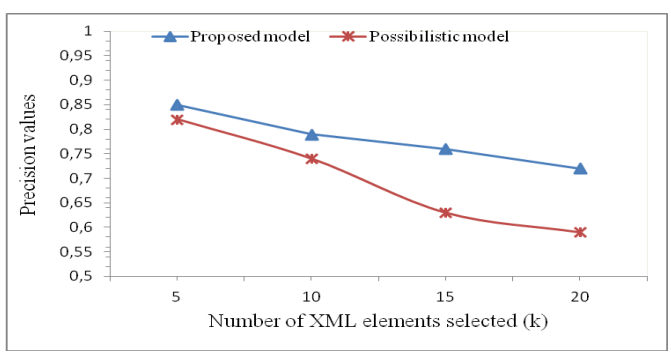

Fig.3. Precision values of the Possibilistic and the proposed models

The obtained results indicate that the proposed model performs better than Possibilistic model and confirm us that the use of genetic algorithms improves the relevance of search results relative to the Possibilistic model.

5. Conclusion

This paper defined a new XML information retrieval model using genetic algorithms for improving the retrieval process in content-based XML information retrieval. The reported results show that the use of a genetic algorithm approach improved relevance metrics in XML information retrieval systems.

Furthermore, thanks to the genetic algorithm approach, the proposed model provides the meaningful and optimal solution (among all possible solutions) that prevents the user from spending time searching the appropriate content in the displayed list of results.

The proposed model architecture consists of two phases, namely XML element selection, and result merging. This architecture is close to that of a distributed information retrieval system which motivates us in future works to extend the proposed model to provide an efficient solution to distributed XML information retrieval dealing with multiple sources of heterogeneous XML documents.

References

Fuhr, N., Lalmas, M., Malik, S. and Szlávik, Z., 2004. Advances in XML Information Retrieval. In Proceedings of Initiative of the Evaluation for XML Retrieval Workshop.

Kamps, J., Marx, M., de Rijke, M. and Sigurbjörnsson, B., 2003. XML Retrieval: What to retrieve? ACM SIGIR Conference on Research and Development in Information Retrieval, :409-410.

Kopliku, A., Pinel-Sauvagnat, K., Boughanem, M., 2014. Aggregated search: A new information retrieval paradigm. ACM Comput. Surv., 46(3): 41:1-41:31.

Sauvagnat, K., Boughanem, M., Chrisment, C., 2006. Answering content and structure-based queries on XML documents using relevance propagation. Information Systems, 31(7):621–635.

Sigurbjornsson, B., Kamps, J., De Rijke, M., 2003. An element-based approach to XML retrieval. In Proceedings of Initiative of the Evaluation for XML Retrieval Workshop.

Suma, D., Dinesh Acharya, U., Geetha, M. & Raviraja Holla, M., 2014. XML Information Retrieval: An overview. International Global Journal for Engineering Research, 10(1).

Abualigah, L.M.Q., Anandeh, E.S., 2015. Applying Genetic Algorithms to Information Retrieval using Vector Space Model. International Journal of Computer Science, Engineering and Applications (IJCSEA), 5(1).

Ahmed, A., Bahgat, A., Abdel Mgeid, A. and Osman, A.S., 2008. Using Genetic Algorithm to Improve Information Retrieval Systems. World Academy of Science, Engineering and Technology.

Arguello, J., 2013. "Evaluation Metrics. INLS 509: Information Retrieval.

Bangorn, K., Quen, P., 2005. Applied genetic algorithms in information retrieval. In Proceedings of International Journal of Production Research, 43: 4083-4101.

Katoch, S., Chauhan, S. S., Kumar, V., 2021. A review on genetic algorithm: past, present, and future. Multimedia Tools and Applications (2021) 80:8091–8126, https://doi.org/10.1007/s11042-020-10139-6.

Ravi, S., Neeraja, G., Raju, V., 2012. Search engines using evolutionary algorithms. International Journal of Communication Network Security ISSN: 2231–1882, 1(4):39–44.

Lalmas, M., 2014. Xml information retrieval. Understanding Information Retrieval Systems: Management, Types, and Standards.

Kopliku, A., Pinel-Sauvagnat, K., Boughanem, M., 2014. Aggregated search: a new information retrieval paradigm. ACM Computing Surveys, 46 (3):1-31.

Liu, Z., Chen, Y., 2007. Identifying meaningful return information for xml keyword search. Special Interest Group on Management of Data SIGMOD, :329–340.

Feng, G. Li, J., Zhou, J., 2007. Effective keyword search for valuable lcas over XML documents. Conference on Information Knowledge Management CIKM, :31–40.

Sun, C., Chan, C., Geonka, A., 2007. Multiway SLCA-based Keyword Search in XML Data. International World Wide Web Conference, 2007, :1043–1052.

Polyzotis, N., Garofalakis, M., 2006. XCluster Synopses for Structured XML Content. In proceedings of 22nd International Conference on Data Engineering.

Huang, Y., Liu, Z., Chen, Y., 2008. Query Biased Snippet Generation in XML Search. ACM SIGMOD, :315-326.

Bessai-Mechmache, F.Z., Alimazighi, Z., 2011. Possibilistic Networks for Aggregated Search in XML Documents. In proceedings of International Conference on Information & Communication Systems, :67-72, ISBN 978-1-4507-8208-1.

Bessai-Mechmache, F.Z., Alimazighi, Z., 2012. Possibilistic Model for Aggregated Search in XML Documents. International Journal of Intelligent Information and Database Systems, 6(4):381-404, H Index 10.

Prajakta, M., Deipali, G., 2016. Improving the Performance of Information Retrieval System using AGA in Distributed Environment. International Journal of Innovative Research in Computer and Communication Engineering, 4(7).

Siva, S., Philomina, S., 2010. A Document Retrieval System with Combination Terms Using Genetic Algorithm. International Journal of Computer and Electrical Engineering, ; 2(1).

Al Mashagba, E., Al Mashagba, F., Nassar, M.O. , 2011. Query optimization using genetic algorithm in the vector space model. Int. J. Comp. Sci., 8(3): 450–457.

Trotman, A ., 2005. Choosing document structure weights. Information Processing and Management, 41(2): 243-264, IF 3.444.

Ogilvie, P., Callan, J., 2003. "Using language models for flat text queries in XML retrieval", In Proceedings of Initiative of the Evaluation for XML Retrieval Workshop, :12-18.

Eiben, A.E., Smith, J.E., 2007. Introduction to Evolutionary Computing, Springer, Heidelberg, ISBN 978-3-540-40184-1.

Melanie, M., 1999. An Introduction to Genetic Algorithms, A Bradford Book, The MIT Press Cambridge, Massachusetts, England.

Sivanandam, S.N., Deepa, S.N., 2008. Introduction to Genetic Algorithms, Springer, Heidelberg New York, ISBN 978-3-540-73189-4.

Gen, M., Cheng, R. , 1997. Genetic Algorithms and Engineering Design. Copyright © 1997 John Wiley & Sons, Inc.