A Framework for Efficient Document Ranking Using Order and Non Order Based Fitness Function

Hazra Imran , Aditi Sharan

***Abstract*—One central problem of information retrieval is to determine the relevance of documents with respect to the user information needs. The choice of similarity measure is crucial for improving search effectiveness of a retrieval system. Different similarity measures have been suggested to match the query and documents. This study investigates the use of Genetic Algorithm to increase the efficiency of information retrieval by defining a combined similarity measure. Genetic Algorithm has been used for learning weights of the components of the combined similarity measure. We have provided a weight-learning algorithm for the same. We have considered order based and non-order based fitness functions to evaluate the goodness of the solution. A non-order based fitness function is based on recall-precision values only. However, it has been observed that a better fitness function can be obtained if we also consider the order in which relevant documents are retrieved. This leads to an idea of order based fitness functions. We evaluated the efficacy of a genetic algorithm with various fitness functions. Further, we provide a framework for applying genetic algorithms to improve the retrieval efficiency by combing various similarity measures. The experiments have been carried out on TREC data collection. The results have been compared with various well-known similarity measures.**

***Index Terms*—Document retrieval, genetic algorithms, similarity measures, information retrieval, vector space Model.**

I. Introduction

Information retrieval system is devoted to finding "relevant" documents with respect to users query. Standard IR systems are based on Boolean [1], vector [10], and probabilistic models [11]. Each model describes documents, queries and provides algorithms to compute similarity between user's query and documents. The objective of an information retrieval system is to provide its users with satisfactory retrieval results. Toward this objective a retrieval result should be scientifically measured. Two essential evaluation criteria for retrieval success are recall and precision.

Precision is defined as the ratio of the number of relevant retrieved documents to the total number of retrieved documents.

Manuscript received December 3, 2009.

Hazra Imran, is working as an Assistant Professor in Department of Computer Science, Jamia Hamdard ,India. She is pursuing her PhD from Jawaharlal Nehru University, India. (Phone 0-9818067996 e-mail: hazrabano@ gmail.com).

Dr Aditi Sharan, is working as an Assistant Professor in School of Computers and System Sciences, Jawaharlal Nehru University, India (e-mail: aditisharan@mail.jnu.ac.in).

$$\text{Precision} = \frac{\text{no. of relevant documents retrieved}}{\text{no. of documents retrieved}} \quad (1)$$

Recall is defined as ratio of number of relevant retrieved documents to the total number of relevant documents.

$$\text{Recall} = \frac{\text{no. of relevant documents retrieved}}{\text{Total no. of relevant documents}} \quad (2)$$

Our proposed framework is based on Vector Space Model (VSM). In VSM, both documents and the user given query are represented as vector of terms. Suppose there are t index terms in a collection of documents. Then document D_i and query Q can be represented as

$$D_i=(d_{i1}, d_{i2},... d_{it})$$

$$Q=(w_{q1},w_{q2},...w_{qt})$$

Where d_{ij} (j=1 to t) are term weights in document D_i and w_{qj}(j=1 to t) are term weights in the query Q. There are different methods of assigning weights to d_{ij}. Most popular method being used is TFXIDF. Here TF (term frequency) measures the number of times a term appears in a document or query. The IDF (inverse document frequency) is based on the intuition that a term, which occurs in many documents, is not a good discriminator and should be given less weight than one, which occurs in few documents. One of the ways of calculating IDF is log(N/DF) where N is the total number of documents in the collection and DF is the number of documents in which term has appeared in entire document collection. Other measures can be found in [4].

For calculating similarity of query and document cosine and jaccard are the popularly used similarity measures. Cosine, jaccard and dice similarity measures are defined as follows:

$$\text{Cos}\ (Q,D_i) = \frac{\sum_{j=1}^{t} w_{qj} d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}} \quad (3)$$

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\text{JACCARD}(Q,D_i)= \frac{\sum_{j=1}^{t} w_{qj} d_{ij}}{\sum_{j=1}^{t} (d_{ij})^2 + \sum_{j=1}^{t} (w_{qj})^2 - \sum_{j=1}^{t} w_{qj} d_{ij}} \quad (4)$$

$$Dice = \frac{2\sum_{j=1}^{t} w_{qj} d_{ij}}{\sqrt{\sum_{j=1}^{t} (d_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}} \quad (5)$$

In this paper we focus on applicability of GA for improving relevance of retrieved documents with respect to user query. The method is tested on real document collections TREC and results are very encouraging. The paper is organized as follows: Section 1 dealt with the introduction of information retrieval system. Section 2 deals with Combined Similarity measure using Genetic algorithm. Section 3 discusses the framework and the algorithm in detail. Section 4 focuses on the experiments and results. Section 5 concludes the work.

II. Combined Similarity Measure Using Genetic Algorithm

In our problem, we have defined a combined similarity measure consisting of standard similarity measures. We then assign appropriate weights to the components of these measures so as to achieve maximum retrieval efficiency. Our combined similarity measure is a weighted sum of the scores returned by different matching measures. In general a combined similarity measure is represented as:-

$$combined\ \_similarity\ \_measure\ (D,Q) = \sum_{i=1}^{M} (wt_i \times SM_i(D,Q)) \quad (6)$$

Where $SM_i(D,Q)$ signifies the score of document D for given query Q for i^{th} similarity measure. M is total number of standard similarity measures considered. We have used two similarity measures: cos(SM_1) and jaccard(SM_2) (Refer Eqn 3 and 4). wt_i is the weight assigned to i^{th} similarity measure. Weights wt_1 and wt_2 range from 0.0 to 1.0. A less weight signifies that associated similarity measure is less significant. We have used GA for finding optimal set of weights. Genetic Algorithms (GA) comes under the intelligent search methods. GA's apply natural evolution process in artificial intelligence to find a globally optimal solution to the optimization problem. Genetic Algorithms are learning algorithms as well as offer a domain independent search ability that can be used in many learning tasks. Due to this reason, the application of GAs to IR has increased in the last decade [5,6,8, 9, 12].

In next section we present our framework for finding optimal weights using combined similarity measure.

III. Framework

Figure 1 presents overall framework of our proposed system. The input to the framework is vector representations of document and query and output is the optimized weights for combined similarity measure .We will now discuss the working of each module.

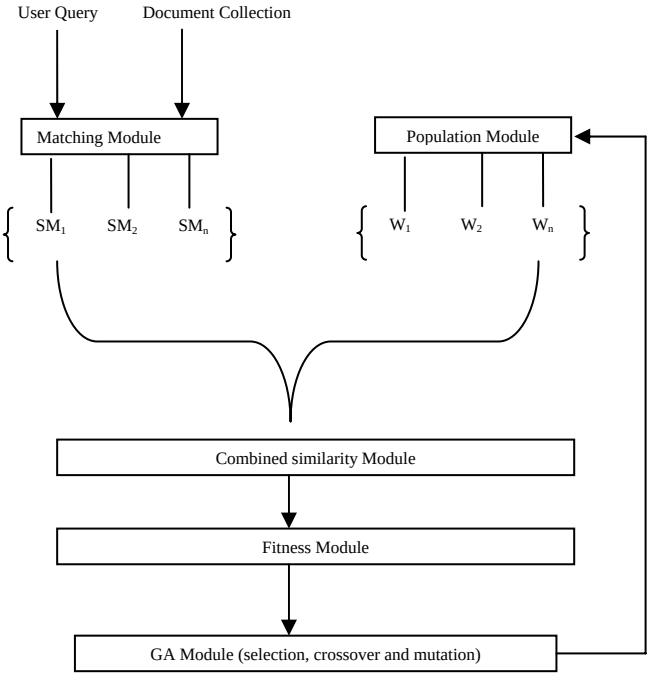


Fig. 1. Framework of Proposed System

3.1 Matching Module

The inputs to the module are the vector space representations for user query and the documents. Matching module calculates the similarity of the documents and query with respect to different similarity measures given in above equations and returns these values as $SM_1, SM_2, \ldots SM_n$.

3.2 Population Module

The population module generates chromosomes for initial population of GA. The chromosome is represented in the following way:

wt_1	wt_2	…	wt_n

Where wt_i = weight of the i^{th} similarity measure.
We have used a real-valued chromosome because it is more natural representation for our problem and it decreases dramatically the number of genes required to specify a design, thus making the solution space easier to search.

3.3 Combined Similarity Module

Considering similarity measures and weights as input, combined similarity module calculates similarity of the document with respect to combined similarity measure using equation 6.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

3.4 Fitness Module

Fitness module is used to find the fitness of the solution.

3.4.1 Fitness Function

Fitness function is a performance measure that is used to evaluate how good each solution is. Given a chromosome, the fitness function must return a numerical value that represents the chromosome's utility. Previous work has been done in GA considering fitness functions based on recall and precision only. However it has been observed that a better fitness function can be obtained if we also consider the order of the retrieval of documents.

In our work we have compared the order and non-order based fitness functions. Non order- based fitness function is based on recall and precision only. We have used following non-order and order- based fitness functions [2,7].

$$\text{Fitness1}_{\text{non order-based}} = \frac{(2 \times recall \times precision)}{recall + precision} \quad (7)$$

$$\text{Fitness2}_{\text{order-based}} = \frac{1}{|D|}\sum_{i=1}^{|D|}\left(r(d_i)\sum_{j=i}^{|D|}\frac{1}{j}\right) \quad (8)$$

Where $|D|$ is the total number of documents retrieved, and r(d) is the function that returns the relevance of document d, giving 1 if the document is relevant and a 0 otherwise.

Order-Based Fitness function takes into account the number of relevant and irrelevant documents along with the order of their appearance. Note that the documents retrieved earlier in order have higher utility in comparison to documents retrieved later. The basic idea being that it is not the same that the relevant documents appear at the beginning or at the end of the list of retrieved documents. The importance of the order-based fitness function can be justified with the following example.

Let us assume that two similarity measures retrieve 6 relevant documents in the top 15 result giving Case1 and Case2, where

Case 1 = 100100100111000

Case 2 = 111010110000000

The relevance information is coded as 1 (if relevant) and 0 (if irrelevant).

Here we observe that the recall in both the cases is same. However one can easily see that Case2 has a better performance as all the 6 documents are retrieved earlier in comparison to Case1. If we consider the total number of relevant document as 10 then the values for $\text{Fitness1}_{\text{non order-based}}$ is .48 for both the cases whereas $\text{Fitness2}_{\text{order-based}}$ are calculated as 0.45 for Case1 and 0.64 for Case2. Therefore we can justify that $\text{Fitness2}_{\text{order-based}}$ is better.

3.4.2 Working of Fitness Module

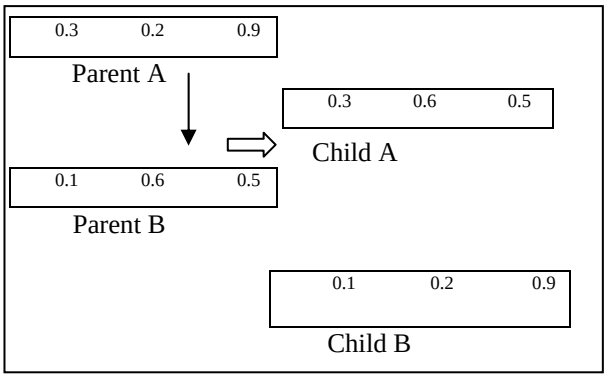
The fitness module sorts the population on the basis of the combined similarity measure obtained. Further precision and recall are calculated and finally the fitness is calculated for each member of the population using equation 7 and 8. Once the fitness is calculated next modules applies standard GA functions: selection, crossover and mutation and generates new population. The whole process is repeated iteratively until population converges or maximum number of iterations has been carried out.

3.5 Genetic Algorithm Module

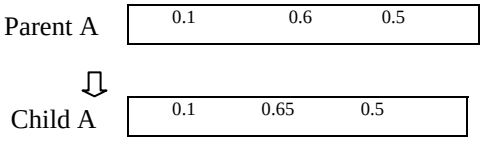
This module applies the standard GA functions (selection, crossover, mutation) to generate new population from the old population, which is discussed as follows.

• Selection: - Selection embodies the principle of 'survival of the fittest'. Chromosomes having higher fitness are selected for crossover. The roulette wheel reproduction process [3] was used to select individuals.

• Crossover: - Crossover is the genetic operator that combines two chromosomes together to form new chromosome. We have used single point crossover where a locus position is selected within two parent chromosomes and the genes are swapped from that position to the end of parent.



Mutation: - Mutation involves the modification of the value of each gene of a solution with some probability (mutation probability).



3.6Algorithm

We have used the following algorithm to find the weights of terms in CSM using Genetic Algorithm.

GA_combined_similarity_measure (query, documents)
Begin
Indexing (documents and query)
Build TF-IDF term document matrix
For each document find SM_1, SM_2 , . . . SM_n
Generate initial random population w_1, w_2. . . w_n
Repeat
For each member of population and for each document
Find combined_similarity_measure
For each member of population

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Fitness_calculation(current population, combined_
similarity _measure for all documents)
Perform fitness based selection, crossover, mutation
Make rank based selection and generate new population
Until fitness value is stabled OR reaches to maximum number of generations
End

Fitness_calculation (Individual population, combined
similarity measure for all documents)
Begin
Sort the documents based on combined similarity measure
Select top N documents
Find recall and precision values
Calculate fitness value using formula given
End

IV. Experiments and Results

For our experiments, we used volume 1 of the *TIPSTER* document collection, a standard test collection in the IR community. Volume 1 is a 1.2 Gbyte collection of full-text articles and abstracts, divided into seven main files. The documents came from the following sources.

1. WSJ -- Wall Street Journal (1986, 1987, 1988, 1989,1990,1991 and 1992)
2. AP -- AP Newswire (1988,1989 and 1990)
3.ZIFF -- Information from Computer Select disks (Ziff-Davis Publishing)
4. FR -- Federal Register (1988)
5. DOE -- Short abstracts from Department of Energy

The experiments were done for non-order-based (Fitness1nonorderbased) and order based (Fitness2orderbased) fitness functions. The control parameters were as follows: population size=100, probability of crossover (Pc = 0.7) , and probability of mutation (Pm = 0.01).The document cut off was 10. We have performed our experiment on 50 queries. Below is the example of applying Genetic Algorithm on TREC Query 50.

Analyzing query: -
Topic: Airbus Subsidies
Description:
Analyzing query: -
Topic: Airbus Subsidies
Description:
Document will discuss government assistance to Airbus Industrie, or mention a trade dispute between Airbus and a U.S. aircraft producer over the issue of subsidies.

Individual
[0.8627470125595492,
0.06433172471523907,0.6567819656192779]
Fitness : 0.6028985507786274

Individual
[0.8627470125595492, 0.06433172471523907,
.6567819656192779] Fitness : 0.6057971015572547

Individual
[0.8627470125595492, 0.06433172471523907,
0.6567819656192779] Fitness : 0.6086956523358822

Individual
[0.702370157345306, 0.2375410647497328,
0.9217589734864721] Fitness : 0.6028985507786274

Below are the graphs to show the results of experiment.

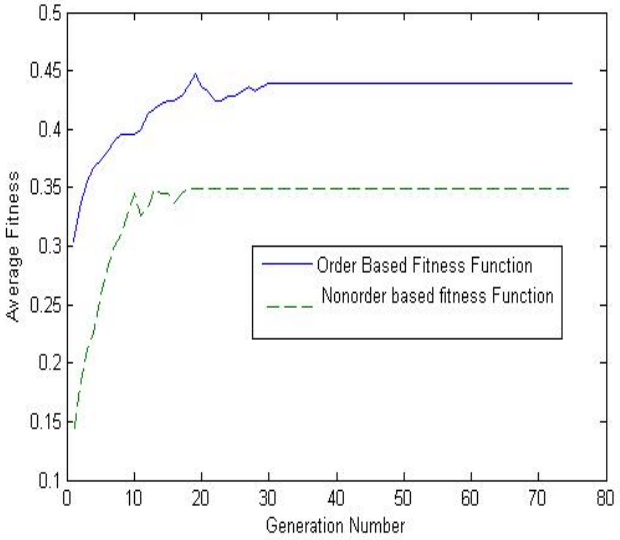


Fig. 2. Curve showing variation of Average fitness with Generation Number for both the fitness function

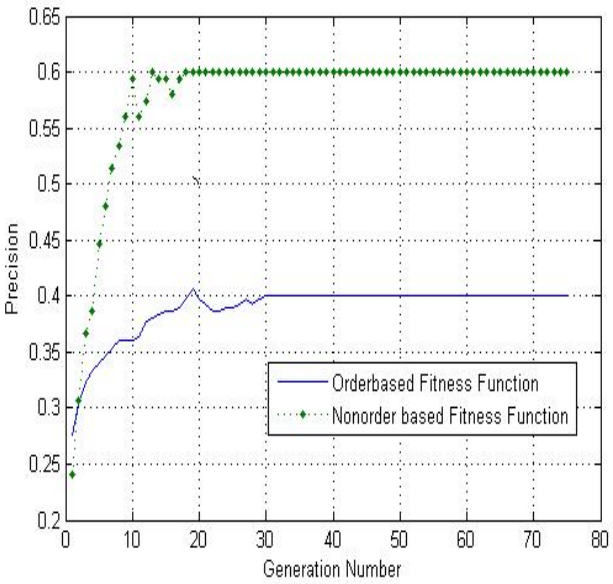


Fig. 3. Curve showing variation of Precision with Generation number for both the fitness function

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

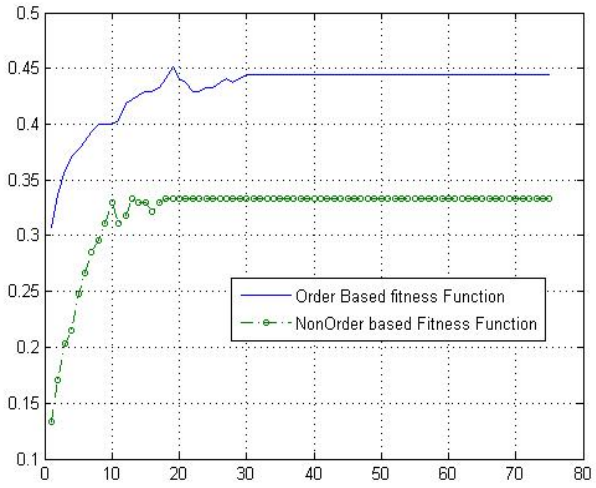


Fig. 4. Curve showing variation of Recall with generation number for both the fitness functions.

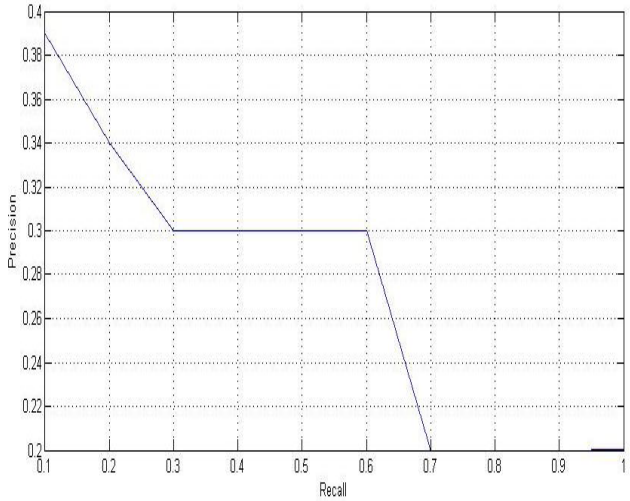


Fig. 5. Curve showing Recall and Precision Graph of combined Similarity Measure

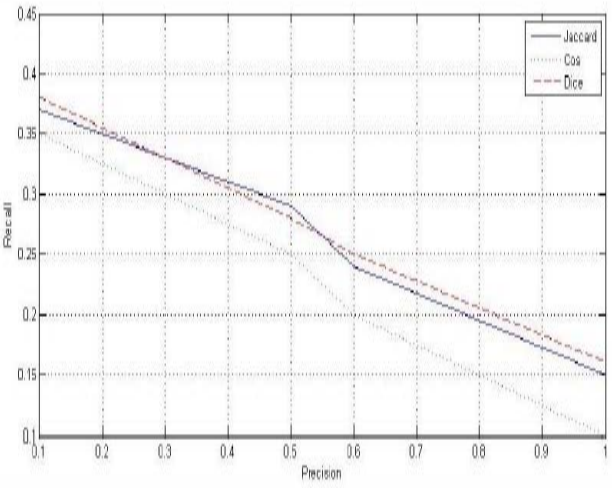


Fig. 6. Curve showing Recall Precision Graph of cos ,Jaccard and Dice Coefficient

Figure2 shows variation of average fitness with generation number for a specific query 50 of TREC dataset. As shown the average fitness increases with generation number. On the basis of our experiment we observed that for 55% and 60% of the queries of TREC dataset has increased average fitness in successive generations. Figure 3 shows curve showing variation of Precision with Generation Number for both the order and non order based fitness functions and Figure 4 shows curve showing variation of Recall with generation Number for both the fitness function. We compared the results of our experiments with the recall and precision values obtained by all three standard similarity measure ie. Cos, Jaccard Dice and combined similarity measure. For almost all the queries our experiment gave better results. Figure 5 shows recall precision curve for all the three standard measure for a specific query in TREC. Document cutoff was 10. Figure 6 shows recall precision graph for GA based experiment for the same dataset for the combined similarity measure. Recall-precision curve is a standard curve for measuring efficiency of any retrieval algorithm. They are useful because they allow us to evaluate quantitatively both quality of overall answer set and breadth of retrieval algorithm. Good quality information retrieval algorithm gives high precision at low recall values. By comparing figure 5 with figure 3 and 4 one can easily observe that initially all similarity measures start with high precision values. However standard similarity measures show a sharp decline in the beginning whereas our combined similarity measure shows gradual decrease in precision. This indicates that combined similarity measure gives better quality of result. The breadth of combined similarity measure is maximum as it is giving a recall value 1.

V. CONCLUSION

In this paper we have focused on the application of Genetic Algorithm for document ranking. We have used a combined similarity measure consisting of different standard similarity measures. GA was used for learning the weights of the components of the combined similarity measure. The paper provides a framework for learning optimized weights used in combined similarity measure. . The algorithm was implemented for TREC dataset with the order and non order based fitness functions. The results were compared with standard similarity measures using standard recall precision graphs. It was expected that order based fitness function should gives better result. As expected the empirical result verified the same. Therefore we can conclude that it is desirable to use fitness functions that value not only whether the possible solution retrieves many relevant documents and few irrelevant documents, but also whether the relevant documents are at the top of the retrieval list or at the end.

REFERENCES

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

[2] C.-H Chang, “The design of an information system for hypertext retrieval and automatic discovery on WWW”, PhD thesis, Department of CSIE, National Taiwan University,1999.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[3] D. E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning” , Addison-Wesley, Reading, Mass,1989.
[4] G. Salton and C. Buckley, “Term weighting approaches in automatic text retrieval,” *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.
[5] R. José, Pérez-Agüera,” Using Genetic Algorithms for Query Reformulation, BCS IRSG Symposium: Future Directions in Information Access (FDIA 2007), 2007.
[6] D.H. Kraft, et. al. “The Use of Genetic Programming to Build Queries for Information Retrieval”, *In Proceedings of the First IEEE Conference on Evolutional Computation.* New York: IEEE Press., PP. 468-473.,1994.
[7] K. L. Kwok, “Comparing representations in Chinese information retrieval”, *In Proceedings of the ACM SIGIR*, pages 34-41,1997.
[8] Lourdes Araujo , R Jose , “Improving Query Expansion with Stemming Terms: A New Genetic Algorithm Approach “Evolutionary Computation in Combinatorial Optimization,2008
[9] M.J. Martin-Bautista, et. al. “An Approach to An Adaptive Information Retrieval Agent using Genetic Algorithms with Fuzzy Set Genes”, *In Proceeding of the Sixth International Conference on Fuzzy Systems*. New York: IEEE Press. PP.1227- 1232.,1997.
[10] S. E. Robertson, “The probabilistic character of relevance”, Information Processing & Management, 13, pp: 247-251,1997.
[11] G. Salton, & M. McGill, "Introduction to Modern Information Retrieval", New York: McGraw-Hill, 1983.
[12] Z. Michalewicz, “Genetic Algorithms +Data Structures=Evolution Programs”,Springer-Verlag, 1996.

ISBN: 978-988-17012-8-2
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)