Australian Journal of Basic and Applied Sciences, 7(13) November 2013, Pages: 66-72

AENSI Journals

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Investigating Genetic Algorithms To Optimize The User Query In The Vector Space Model

[1]Mohammad Othman Nassar, [2]Feras Fares Al Mashagba, [3]Eman Fares Al Mashagba

[1]*Amman Arab University, Computer Information Systems department, Jordan.*
[2]*Amman Arab University, Computer Information Systems department, Jordan.*
[3]*Zarqa University, Computer Information Systems department, Jordan.*

ARTICLE INFO

Article history:
Received 10 January 2013
Received in revised form 14 November 2013
Accepted 15 November 2013
Available online 20December 2013

Keywords:
information retrieval, vector space model, query optimization, genetic algorithms.

ABSTRACT

Background: Very little research has been carried out on Arabic text. Arabic text nature and properties are different than the English and other languages texts, Arabic language text preprocessing is more challenging. This study proposed then explored the use of Genetic Algorithms (GA) in the vector space model (VSM) for the Cosine, and Jaccard as a similarity measures and fitness functions**. Objective:** This paper proposed, created, then compared 15 different combinations for the GA to improve the user query in the VSM when the Cosine, and Jaccard are used as a similarity measures. The GA combinations was created based on mixing a number of crossover strategies, mutations ideas, and fitness functions. Our goal is to find the best GA combination that can achieve the highest improvement for the user query in the VSM when the data collection is an Arabic language data collection. **Results:** Our results shows that; when using the Cosine, and Jaccard as a similarity measures then the GA approach which uses the Jaccard similarity as a fitness function, one-point crossover operator, and chromosomal mutation is the best IR system in VSM with 12.48% improvement compared to the traditional approach**. Conclusion:** when comparing our results with other studies that used the DICE, and Inner Product as a similarity measures; we recommend the GA approach which uses one-point crossover operator, point mutation, and Inner Product similarity as a fitness function to be used with the Arabic data collections since it is the best combination that can improve the performance.

To Cite This Article: Mohammad Othman Nassar, Feras Fares Al Mashagba, Eman Fares Al Mashagba., Investigating Genetic algorithms to optimize the user query in the vector space model. ***Aust. J. Basic & Appl. Sci.****, 7(13): 66-72, 2013*

INTRODUCTION

Information retrieval (IR) can be defined as the study of how to retrieve from a collection of stored information the parts which are responsive and related to particular query (Tengku *et al.*, 1990). Retrieval models such as Boolean model, vector space model, Fuzzy sets model and the probabilistic retrieval model, are used to find the similarity between the user query and the documents set in order to retrieve the documents that reflect and answer the user query. Vector space model can be implemented using one from the following well known similarity measures: Jaccard, Cosine, DICE, and Inner Product. Usually and for evaluation purposes; Precision and Recall are the measures that are widely used to evaluate the effectiveness of IR system.

A (GA) is an adaptive heuristic search algorithm which simulate the ideas from natural selection and genetics (Goldberg, 1989). Global solutions for many problems such as machine learning problems can be found using Genetic algorithms (GA).

In this paper, we will use the Cosine and Jaccard as similarity measures in the VSM since the DICE, and Inner Product similarity measures was studied by (Eman *et al*., 2011), for each similarity measure and based on using a combination of mutation ideas, fitness functions, crossover methods, we are going to create 15 different genetic algorithms alternatives. The idea behind this is to optimize the user query. As a test collection for this work; we are going to use an Arabic data collection with 242 documents and 59 queries, this collection was used by (Mohammad Othman Nassar, *et al*., 2011; Mohammad Othman Nassar *et al*., 2010; Eman *et al*., 2011).

Syntactic, morphologic, and semantic differences and difficulties of the Arabic language if compared to other languages was studied and discussed by many researchers such as (Mohammad Othman Nassar, *et al.*, 2011; Eman *et al*., 2011; Khoja, 2001; yahaya, 1989; Goweder and De Roeck, 2001). Arabic language if compared to English is usually more sparsed, this means that if we take the same text length from both languages, English words will be repeated more often than Arabic words (yahaya, 1989; Goweder and De

Corresponding Author: Mohammad Othman Nassar. Amman Arab University, Computer Information Systems department, Faculty Of Computer Sciences And Informatics. Amman, Jordan.
Phone numbers (00962788780593); E-mail: moanassar@yahoo.com.

Australian Journal of Basic and Applied Sciences, 7(13) November 2013, Pages: 66-72

Roeck, 2001). This fact can negatively affect the retrieval quality in Arabic language (Mohammad Othman Nassar, *et al*., 2011; Eman *et al*., 2011). Other differences within the Arabic language are related to the complexity of the Arabic roots, this is due to the existence of many forms for the same letter, and to the punctuation over the letters that may change the meaning of two identical words.

The uniqueness of the Arabic language, its differences from the English and other languages, and the absence of similar studies in the literature was our motivator to conduct this study based on Arabic data collection and using GA. We are going to use the same data collection and procedure as in (Eman *et al*., 2011), using the same data collection and procedure as in (Eman *et al*., 2011) will allow us to compare our results with their results.

Previous Studies:

GAs are known as a robust and powerful optimization techniques, and because of that many studies have been conducted using them such as (Mohammad Othman Nassar, *et al*., 2011; Hsinchun, 1995; D. Vrajitoru, 1998; Hananda, 2008; Vicente and Cristina, 2007; Rocio *et al*., 2005; Mercy and Naomie, 2005; Ahmed *et al*., 2006; Abdelmgeid, 2007; Fatemeh and Solmaz, 2010).

The authors in (Andrew T, 2004; Hsinchun, 1995; Hananda, 2008) presented many methods in the VSM, the methods included: the connectionist Hopfield network; the symbolic ID3/ID5R, symbolic ID3 Algorithm, Simulated Annealing, neural networks, evolution- based genetic algorithms, and genetic programming. All of the previous mentioned techniques was used to explore and analyze the user queries; they were promising in their ability to analyze the user queries, identifying the information needs for the users, and in suggesting alternatives for the search.

In (Rocio *et al*., 2005; Ahmed *et al*., 2006; Vicente and Cristina, 2007; D. Vrajitoru, 1998; Abdelmgeid, 2007) the VSM have been used, the idea is to improve the IR performance by creating different mutation probabilities, introducing new crossover operation, and using new fitness functions for the GA.

Mercy and Naomie (2005) propose a Genetic Algorithm (GA) based framework of data fusion based on linear combinations of retrieval status values; the authors framework was based on the Vector Space Model and the Probability Model. Their idea is to find the most excellent linear combination of weights that should be assigned to the scores of different retrieval systems to get the best possible retrieval performance.

Using GA to improve and enhance the performance of Arabic information system is rare in the available literature. As an example the authors in (Bassam *et al*., 2009) used the GA to enhance and improve the performance through the usage of a proposed adaptive matching function, which was created and obtained from four similarity measures which are: inner product, Cosine similarity, Jaccard and Dice.

Using GAs to improve the user query in the Vector space model and in the Boolean model was studied by (Mohammad Othman Nassar, *et al*., 2011; Eman *et al*., 2011) based on Arabic data collection. In (Eman *et al*., 2011) the researchers created and compared different fitness functions, mutations and crossover strategies to find the best mutation and crossover combination that can be used with the VSM Dice, and Inner Product similarity measures to improve the user query. This paper will study the VSM Cosine, and Jaccard similarity measures and compare them to the work of (Eman *et al*., 2011).

Vector Space Model (VSM):

In the VSM both documents and queries are represented as a multidimensional vectors, the vector should have dimensions; those dimensions are the terms. To identify the terms for the vectors we need what is called lexical scanning to be implemented, after that stemming process is applied to the words to get the stems, then the frequency of those stems is calculated. In the final stage the extracted query and the document vectors are compared using similarity measures such as (e.g. Cosine, DICE, Jaccard, Inner Product), Table 1 taken from (Eman *et al*., 2011) shows those similarity measures.

Table 1: Different Similarity Measures. Source (Eman *et al*., 2011).

Similarity Measure	Evaluation for Binary Term Vector	Evaluation for Weighted Term Vector
Cosine	$sim(d,q) = 2\frac{\lvert d \cap q \rvert}{\lvert d \rvert^{1/2} \bullet \lvert q \rvert^{1/2}}$	$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{t} w_{i,j}^2} \times \sqrt{\sum_{j=1}^{t} w_{i,q}^2}}$
Dice	$sim(d,q) = 2\frac{\lvert d \cap q \rvert}{\lvert d \rvert + \lvert q \rvert}$	$sim(d_j, q) = \frac{2\sum_{i=1}^{t} w_{i,j} \times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2 + \sum_{i=1}^{t} w_{i,q}^2}$

Australian Journal of Basic and Applied Sciences, 7(13) November 2013, Pages: 66-72

Jaccard	$sim(d,q)=\frac{\lvert d\cap q\rvert}{\lvert d\rvert+\lvert q\rvert-\lvert d\cap q\rvert}$	$sim(d_j,q)=\frac{\sum_{i=1}^{t} w_{i,j}\times w_{i,q}}{\sum_{i=1}^{t} w_{i,j}^2+\sum_{i=1}^{t} w_{i,q}^2-\sum_{i=1}^{t} w_{i,j}\times w_{i,q}}$
Inner Product	$\lvert d_i \cap q_k \rvert$	Sim= $\sum_{k=1}^{t}(d_{ik}\bullet q_k)$

Where $w_{i,j}, w_{i,q}$ are the weights of the i^{th} term in $document_j$, and in the query respectively.

Genetic Algorithm (GA):

We can use Genetic algorithms to generate new and better generations. The GA algorithm flowchart is illustrated in Figure 1 taken from (Eman *et al*., 2011). According to figure 1; Genetic algorithm operations includes three main operations: reproduction, crossover, and mutation. In reproduction; the best individuals are chosen based on the fitness function. In crossover; we exchange genes between two chromosomes. In Mutation; we randomly alter one or more genes in a particular chromosome based on certain probability. Mutation can be implemented using the Chromosomal mutation in which the genes can be changed and replaced with another genes completely.

Experiment:

In this study we will follow the same experimental procedure implemented by Mashkba (2009). with differences related to the similarity measures and mutation strategies. in similarity measures we used Cosine and Jaccard. In mutation strategies we used two point mutation strategies. In this study we used an initial population of 15 top documents retrieved from an traditional IR system that uses the VSM model which is built by (Hananda, 2008). In this study and to control the maximum number of generations for the GA we used is 75 iterations.

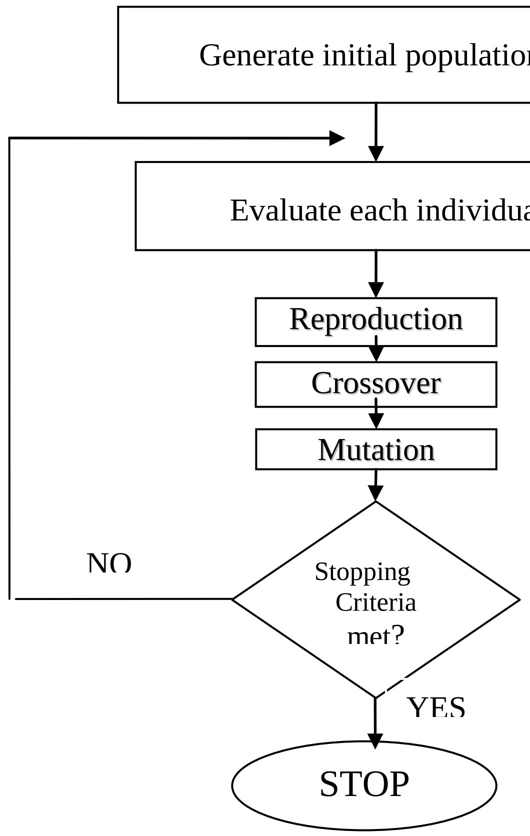

Fig. 1: Flowchart for Typical Genetic Algorithm (GA). Source (Eman *et al*., 2011)

Australian Journal of Basic and Applied Sciences, 7(13) November 2013, Pages: 66-72

The experiment started by extracting the significant terms from all Arabic documents; then they are assigned weights. Then a query vector is built using the binary weights for the terms and then it is adapted as a chromosome. Then the GA process is applied to optimize the query and to get global or near global query vector. Finally we compared the results for the 15 proposed GA methods with the result when using a traditional IR system; the traditional IR system that did not uses the GA to improve the user query. This study uses the jaccard and Cosine similarity measures as a fitness functions to evaluate the chromosomes. The chromosomes crossover operation in this study is used to produce the new offspring with crossover probability equals to 0.8. In this study five different crossover strategies were used for VSM, and they are: One-point (Ahmed *et al.*, 2006), dissociated crossover, uniform, fusion operator and restricted crossover (Vicente and Cristina, 2007).

Mutation is implemented on the chromosomes to create a random change in the chromosomes with a probability equals to 0.7. this experiment will used three different mutation strategies; one Point mutation with a probability equals to 0.7, two point mutation with a probability equals to 0.7, and Chromosomal mutation with a probability equals to 0.7. In Chromosomal mutation a complete chromosome is chosen and replaced. While in one point and two point mutations only selected number of genes (one or two) are changed based on the chosen probability.

Finally and for each similarity measure (Cosine, and Jaccard) we created 15 different GA strategies:

GA1: GA approach that is created using one-point crossover operator and point mutation.
GA2: GA approach that is created using one-point crossover operator and chromosomal mutation.
GA3: GA approach that is created using restricted crossover operator and point mutation.
GA4: GA approach that is created using restricted crossover operator and chromosomal mutation.
GA5: GA approach that is created using uniform crossover operator and point mutation.
GA6: GA approach that is created using uniform crossover operator and chromosomal mutation.
GA7: GA approach that is created using fusion operator and point mutation.
GA8: GA approach that is created using fusion operator and chromosomal mutation.
GA9: GA approach that is created using dissociated crossover and point mutation.
GA10: GA approach that is created using dissociated crossover and chromosomal mutation.
GA11: GA approach that is created using one-point crossover operator and two point mutation.
GA12: GA approach that is created using restricted crossover operator and two point mutation.
GA13: GA approach that is created using uniform crossover operator and two point mutation.
GA14: GA approach that is created using fusion operator and two point mutation.
GA15: GA approach that is created using dissociated crossover and two point mutation.

Ga Strategies Using Cosine Similarity:

The results for the GA strategies using cosine similarity are shown in Table 2 and Table 3. From those tables we can see that GA1, GA2, GA4, GA5, GA8, GA9 , GA10, GA11, GA14 and GA15

give an improvement over the traditional IR system with 12.42%, 6.96%, 7.39%, 5.41%, 7.99%, 7.26%, 4.53%, 6.73%, 6.25%, and 4.13% respectively, while GA3, GA6, GA7, GA12, and GA13 have no improvement over the traditional IR system with -1.36021%, -2.44788%, -1.26468%, -2.63%, and -2.87% respectively. This means that GA1 that use one-point crossover operator and point mutation gives the highest improvement over the traditional approach with 12.4245%.

Table 2: Average Recall and Precision Values for 59 Query by Applying GA's on Cosine Similarity.

Recall	Cosine	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12	GA13	GA14	GA15
0.1	0.132	0.165	0.151	0.133	0.135	0.15	0.133	0.13	0.135	0.137	0.141	0.156	0.128	0.130	0.131	0.144
0.2	0.14	0.164	0.157	0.135	0.16	0.166	0.141	0.138	0.162	0.163	0.151	0.159	0.129	0.137	0.158	0.149
0,3	0.147	0.182	0.165	0.142	0.175	0.151	0.144	0.15	0.179	0.164	0.152	0.170	0.139	0.145	0.171	0.151
0.4	0.151	0.166	0.167	0.149	0.161	0.149	0.15	0.146	0.167	0.167	0.159	0.155	0.145	0.151	0.164	0.159
0.5	0.156	0.179	0.172	0.153	0.178	0.172	0.152	0.152	0.177	0.179	0.171	0.172	0.150	0.152	0.179	0.168
0.6	0.178	0.191	0.18	0.172	0.188	0.181	0.164	0.176	0.188	0.187	0.179	0.181	0.173	0.169	0.189	0.182
0.7	0.183	0.193	0.181	0.181	0.193	0.181	0.181	0.179	0.189	0.188	0.19	0.185	0.182	0.179	0.190	0.189
0.8	0.234	0.244	0.239	0.236	0.231	0.241	0.222	0.23	0.231	0.232	0.24	0.230	0.239	0.218	0.225	0.237
0.9	0.241	0.251	0.243	0.243	0.242	0.244	0.231	0.242	0.242	0.244	0.243	0.239	0.245	0.231	0.237	0.239
Average	0.174	0.193	0.184	0.172	0.185	0.182	0.169	0.171	0.186	0.185	0.181	0.183	0.170	0.168	0.183	0.180

Ga Strategies Using Jaccard Similarity:

The results for the GA strategies using the Jaccard similarity are shown in Table 4 and Table 5. From those tables we can see that GA1, GA2, GA4, GA5, GA8, GA9, GA10, GA14, and GA15 give improvement over the traditional IR system with 3.79%, 12.48%, 7.65%, 8.64%, 7.81%, 7.91%, 9.92%, 5.56%, and 5.07% respectively. While GA3, GA6, GA7, GA11, GA12, and GA13 did not give any improvement over the traditional IR system with -1.20423%, -1.72545%, -3.85975%, -0.25%, -2.54%, and -4.64% respectively. This means that GA2 that use one-point crossover operator and chromosomal mutation gives the highest improvement over the traditional approach with 12.48%.

Table 3: GA's Improvement in Cosine Similarity (GA's Improvement %).

Recall	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12	GA13	GA14	GA15
0.1	25	14.39	0.76	2.27	13.64	0.76	-1.52	2.27	3.79	6.82	18.18	-3.03	-1.52	-0.76	9.09
0.2	17.14	12.14	-3.57	14.29	18.57	0.71	-1.43	15.71	16.43	7.86	13.57	-7.86	-2.14	12.86	6.43
0,3	23.81	12.24	-3.4	19.05	2.72	-2.04	2.04	21.77	11.56	3.4	15.65	-5.44	-1.36	16.33	2.72
0.4	9.93	10.6	-1.32	6.62	-1.32	-0.66	-3.31	10.6	10.6	5.3	2.65	-3.97	0	8.61	5.3
0.5	14.74	10.26	-1.92	14.1	10.26	-2.56	-2.56	13.46	14.74	9.62	10.26	-3.85	-2.56	14.74	7.69
0.6	7.3	1.12	-3.37	5.62	1.69	-7.87	-1.12	5.62	5.06	0.56	1.69	-2.81	-5.06	6.18	2.25
0.7	5.46	-1.09	-1.09	5.46	-1.09	-1.09	-2.19	3.28	2.73	3.83	1.09	-0.55	-2.19	3.83	3.28
0.8	4.27	2.14	0.85	-1.28	2.99	-5.13	-1.71	-1.28	-0.85	2.56	-1.71	2.14	-6.84	-3.85	1.28
0.9	4.15	0.83	0.83	0.41	1.24	-4.15	0.41	0.41	1.24	0.83	-0.83	1.66	-4.15	-1.66	-0.83
Average	12.42	6.96	-1.36	7.39	5.41	-2.45	-1.27	7.98	7.26	4.53	6.73	-2.63	-2.87	6.25	4.13

Table 4: Average Recall and Precision Values for 59 Query by Applying GA's on Jaccard Similarity.

Recall	Jaccard	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12	GA13	GA14	GA15
0.1	0.13	0.134	0.141	0.133	0.137	0.141	0.129	0.122	0.142	0.139	0.141	0.124	0.128	0.125	0.142	0.140
0.2	0.17	0.176	0.199	0.165	0.182	0.184	0.165	0.162	0.182	0.185	0.191	0.166	0.158	0.160	0.162	0.172
0,3	0.261	0.271	0.288	0.243	0.277	0.281	0.256	0.254	0.271	0.274	0.28	0.262	0.240	0.230	0.255	0.254
0.4	0.213	0.222	0.277	0.211	0.266	0.269	0.214	0.211	0.271	0.277	0.278	0.212	0.215	0.209	0.276	0.279
0.5	0.355	0.377	0.387	0.342	0.373	0.375	0.345	0.333	0.373	0.377	0.385	0.350	0.331	0.345	0.363	0.360
0.6	0.335	0.343	0.401	0.341	0.38	0.384	0.323	0.311	0.382	0.381	0.386	0.337	0.341	0.315	0.384	0.371
0.7	0.385	0.399	0.398	0.381	0.385	0.387	0.371	0.362	0.382	0.359	0.381	0.385	0.378	0.360	0.389	0.361
0.8	0.389	0.401	0.415	0.392	0.406	0.41	0.385	0.389	0.407	0.411	0.414	0.393	0.385	0.380	0.400	0.405
0.9	0.434	0.452	0.467	0.433	0.445	0.438	0.437	0.43	0.434	0.441	0.441	0.454	0.437	0.431	0.423	0.427
Average	0.297	0.308	0.330	0.293	0.317	0.319	0.291	0.286	0.316	0.316	0.322	0.298	0.290	0.284	0.310	0.308

Table 5: GA's Improvement in Jaccard Similarity (GA's Improvement %)

Recall	GA1	GA2	GA3	GA4	GA5	GA6	GA7	GA8	GA9	GA10	GA11	GA12	GA13	GA14	GA15
0.1	3.08	8.46	2.31	5.38	8.46	-0.77	-6.15	9.23	6.92	8.46	-4.62	-1.54	-3.85	9.23	7.69
0.2	3.53	17.06	-2.94	7.06	8.24	-2.94	-4.71	7.06	8.82	12.35	-2.35	-7.06	-5.88	-4.71	1.18
0,3	3.83	10.34	-6.9	6.13	7.66	-1.92	-2.68	3.83	4.98	7.28	0.38	-8.05	-11.88	-2.3	-2.68
0.4	4.23	30.05	-0.94	24.88	26.29	0.47	-0.94	27.23	30.05	30.52	-0.47	0.94	-1.88	29.58	30.99
0.5	6.2	9.01	-3.66	5.07	5.63	-2.82	-6.2	5.07	6.2	8.45	-1.41	-6.76	-2.82	2.25	1.41
0.6	2.39	19.7	1.79	13.43	14.63	-3.58	-7.16	14.03	13.73	15.22	0.6	1.79	-5.97	14.63	10.75
0.7	3.64	3.38	-1.04	0	0.52	-3.64	-5.97	-0.78	-6.75	-1.04	0	-1.82	-6.49	1.04	-6.23
0.8	3.08	6.68	0.77	4.37	5.4	-1.03	0	4.63	5.66	6.43	1.03	-1.03	-2.31	2.83	4.11
0.9	4.15	7.6	-0.23	2.53	0.92	0.69	-0.92	0	1.61	1.61	4.61	0.69	-0.69	-2.53	-1.61
Average	3.79	12.48	-1.2	7.65	8.64	-1.73	-3.86	7.81	7.91	9.92	-0.25	-2.54	-4.64	5.56	5.07

Table 6: Comparison Between the Best GA Strategies (Each Similarity Measures).

Recall	Cosine(GA1)	Jaccard(GA2)	Dice(GA9)	Inner Product(GA1)
0.1	0.165	0.141	0.141	0.146
0.2	0.164	0.199	0.197	0.208
0,3	0.182	0.288	0.298	0.301
0.4	0.166	0.277	0.277	0.283
0.5	0.179	0.387	0.402	0.405
0.6	0.191	0.401	0.408	0.409
0.7	0.193	0.398	0.396	0.413
0.8	0.244	0.415	0.412	0.437
0.9	0.251	0.467	0.441	0.487
Average	0.193	0.330	0.330222	0.343222

Comparison Between The Best Ga's Strategies:

To create a detailed and useful comparison we will bring the results for the DICE and Inner Product from (Eman *et al*., 2011), and put them with our results for the Jaccard and Cosine. Table 6, and Figure 2 show the comparison between Cosine (GA1), Jaccard(GA2), Dice(GA9) and Inner Product (GA1). It is clear that we used only the best GA strategy for each similarity measure (Cosine, DICE, Jaccard, Inner Product) in the VSM. From this table we notice that the Inner Product(GA1) represent the best strategy over Cosine(GA1), Jaccard(GA2) and Dice(GA9). Which means that Inner Product(GA1) that use one-point crossover operator, point mutation, and Inner Product similarity as a fitness function represent the best IR system in VSM to be used with the Arabic data collection. Figure 2 also present the data in the table 6.

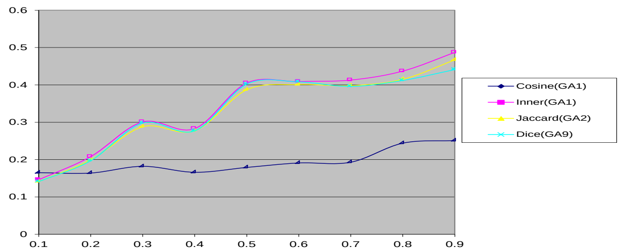

Fig. 2: Comparison Between the Best GA Strategies (Each Similarity Measures).

Conclusions:

For each similarity measure (Cosine, Jaccard) in the VSM we proposed then compared 15 different GA approaches, and by calculating the improvement of each approach over the traditional IR system, we noticed that the proposed approaches in general can improve the user query if compared to the mentioned traditional IR system that did not uses the GA. As a recommendation and based on the comparison conducted in table 6, we recommend the GA approach which uses one-point crossover operator, point mutation, and Inner Product similarity as a fitness function to be used with the Arabic data collections.

REFERENCES

Abdelmgeid, A., 2007. "Applying Genetic Algorithm in Query Improvement Problem", International Journal "Information Technologies and Knowledge, 1: 309-316.

Ahmed, A.A., Radwan, Bahgat A. Abdel Latef, Abdel Mgeid A. Ali, Osman A. Sadek, 2006. "Using Genetic Algorithm to Improve Information Retrieval Systems", proceedings of world academy of since, engineering and technology, 17: ISSN 1307-6884.

Andrew, T., 2004. "an Artificial Intelligence Approach to Information Retrieval", Information Processing and Management, 40(4): 619-632.

Bassam Al-Shargabi, Islam Amro, and Ghassan Kanaan, 2009. "Exploit Genetic Algorithm to Enhance Arabic Information Retrieval", 3rd International Conference on Arabic Language Processing (CITALA’09), Rabat, Morocco, pp: 37-41.

Vrajitoru, D., 1998. “Crossover improvement for the genetic algorithm in information retrieval”, Information Processing& Management, 34(4): 405-415.

Eman Al Mashagba, Feras Al Mashagba, and Mohammad Othman Nassar, 2011. "Query Optimization Using Genetic Algorithms in the Vector Space Model," International Journal of Computer Science Issues (IJCSI), ISSN (online): 8(5): 1694-0814.

Fatemeh Dashti, and Solmaz Abdollahi Zad, 2010." Optimizing the data search results in web using Genetic Algorithm", international journal of advanced engineering and technologies, 1(1): 016- 022, ISSN: 2230-781.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley.

Goweder, A., A. De Roeck, 2001. "Assessment of a Significant Arabic Corpus", Arabic Natural Language Processing Workshop (ACL2001), Toulouse, France. Downloaded from: (http://www.elsnet.org/acl2001 arabic.html).

Hananda, E., 2008. "Evaluation of Different Information Retrieval models and Different indexing methods on Arabic Documents", Phd Thesis, ARAB Academy.

Hsinchun, C., 1995. "Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms", Journal of the American Society for Information Science, 46(3).

Khoja, S., 2001. "APT:Arabic part-of-speech tagger", proceedings of the student workshop at second meeting of north American chapter of Association for Copmputational Linguistics (NAACL2001), Pittsburgh, Pennsylvania, pp: 20-26.

Mashkba, F., 2009. "Evaluate the Effectiveness of Genetic Algorithm (GA) in Information Retrieval Based on Arabic Documents ", Phd Thesis, Arab Academy.

Mercy, T., S. Naomie, 2005. "A Framework for Genetic-Based Fusion of Similarity Measures In Chemical Compound Retrieval", International Symposium on Bio-Inspired Computing, Puteri Pan Pacific Hotel Johor Bahru, 5-7.

Mohammad Othman Nassar, Feras Al Mashagba, and Eman Al Mashagba, 2011. "Improving the User Query for the Boolean Model Using Genetic Algorithms," International Journal of Computer Science Issues (IJCSI), ISSN (online): 8(5): 1694-0814.

Mohammad Othman Nassar, Ghassan Kanaan and Hussain A.H. Awad, 2010. “Comparison between different global weighting schemes,” Lecture Notes in Engineering and Computer Science journal, ISSN: 2078-0966 (online version); 2078-0958 (print version), 2180; Issue: 1; pp: 690-692; Date: 2010; published by Newswood Limited.

Rocio, C., M. Carlos Lorenzetti, Ana, B. Nelida, 2005. "Genetic Algorithms for Topical Web Search: A Study of Different Mutation Rates", ACM Trans. Inter. Tech., 4(4): 378-419.

Tengku, M.T., C.J. Sembok and van Rijsbergen, 1990. "A simple logical-linguistic document retrieval system", Information Processing & Management, 26(1): 111-134.

Vicente, P., P. Cristina, 2003. "Order-Based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback", Journal Of The American Society For Information Science And Technology, 54(2): 152-160.

yahaya, A., 1989. "on the Complexity of the initial stage of Arabic text processing", First Great Lakes Computer Science Conference, Kalamazoo, Michigan, USA.