IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 5, Ver. I (Sep. – Oct. 2015), PP 07-17
www.iosrjournals.org

Increasing the Visibility of Search using Genetic Algorithm

Jaswinder Singh[1], Parvinder Singh[2], Yogesh Chaba[3]
[1,3]*(Department of Computer Science & Engg., Guru Jambheshwar University of Science & Technology),Hisar, Haryana, India.*
[2]*(Department of Computer Science & Engg., Deenbandhu Chhotu Ram University of Science & Technology), Sonepat, Haryana, India.*

Abstract : *The vast repository of informational databases i.e. Web is available to the user in the form of textual documents. It's a challenge to develop an effective information retrieval approach that can ease the user search and increases the visibility of search. Genetic Algorithm based approach has been implemented to increase the visibility of search by expanding the query using Jaccard similarity function as fitness function. The step by step implementation of genetic algorithm for one generation has been explained in the paper and the experiment was repeated for 500 generations to obtain optimum keywords out of which the best suited keyword was considered for expanding the query. The effectiveness of the approach has been experimentally evaluated on manually created training data of retrieved documents for formulated queries using the Google search engine.*
Keywords: *Information Retrieval, Search Engine Visibility, Similarity Function, Genetic Algorithm*

I. Introduction

The basic and ultimate goal of the information retrieval system is to deliver the most similar documents that have the potential to satisfy the user's need and the success of information retrieval system depends on the ability to assess the relevance of objects in its database i.e. information units, documents, functions, commands etc. to the given user's request [1] . With the increase in the content of the information on the internet it is difficult for the user to get the relevant information when a query of two or three words is usually typed by the user for searching any information of interest from the web world. These short queries and the incompatibility between the terms of queries and the documents affect the relevancy of the retrieved documents. When user enters his request in the form of query then the matching mechanism of the search system delivers the ranked list of documents to the user using the similarity functions. The documentary database, query subsystem and matching mechanism are three basic components of information retrieval system [2] [3] [4]. The similarity measurement between the different objects is the fundamental function of any information retrieval application and there are varieties of ways to compute the similarity among the different object representations. Textual similarity functions play a vital role in tasks and applications of information retrieval i.e. document clustering, topic detection, question answering, machine translation, text classification and others. Textual similarity can be measured lexically and semantically. If the words have similar character sequence then they are said to be lexically similar but if they are used in the same context one is type of another then they are said as semantically similar. If the user is not satisfied with the results returned by the search system then user reformulates the query there by increasing the retrieval effectiveness iteratively and incrementally [3].The user evaluates the results on the basis of retrieved documents and provides the relevant feedback for the expansion of terms of initial query. This feedback can be used to increase the effectiveness of the retrieval system. Query expansion is a technique used to increase the effectiveness of the information retrieval [4]. It is the process of adding some additional terms or phrases to the original query to improve relevancy of the retrieved documents. The reformulated query contains more terms so the probability of matching them with terms in relevant documents is therefore enhanced. The key problem of query expansion is the selection of additional terms based on which user's original query is enhanced. Initial query can be expanded in three different ways i.e. manual, interactive and automatic. In case of the former two, the user's involvement is required where as the user's intervention is not required in case of the automatic query expansion techniques. This paper focuses on the formulation of appropriate query terms for text based search that can result in the increased relevancy of the retrieved textual documents using genetic algorithm.

This paper is organized as follows. The first section of paper is related with the brief introduction regarding the effectiveness of the information retrieval system. The second section of paper is related to the literature on applicability of the genetic algorithm in information retrieval and related work. The third section of the paper describes the detailed description of the experiment followed for the implementation of genetic algorithm for the improvement of the relevancy of the retrieved documents by the addition of term in the original query. Fourth section of paper describes the conclusion.

II. Genetic Algorithm In Information Retrieval And Related Work

One of the most important methods for modifying the query is the relevance feedback. The main feature of this technique is to use the information retrieved from the retrieved documents which are identified as relevant to revise the query. Then the query is modified and executed and new document set is returned .The new set contain the documents from the original set although these can appear at the different rank orders [5]. V. N. Gudivada *et al.* [4] categorized the relevance feedback as the positive feedback and negative feedback based on the set of documents retrieved. The set of documents deemed relevant to the user constitute a positive feedback and the set of documents deemed irrelevant constitute negative feedback. The results of the positive feedback were found more effective because the documents in the positive feedback set are more similar than the documents in the negative feedback. To improve the efficiency and effectiveness of the ranked outputs the information retrieval systems make use of document clustering. Documents relevant to a query tend to be highly similar in the context defined by the query. The pair of documents has an overall similarity and a specific similarity. So a Query-Sensitive Similarity Measure mechanism was proposed to measure the similarity of two documents for a given query [11]**.** Genetic algorithms are robust and efficient search and optimization techniques developed in 1960s, inspired by the Darwin's theory of natural evolution. Genetic algorithm searches the space by iterating steps like fitness evaluation, selection, crossover and mutation. The search space of document search represents the high dimensional search space and genetic algorithm is an optimization technique which can be used to search relevant information from the document search space. Genetic Algorithm is one of the powerful searching mechanism known for its robustness and quick search capabilities and so it is suitable for the information retrieval [8][13][17].To use genetic algorithms in the problems of optimization, there is need to define the candidate solutions by the chromosomes consisting of genes and fitness function. A population of candidate solution is maintained and the ultimate goal is to obtain the better solutions after some generations. Genetic operators (selection, crossover and mutation) are applied to produce the new generation. Parents are selected to produce the offspring, favoring those parents with more values of fitness function. Crossover of the population members takes place by exchanging the subparts of the parent chromosomes. Mutation takes place by flipping of genes. However a simple genetic algorithm [9] [22] [23] make use of the following steps.

Step1: Random generation of population
Step2: Fitness evaluation of each individual in the population
Step3: Selection of individuals to reproduce on the basis of fitness.
Step4: Apply Crossover
Step5: Apply Mutation
Step6: Replacement of population by new generation
Step7: Go to step 2.

Lopez-Pujalte *et al.* [6] implemented and compared the different applications of GA to relevance feedback and concluded that the design of fitness function was fundamental to modify the query and further evaluated the efficiency of genetic algorithm with order-based fitness functions for relevance feedback [7]. Rocio L. Cecchini *et al.* [9] describes the optimization techniques based on Genetic Algorithms to evolve the query terms in the context of given topic and studied the effect of the different mutation rates. Poltak Sihombing *et al.* [10] implemented the genetic algorithm using the Horng and Yeh's coefficient as fitness function and fitness of each chromosome was selected on the basis of score of coefficient. Ahmad *et al.* [12] introduced the new fitness function and compare the result based on the Cosine fitness function and classical information retrieval in query learning problems using the genetics techniques. S.S. Sathya *et al.* [13] proposed a retrieval system that uses the two stage approach that uses genetic algorithm to obtain the combination of terms in the first stage and the second stage uses the output from the first stage to retrieve the relevant results. Tournament selection was used as the selection process and single point crossover was used. Different authors used Genetic Algorithm in information retrieval in different ways.

From the literature it was found that the relevance feedback is important technique to modify the query and this technique uses the documents returned by the system when the query is entered by the user in the search system. Relevance feedback introduces many design issues. In the literature, Genetic Algorithm was used by many authors to modify the query and in the literature related to the applications of Genetic Algorithm in information retrieval it was found that the design of the fitness function is the fundamental for the genetic algorithm to modify the query optimally. The work in this paper focuses on how to formulate appropriate query terms for text based search that can result in the increased relevancy of the retrieved textual documents. It was found from the literature that there are similarity functions i.e. Jaccard, Dice, Cosine and Overlap which are used in the field of information retrieval [20] and all of these are term based similarity functions [16]. Similarity function measures the degree of similarity between two sub sets X and Y of the entire data base of the documents in the repository.

X is defined, a set of all terms occurring in document X

Y is set of all terms occurring in document Y.

|X| = Numbers of terms that occur in set X.

|Y| = Number of terms that occur in set Y.

$|X \cap Y|$ =Number of terms occur in both X and Y.

In this work Jaccard similarity function is used as the fitness function and the step by step implementation of Genetic Algorithm is explained in the next section. The formula of Jaccard coefficient was given by Paul Jaccard [14]. Jaccard similarity function is defined as the number of shared terms over the number of unique terms in both strings [16][18][19]. String similarity functions are further categorized as character based and the term based similarity function and Jaccard similarity function is a term based similarity function [16]. Jaccard similarity function measures the similarity between the two objects between 0.0 and 1.0 For X and Y subsets of documents retrieved from the entire repository of documents. Then Jaccard similarity i.e. J(X, Y) between the set of terms of document X and set of terms of document Y is defined as is the number of terms that occurs in both the documents to the total number of terms present.

$$\frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

If we have two sets of documents and each document has term present or absent in the document is represented by value 0 and 1 as shown below. Then Jaccard similarity between the two sets of documents is measured as 2/ (3+3-2) = 2/4 = 0 .5.

X:	1	1	1	0	0
Y:	1	1	0	1	0

The above set theoretic definition is true for vector definition [26].If vector X = (x_1, x_2... x_n) and another vector Y= (y_1, y_2, …y_n) where x_1, x_2,.. x_n and y_1, y_2,..y_n are the weights of the term i.e. presence and absence of term. Binary term vectors have been used in the experiment in which weight of term is taken as one if the term is present in the document and zero if the term is absent in the document. The Jaccard similarity i.e. J(X, Y) between vector X and vector Y is represented by the following formula.

$$= \frac{\sum_{i=1}^{n} xi.yi}{\sum_{i=1}^{n} xi^2 + \sum_{i=1}^{n} yi^2 - \sum_{i=1}^{n} xi.yi}$$

III PROCESS OF EXPERIMENT

The following process was followed in the experiment to increase the visibility of the search engine by the enhancing the query by addition of terms in the original query.

1. Measurement of the relevancy of retrieved documents using Jaccard similarity function.
2. Implementation of Genetic Algorithm to optimize terms
3. Measurement of relevancy of documents after addition of new term.

The experiment was performed by preparing the training data using the Google search engine. The training data includes the queries used for the experiment and the text of the retrieved documents using the search engine. Top ten documents were considered for the experimentation as it very difficult to look at the entire retrieved documents. In the work positive feedback method has been used which is based on the assumption that when query is sent to search system then first ten documents are considered as the relevant set of documents and rest are the irrelevant. Relevancy of the retrieved documents in terms of similarity between the documents was measured using the similarity functions. In the experiment, Jaccard similarity was used to measure the similarity between the documents for the entered query using the vector space model. In the experiment binary term vectors were used i.e. the weight of the term is taken as 1 if the term is present in the document and 0 if the term is not present in the document.

The first step of the experiment is to measure the relevancy of retrieved documents using the Jaccard similarity function and the results obtained are shown in table 11. The detailed process of similarity measurement using Jaccard similarity function using the same training data is described in [21].The second step is to implement Genetic Algorithm to enhance the visibility of search and it was implemented using Jaccard similarity functions as the fitness function. With the help of new keyword as third part of the experiment the similarity improvements in terms of relevancy of the documents in the percentage was calculated and results are shown in table 10 . The complete step by step implementation of genetic algorithm using Jaccard similarity function is explained in the following sections as different steps.

Step1. Extraction of Key words from the retrieved documents for formulated query

The keyword set of 25 terms representing all the ten retrieved documents was formed using the Textalyser tool [25] for the entered query i.e. “Terrorist Attack Mumbai” and following keyword set is obtained by the analysis

of text contained in the documents [21] i.e.{Afzal, Attack, Bandra, Blast, Bomb, Case, Friday, Government, Headly, Hillary , India, Injured, Intelligence, Juhu, Kasab, Killed, Maharashtra, Minister, Mumbai, Pakistan, People, Police, Rana, Taj, Terrorist}

Step2: Encoding of Documents for the formulated query

As Genetic Algorithm is to be applied so the above set of documents is represented as strings of 0's 1's i.e. if the term of the keyword set is present in the text of documents then 1 is placed otherwise 0 was placed. The frequency of terms is not considered in the experiment for the simplicity of the experiment. The chromosomes are thus formed and these are represented as C_1, C_2, C_3… C_{10}. The population of size of 10 chromosomes was created and the length of the chromosomes was taken as 25 because the size of the keyword set is 25. In this process the genes of the chromosomes is fixed but it can vary according to the query that is processed and documents retrieved.

C_1 = (0100000000100010001101010)

C_2 = (0000000100100011000010001)

C_3 = (0101000000110000111001000)

C_4 = (0100000000000010010000000)

C_5 = (0100000000100000001010001)

C_6 = (0100000011100000001000101)

C_7 = (0101001010100000001100001)

C_8 = (1110000000000100001001000)

C_9 = (0100010010100000001000001)

C_{10} = (0000100000001000000001001)

Step3: Implementation of Selection, Crossover and Mutation Operators

The new generation in the Genetic Algorithm is determined by applying the genetic operators i.e. selection, crossover and mutation on the chromosomes of the current population. The experiment was performed first for the one generation and then run for 500 generations by setting the Genetic Algorithm parameters which are shown in table 1.

TABLE 1: Parameters chosen for experiment

Parameters	Value/method
Size of Population	10
Length of Chromosome	25
Probability of Crossover	0.5 and other values
Probability of Mutation	0.001and other values
Selection Process	Roulette wheel
Crossover process	One point Crossover
Mutation Process	Bit Flipping

Selection Operator:

Genetic Algorithm uses simple random sampling [6] [24] as selection mechanism. For the selection process, a roulette wheel was constructed which is implemented by assigning each chromosome a selection probability equal to its fitness value divided by the sum of the fitness values of all the chromosomes [24] .With the roulette wheel implemented in this manner selection mechanism is to spin the wheel population size times each time selecting a chromosome for intermediate population. The process of construction of roulette wheel for selection includes the following steps.

- Calculation of the fitness value for each chromosome.
- Calculation of the total fitness of the population.
- Calculation of the probability of selection i.e. p_i for each chromosome C_i (i = 1, 2... population size).
- Calculation of the cumulative probability qi for each chromosome C_i (1=1, 2…… population size)

This process of selection is based on the spinning of roulette wheel up to population size times [24]. In the experiment the size of population is 10 as there are 10 chromosomes. A single chromosome was selected for a new population in the following way every time.

- Random number, r which is float is generated in the range of [0, 1].
- If random number $<q_1$ then select the first chromosome otherwise select the i^{th} chromosome.

Step3.1 : Description of selection process for one generation of Genetic Algorithm in the experiment for query "Terrorist Attack Mumbai".

The selection mechanism explained above was implemented using the Jaccard similarity function and the complete process of selection is explained as below.

- **Calculation of fitness value for each chromosome**

In the experiment, according to the step explained above the fitness value of each chromosome was calculated using the Jaccard similarity function and then average was taken as shown below in table 2.The fitness was calculated using Jaccard similarity function for the query "Terrorist Attack Mumbai".

TABLE 2: Fitness value using Jaccard similarity function for query "Terrorist Attack Mumbai".

Chromo some no.	Fitness value with Jaccard Similarity Function for one Generation of Genetic Algorithm										Avg (f1).
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	
C_1	1	0.1818	0.3636	0.25	0.3333	0.2727	0.3636	0.3	0.3	0.1	0.3465

- **Calculation of the total fitness for population**

In the similar way the fitness value for each chromosome i.e. 10 chromosomes was calculated and average is taken and all these ten fitness values for each chromosome are shown in the table below. If f_1, f_2, f_3.....f_{10} are the ten fitness values for the chromosome C_1, C_2….C_{10} respectively and these measured values are shown in table 3.

TABLE 3: Total fitness value using Jaccard Similarity Function for query "Terrorist Attack Mumbai"

Fitness value with Jaccard Similarity Function for whole population									
f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
0.3465	0.2418	0.3182	0.2201	0.4014	0.3722	0.3721	0.2579	0.3960	0.1840

Then the sum of all these values is considered to calculate the total fitness of the population. It is clear that the chromosome 5^{th} is the strongest as it has the highest fitness value and chromosome 10^{th} is weakest one. Now the system constructs a roulette wheel for the selection process and the total fitness of the population is calculated as sum which is equal to 3.1106 i.e. the sum of the above ten fitness values of table 3.

- **Calculation of the probability of selection**

The probability of selection of each chromosome was calculated. The probability of selection as explained above is the fitness of each chromosome divided by the total fitness value of the population. If p_1 is the probability of the selection of chromosome number one i.e. C_1 which is obtained by dividing the average fitness value of chromosome no one by the total fitness value of the population which is 3.1106 in this case. Then p_1= 0.3465/3.1106 i.e. 0.114.Similarly the probability of selection of all the ten chromosomes was calculated. The probability of selection i.e. p_i for each chromosome C_i (i =1, 2 ...10) is shown in table 4.

TABLE 4: Probability of Selection of Each Chromosome

Probability of selection i.e. p_i of each chromosome C_i(i=1,2 ...10)
p_1= 0.3465 /3.1106 = 0.1114
p_2 = 0.2418/3.1106 = 0.0777
p_3 = 0.3182/3.1106 = 0.1023
p_4 = 0.2201/3.1106 = 0.0708
p_5 = 0.4014/3.1106 = 0.1291
p_6 = 0.3722/3/1106 = 0.1197
p_7 = 0.3721/3.1106 = 0.1196
p_8 = 0.2579/3.1106 = 0.0829
p_9 = 0.3960/3.1106 = 0.1273
p_{10} = 0.1840/3.1106 = 0.0592

- **Calculation of cumulative probability**

The cumulative probability for each chromosome was calculated. If p_1 is the probability of the selection of chromosome number one then the cumulative probability of chromosome number one is same as p_1.The cumulative probability of second chromosome is obtained by adding q_1 to p_2. In this way the cumulative probability for each chromosome was calculated. The cumulative probability q_i for each chromosome C_i (i=1, 2 ...10) are shown in table 5.

TABLE 5. Cumulative Probability of Each Chromosome

The cumulative probability q_i for each chromosome C_i (i=1, 2 ...10)
q_1= 0.1114
q_2= 0.1891
q_3 = 0.2915

$q_4 = 0.3622$
$q_5 = 0.4913$
$q_6 = 0.6109$
$q_7 = 0.7306$
$q_8 = 0.8135$
$q_9 = 0.9408$
$q_{10} = 1$

- **Generation of Random Numbers:**

Roulette wheel was spun 10 times as the size of population is 10 i.e. the ten chromosomes were used in the experiment and ten float random numbers [21] are generated in the range [0, 1]. On each run new numbers were generated. In our case these ten random numbers are 0.9501, 0.2311, 0.6068, 0.486, 0.8913, 0.7621, 0.4565, 0.0185, 0.8214, and 0.4447. These generated random numbers were compared with the cumulative probability of the table 5. The first generated random number r = 0.9501 was compared and was found less than cumulative probability of tenth chromosome i.e. q_{10} and so chromosome 10^{th} was selected for the new population because the cumulative probability of the 10^{th} chromosome is more than the first generated random number 0.9501.The second number r = 0.2311 is greater than the q_2 but less than q_3, means that the 3rd chromosome was selected for the new population. So we had the chromosomes which were selected for the new populations as shown below.

C_{10}, C_3, C_6, C_5 , C_9, C_8, C_5, C_1, C_9, C_5.

New population of chromosomes after the selection process

After the selection process of the algorithm, new population consists of chromosomes as C_1', C_2'… C_{10}'. So we had C_1' as new selected chromosome which was the 10^{th} chromosome of the initial population i.e. C_{10}. C2' as new selected chromosome which was the 3^{rd} chromosome of initial population and in this way we obtained the following new population of chromosomes.

C_1'= (0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1)(C_{10})

C_2'= (0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 0 0 0)(C_3)

C_3'= (0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1)(C_6)

C_4'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1)(C_5)

C_5'= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1)(C_9)

C_6'= (1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0)(C_8)

C_7'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1)(C_5)

C_8'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 1 0 1 0)(C_1)

C_9'= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1)(C_9)

C_{10}'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 1)(C_5)

Crossover Operator: One of the parameter of the parameter of the genetic system is the probability of crossover i.e. P_c .This probability of crossover gives us the expected number which undergoes the crossover operation. The following mechanism was followed for the crossover operation.

- Generation of random numbers (float) r in the range [0, 1].
- If random number < probability of crossover then select the given chromosome for crossover.

Now the selected chromosomes randomly mated and for each pair of the coupled chromosomes the random integer number in the range [0, m-1] is generated. Where m is the total length of chromosome and crossover point was found. One point crossover method [24] was used in the experiment.

Step 3.2: Description of crossover process for one generation of Genetic Algorithm in the experiment for query "Terrorist Attack Mumbai".

The recombination operator, crossover was applied to the individuals in the new population. As in the experiment the probability of crossover $P_c = 0.5$, so it is expected that on average 50% of chromosomes will undergo crossover. The following method was used and for r<0.5 we select the given chromosome for crossover.

Two types of random number were generated. One random number was generated for the crossover point and this generated number has the integer value let this be ra. In the experiment we obtained random number (ra) i.e. 16, which is even. If the number of selected chromosome was odd then one extra chromosome was to be added or removed [24]. Another 10 random numbers were generated i.e. ran, If the sequence of random number is like 0.7919, 0.9218, 0.7382, 0.1763, 0.4057, 0.9355, 0.9169, 0.4103, 0.8936, 0.0579. This sequence of random number is compared with probability of crossover (P_c) i.e. the chromosomes with less than P_c value i.e. (0.5) are selected for crossover. This means that chromosomes 4^{th}, 5^{th}, 8^{th} and 10^{th} of the selected

population are selected for crossover. The selected pair of chromosomes i.e. (C_4', C5') and (C_8', C10') of the selected population are considered for the crossover and the crossover point i.e.16 are shown below.

C_4'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 |0 0 1 0 1 0 0 0 1) (C_5)

C_5'= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 |0 0 1 0 0 0 0 0 1) (C_9)

C_8'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 |0 0 1 1 0 1 0 1 0) (C_1)

C_{10}'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0| 0 0 1 0 1 0 0 0 1) (C_5)

Now the randomly selected chromosomes i.e. 4th and 5th and next two i.e. 8th and 10th are mated. The first pair of chromosome is the C_4' and C5' of new population which was obtained after the selection process which is same as the chromosome C_5 and C_9 of the old population.

C_4'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 |0 0 1 0 1 0 0 0 1) (C5)

C_5'= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 |0 0 1 0 0 0 0 0 1) (C9)

These chromosomes are cut after the 16th bit as the crossover point and replaced by pair of their offspring. Chromosomes C4'' and C5''are obtained after the crossover and are shown below.

C_4''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 |0 0 1 0 0 0 0 0 1) (after crossover)

C_5''= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 |0 0 1 0 1 0 0 0 1) (after crossover)

The second pair of chromosomes i.e. C_8' and C_{10}' are selected for the crossover

C_8'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 |0 0 1 1 0 1 0 1 0) (C1)

C_{10}'= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0| 0 0 1 0 1 0 0 0 1) (C5)

These chromosomes were cut after 16th bit and replaced by pair of their offspring's and chromosomes C8'' and C10''are obtained after the crossover and are shown below.

C_8''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 |0 0 1 0 1 0 0 0 1) (after crossover)

C_{10}''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0|0 0 1 1 0 1 0 1 0) (after crossover)

New population after applying the crossover

This new population obtained after the crossover is same as the population obtained after the selection process except the chromosomes C_4'' i.e. 4th chromosome, C_5'' i.e. 5th chromosome, C_8'' i.e. 8th chromosome and C_{10}'' i.e. 10th chromosome as shown below. C_1', C_2', C_3', C_6', C_7', C_9' shows the same chromosomes as was obtained in the selection process after applying the crossover operator but C_4'', C_5'', C_8'' and C_{10}'' are the chromosomes obtained after the application of crossover operator at the crossover point i.e. 16 in the present case.

C_1'= (0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1) (C10)

C_2'= (0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 0 0 0) (C3)

C_3'= (0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1) (C6)

C_4''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 |0 0 1 0 0 0 0 0 1)

C_5''= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 |0 0 1 0 1 0 0 0 1)

C_6' = (1 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0) (C8)

C_7' = (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1) (C5)

C_8''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 1 0 |0 0 1 0 1 0 0 0 1)

C_9'= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1) (C9)

C_{10}''= (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0|0 0 1 1 0 1 0 1 0)

Mutation Operator:

In the experiment mutation is implemented as a random process [24]. Another parameter of the genetic system is the probability of mutation and this probability gives the expected number of mutated bits and each bit has the equal chance to go for the mutation i.e. change from 0 to 1 or 1 to 0. The mechanism of mutation process was followed as described below.

- Generation of a random number (float) in range [0, 1].
- If random number is less than probability of mutation then mutation of bit take place.

Step3.3: Description of mutation process for one generation of Genetic Algorithm in the experiment for query "Terrorist Attack Mumbai".

Mutation operator was performed on bit- by-bit basis. The probability of mutation is P_m=0.001 i.e. so it was expected that 1% of bit undergo mutation. There are 25x10=250 bits in the whole population. So for every bit in the population a random number in the range [0, 1] was generated, if random number is less than 0.001 then mutation of the bit take place. In this way we have generated 250 random numbers. In the run it was found that three of these numbers were smaller than 0.001. The bit number and random number which were smaller than the probability of mutation are shown in table 6.

TABLE 6. Bit Position and Random number

Bit position	Random number
120	0.0003
138	0.0004
145	0.0001

The table 7 translates the bit position into the chromosome number and the bit number with in the chromosome.

TABLE 7. The bit position, Chromosome Number and Bit number within Chromosome

Bit Position	Chromosome number	Bit number within chromosome
120	5^{th}	20
138	6^{th}	13
145	6^{th}	20

New population after applying mutation is same as that obtained after crossover but the 120^{th} bit, 138^{th} bit, and 145^{th} bit changed due to mutation as shown below in bold and mutated chromosome is shown as C_5''' i.e. 5^{th} chromosome and C_6''' i.e. 6^{th} chromosome. C_1', C_2', C_3', C_4'', C_7', C_8'', C_9', C_{10}'' are chromosomes which are same as obtained after crossover on applying the mutation but C_5''', C_6''' are the mutated chromosomes with mutated bit as 120^{th} bit, 138^{th} bit, 145^{th} bit which are replaced as 1 in place of 0 as shown below.

C_1'= (0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1) (C_{10})

C_2' = (0 1 0 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 0 0 0) (C_3)

C_3' = (0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0 1) (C_6)

C_4'' = (0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1)

C_5'''= (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 **1** 1 0 0 0 1)

C_6'''= (1 1 1 0 0 0 0 0 0 0 0 0 0 **1** 1 0 0 0 0 1 **1** 0 1 0 0 0)

C_7' = (0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 1) (C_5)

C_8'' = (0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0 0 1)

C_9' = (0 1 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1) (C_9)

C_{10}'' = (0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 1 0 1 0)

The process up to this point in the experiment explains the one generation i.e. one iteration in the genetic procedure. It is interesting to find the results of the evaluation process of the new population obtained after mutation. Now the total fitness of the new population which was obtained after mutation process was calculated using the Jaccard similarity and it was found equal to 0.4231 and it was found that this fitness value was more than 0.3111. After this point in the experiment, the genetic operators can be applied again.

Step4: Fitness values using Jaccard similarity functions with different queries

The genetic algorithm was applied and the fitness values were calculated using the Jaccard similarity function as fitness function in the experiment the process explained above was repeated with the different queries of the training data the measured fitness values are shown in table 8.

TABLE 8: FITNESS VALUES USING JACCARD SIMILARITY FUNCTION FOR DIFFERENT QUERIES

Query No.	Query Entered in Search Engine	Fitness Values
Q1	Terrorist Attack Mumbai	0.3111
Q2	Cloud Burst India	0.2277
Q3	Moist Attack India	0.2443
Q4	Corruption Cricket India	0.2906
Q5	Pollution River Ganga	0.4493
Q6	Power Generation India	0.2800
Q7	Sand Mining India	0.3898
Q8	Mid Day Meal India	0.3111
Q9	Sikh Riots India	0.3536
Q10	Moist Attack Train	0.3760

Step5: Criteria for the addition of the term in the original query

When the above process of the experiment is repeated for number of iterations i.e. generations, the mutated chromosomes were checked and compared for adding the term in the original query. The experiment was performed for 500 generations with probability of crossover i.e. 0.5 and probability of mutation i.e. 0.001. The experiment was also repeated by changing the parameters with different crossover probability and mutation rates. New keyword by the application of genetic algorithm can be chosen on the basis of criteria of selecting

the bit position which has the value one in the mutated chromosomes. In the first run of experiment it was found that the mutated chromosomes after the 38th the generation have values 1 at the following positions.
0 1 0 10010101001000010001O1 i.e. 2nd, 4th, 7th, 9th, 11th, 14th, 19th, 23rd, 25th
In second run the mutated chromosomes have values 1 after 7th generation
0 1 0100001011000011100l000 i.e. 2nd, 4th, 9th, 11th, 12th, 17th, 18th, 19th, 22nd

In this way, experiment was run for five times and it was found that bit at the 9th position, 11th position and 19th position becomes 1 maximum time. So bit 9 of the keyword set is then compared from the keyword set. Now From the following keyword set the 9th bit was chosen and it was found that the 9th bit is Headly. Similarly 11th bit is for India and 19th bit is for Mumbai. From these terms best suited word was considered i.e. Headly. The Keyword set [21] of step 1 i.e. {Afzal, Attack, Bandra, Blast, Bomb, Case, Friday, Government, **Headly**, Hillary, India, Injured, Intelligence, Juhu, Kasab, Killed, Maharashtra, Minister, Mumbai, Pakistan, People, Police, Rana, Taj, Terrorist} which was obtained by the same method as described in the next step to form the keyword set with the added term i.e. Headly.

Step6: New Query Formulation and extraction of Key words for newly formulated query to form the set of keywords.

On application of genetic algorithm returns the position at which the term is added i.e. 9th position is returned by algorithm. So the 9th position term from the keyword set was chosen from the keyword set as shown in the above step which is "Headly" and then new query was formulated by adding this new word in the old query. Old query is represented as Q1 and new query is represented as Q1'.

Old Query = Q1= "Terrorist Attack Mumbai"
New Query = Q1'= "**Headly** Terrorist Attack Mumbai"

New query was entered in the search box of Google search engine and again the top ten documents were retrieved for this newly formulated query with the newly added keyword. In the experiment the first ten retrieved documents were represented as D1', D2', D3', D4', D5', D6', D7', D8', D9' and D10'.The text of these documents were considered for the extraction of keywords related to the query. The extracted keywords using the Textalyser tool [25] for the new query "Headly Terrorist Attack Mumbai" are shown below.

D1' = {Headly, Mumbai, Attack, America, Pakistan, India, Sentence, Penalty}
D2' = {Headly, India, Mumbai, Pakistan, America, Prosecutor, Chicago}
D3' = {Headly, Rana, India, Terror, Attack, Mumbai, Alert}
D4' = {Headly, India, Rana, Attack, Hillary, Mumbai, Terror}
D5' = {Headly, India, Mumbai, Lashkar, Pakistan, Attack, Taiba}
D6' = {Headly, Court, Mumbai, Terrorist, India, Sentence}
D7'= {Headly, India, Attack, Rana, Mumbai, Terrorist, Extradition}
D8' = {Headly, Mumbai, Attack, Lashkar, Terrorist, Work}
D9' = {Headly, Official, Café, Mumbai, Surveillance, Intelligence, FBI},
D10'= {Headly, Mumbai, Surveillance, Lashkar, America, Location}

Keyword Set with New Formulated Query was obtained i.e.{Alert, America, Attack, Cafe, Chicago, Court, Extradition, FBI, Headly, Hillary, Location, India, Intelligence, Lashkar, Mumbai, Official, Pakistan, Penalty, Prosecutor, Rana, Sentence, Surveillance, Taiba, Terror, Work}. The process of representation of the terms present or absent [21] in document was done for the measurement of relevancy between the documents.

D1'= 0,1,1,0,0,0,0,0,1,0,0,1,0,0,1,0,1,1,0,0,1,0,0,0,0;
D2'= 0,1,0,0,1,0,0,0,1,0,0,1,0,0,1,0,1,0,1,0,0,0,0,0,0;
D3'= 1,0,1,0,0,0,0,0,1,0,0,1,0,0,1,0,0,0,0,1,0,0,0,1,0;
D4'= 0,0,1,0,0,0,0,0,1,1,0,1,0,0,1,0,0,0,0,1,0,0,0,1,0;
D5'= 0,0,1,0,0,0,0,0,1,0,0,1,0,1,1,0,1,0,0,0,0,0,1,0,0;
D6'= 0,0,0,0,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,1,0,1,0,0;
D7'= 0,0,1,0,0,0,1,0,1,0,0,1,0,0,1,0,0,0,0,1,0,0,1,0,0;
D8'= 0,0,1,0,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1;
D9'= 0,0,0,1,0,0,0,1,1,0,0,0,1,0,1,1,0,0,0,0,0,1,0,0,0;
D10'= 0,1,0,0,0,0,0,0,1,0,1,0,0,1,1,0,0,0,0,0,0,1,0,0,0;

Step7: Similarity Measurement of Retrieved Documents after adding new keywords

Similarity between the retrieved documents was measured using the Jaccard similarity functions for the newly formulated query which is formed by adding the term i.e. "Headly Terrorist Attack Mumbai". Jaccard Coefficients were obtained as Jac1' by measuring the similarity between D1' and D1', Jac2' as similarity between D1' and D2' and so on. Then the average of all the coefficients was obtained to give the overall

similarity between the documents for the query Q1' i.e. "Headly Terrorist Attack Mumbai". This process was repeated for all other documents and the average of all the coefficients was taken as shown in table 9.

TABLE 9. JACCARD SIMILARITY FOR QUERY "HEADLY TERRORIST ATTACK MUMBAI"

Docs	Similarity between documents using Jaccard Similarity Function for new query										Avg.
	D1'	D2'	D3'	D4'	D5'	D6'	D7'	D8'	D9'	D10'	
D1'	1	0.5	0.3636	0.3636	0.5	0.4	0.3636	0.2727	0.1538	0.2727	0.419
D2'	0.5	1	0.2727	0.2727	0.4	0.3	0.2727	0.1818	0.1666	0.3	0.3667
D3'	0.3636	0.2727	1	0.75	0.4	0.3	0.5555	0.4444	0.1666	0.1818	0.4435
D4'	0.3636	0.2727	0.75	1	0.4	0.3	0.5555	0.4444	0.1666	0.1818	0.4435
D5'	0.5	0.4	0.4	0.4	1	0.4444	0.5555	0.4444	0.1666	0.3	0.4611
D6'	0.4	0.3	0.3	0.3	0.4444	1	0.4444	0.2	0.1818	0.2	0.3771
D7'	0.3636	0.2727	0.5555	0.5555	0.5555	0.4444	1	0.3	0.1666	0.1818	0.4396
D8'	0.2727	0.1818	0.4444	0.4444	0.4444	0.2	0.3	1	0.1818	0.3333	0.3803
D9'	0.1538	0.1666	0.1666	0.1666	0.1666	0.1818	0.1666	0.1818	1	0.3	0.2651
D10'	0.2727	0.3	0.1818	0.1818	0.3	0.2	0.1818	0.3333	0.3	1	0.3252

The experiment was conducted as three sub experiments. In the first experiment the relevancy of the retrieved documents is measured for the entered query i.e. "Terrorist attack Mumbai" which was found to be 0.3111. Then in the second part the genetic algorithm was implemented using the Jaccard similarity as the fitness function and new word i.e. "Headly" was added to the original query which was returned using genetic algorithm .In the third part of experiment the relevancy of the retrieved documents was measured by adding the term returned by Genetic Algorithm in the original query i.e. "Headly Terrorist Attack Mumbai" which was found to be 0.3921.So it was found that there is 8.1% of improvement in the relevancy of the retrieved documents using genetic algorithm." This proves that the visibility of the search has been increased in terms of relevancy of documents with the use of the genetic algorithm and result for query Q1 is shown in table 10.

Table 10: Relevancy of retrieved documents for query Q1 with added term

Query No.	Query Entered in Search Engine	Newly Added term	Relevancy of document before adding term	Relevancy of retrieved documents after adding term	Percentage Improvement
Q1	Terrorist Attack Mumbai	Headly	0.3111	0.3921	8.1%

The process was repeated with the training data with the other queries i.e. Q2, Q3,.. Q10, the results obtained are summarized in the table 11 and the results shows that there is improvement in the relevancy of the retrieved documents when the term returned by genetic algorithm was added into the original query.

Table 11: Relevancy of retrieved documents using Jaccard similarity function with the added term

Query No.	Query Entered in Search Engine	Relevancy of documents with original query	New Added term	Relevancy of documents with added term	Percentage Improvement
Q1	Terrorist Attack Mumbai	0.3111	Headly	0.3921	8.1
Q2	Cloud Burst India	0.2277	Uttarakhand	0.3378	11.01
Q3	Moist Attack India	0.2443	Train	0.3760	13.17
Q4	Corruption Cricket India	0.2906	Fixing	0.3457	5.51
Q5	Pollution River Ganga	0.4493	Industrial	0.4756	2.63
Q6	Power Generation India	0.2800	Thermal	0.3540	7.40
Q7	Sand Mining India	0.3898	Illegal	0.4263	3.65
Q8	Mid Day Meal India	0.3111	Bihar	0.3426	3.15
Q9	Sikh Riots India	0.3536	Sajjan	0.4844	13.08
Q10	Moist Attack Train	0.3760	People	0.4205	4.45

III. Conclusion

As the retrieval process is iterative search process in which user interacts iteratively with the search system for the information need. It has been concluded that the visibility of the search of the information retrieval system is enhanced by using genetic algorithm as the relevance feedback technique using Jaccard similarity function as fitness function and further it has been concluded that the visibility of search using genetic

algorithm rely on the design of the fitness function. So as a part of future work we plan to implement genetic algorithm by using the other similarity functions like Cosine, Dice and Overlap similarity functions to study the performance of similarity functions as fitness function in genetic algorithm and other future directions include the investigation of the alternative fitness functions.

References

[1] William P. Jones, George W. Furnas, Pictures of Relevance: A Geometric Analysis of Similarity Measures, Journal of American Society for Information Science, Vol. 38, No.6, pages 420-442, 1987.

[2] Michael Gorden, Applying Probabilistic and Genetic Algorithms for Document Retrieval, Communications of ACM, Vol.31, No. 10, pages. 1208-1218, 1988.

[3] V. N. Gudivada, V. Raghavan, W. I. Grosky and R. Kasanagottu, Information Retrieval on the World Wide Web, IEEE Internet Computing, pages 58-68, 1997.

[4] R. Baeza-Yates and B. Ribiero-Neto, Modern information retrieval (Addison Wesley, New York, 1999).

[5] G. Salton and C. Buckley, Improving Retrieval Performance by Relevance Feedback, Journal of the American Society for Information Science, 41(4), pages 288-297, 1990.

[6] C. Lopez-Pujalte, V.P. Guerrero-Bote and F. Moya-Anegon, A Test of Genetic Algorithms in Relevance Feedback, Information Processing and Management, 38(6), pages 795-807, 2002.

[7] C. Lopez-Pujalte, V.P. Guerrero-Bote and F. Moya-Anegon, Order-Based Fitness Function for Genetic Algorithms Applied to Relevance feedback, Journal of the American Society for information Science and Technology, 54(2),pages 152-160,2003.

[8] M. Boughanem, C. Chrisment, L. Tamine, Multiple Query Evaluation Based on an Enhanced Genetic Algorithm, Information Processing and Management, 39, 215-231, 2003.

[9] Rocio L. Cecchini, Carlos M. Lorenzetti, Ana G. Maguitman and Nelida B. Brignole, Genetic Algorithms for Topical Web Search: A Study of Different Mutation Rates, ACM Trans. Inter. Tech., 4(4), pages 378–419, 2005.

[10] Poltak Sihombing, Abdullah Embong and Putra Sumari, Comparison of Document Similarity in Information Retrieval System by different formulation, Proc. 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications, Malaysia, Penang 2006.

[11] M. Zolghadri Jahromi, and M.R. Valizadeh, A proposed query-sensitive similarity measure for information retrieval", Iranian Journal of Science & Technology, Shiraz University, Vol. 30, no. B2, pages.171-180, 2006.

[12] A. Ahmad, A. Radwan, A. Bhagat, Abdel Latef, Abdel Mgeid A. Ali and Osman A. Sadek, Using Genetic Algorithm to Improve Information Retrieval Systems, World Academy of Science, Engineering and Technology, pages 1021-1027, 2008.

[13] S. S. Sathya and P. Simon, A Document Retrieval System with Combination Terms Using Genetic Algorithm, International Journal of Computer and Electrical Engineering, Vol.2, No.1,Pages 1-6, 2010.

[14] P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bulletin de la Société Vaudo ise des Sciences Naturelles, 37, pages. 547-579, 1901.

[15] Fatemeh Dashti, and Solmaz Abdollahi Zad, Optimizing the Data Search Results in Web using Genetic Algorithm, International Journal of Advanced Engineering and Technologies, Vol. 1, No. 1, pages 016 – 022, 2010.

[16] Wael H. Gomaa, Aly A. Fahmy, "A Survey of Text Similarity Approaches," International Journal of Computer Applications, Vol. 68, No. 13, pp. 13-18, 2013.

[17] Md. Abu Kausar, Md. Nasar, Sanjeev Kumar Singh, A Detailed Study on Information Retrieval Using Genetic Algorithm, Journal of Industrial and Intelligent Information, Vol.1, No.3, pages 122-127, 2013.

[18] Nor Hashimah Sulaiman and Daud Mohamad, A Jaccard Based Similarity Measure for Soft Sets, Proc. of IEEE Symposium on Humanities, Science and Engineering Research, pages 659-663, 2012.

[19] Suphakit Niwattanakul, Jatsada Singhthongchai, Ekkachai Naenudorn and Supachanun Wanapu, "Using of Jaccard Coefficient for Keywords Similarity," Proc. of International Multi Conference of Engineers and Computer Scientists, IMECS 2013, Hong Kong 2013.

[20] Sung-Hyuk Cha, "Comprehensive Survey on the Distance/Similarity Measures between Probability Density Functions," International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 1, Issue 4, pp. 300-307, 2007.

[21] Jaswinder Singh, Parvinder Singh, Yogesh Chaba, Performance Modeling of Information Retrieval Techniques Using Similarity Functions in Wide Area Networks, International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, Issue12, pp.786-793, 2014.

[22] David E. Goldberg, Genetic algorithm in search, optimization, machine learning (Adison Wesley, 1989).

[23] J. H. Holland, Adaptation in natural and artificial systems (2nd ed., MA: MIT Press, Cambridge, 1992).

[24] Z. Michalewicz, Genetic algorithm + data structure = evolution programs (Springer, 1996).

[25] http://textalyser.net.

[26] L. Egghe and C. Michael, Strong Similarity Measures for the Ordered Sets of Documents in Information Retrieval, Information Processing and Management, 38(6), 823-848, 2002.