Fitness Function for IRUGRA in Web Search

V.Vignesh
Assistant professor, Department of Computer Science
Veerammal Engineering College, Dindugul,
Tamil nadu, India

Dr.K.Krishnamoorthy
Professor, Sudharsan Engineering college, Pudukkottai,
Tamilnadu, India

Abstract—The explosive growth and the widespread accessibility of the Web has led to surge of research activity in the area of information retrieval on the World Wide Web. Ranking has always been an important component of any information retrieval system. In the case of Web search its importance becomes critical. Due to the size of the Web, it is imperative to have ranking functions that capture the user needs. Because of its parallel mechanism with high-dimensional space, Genetic Algorithm (GA) has been adopted to solve many of optimization problems where IR is one of them. This paper proposes searching model which is based on GA to retrieve HTML documents. This model is called Information Retrieval Using Genetic Relation Algorithm (IRUGRA). The performance of term-proximity fitness function of IRUGRA will be examined against two well known fitness functions in the IR domain. These fitness functions are the Okapi-BM25 and the Bayesian inference network model functions.

Keywords- Genetic Algorithm (GA), IRUGRA, Okapi-BM25, Bayesian inference network model functions

I. INTRODUCTION

The first web search problem has been investigated by many researchers attempting to develop approaches that are capable of providing search results that satisfy user query, examples are: (Liu, 2006; Marghny and Ali, 2005; Picarougne et al, 2002a; Kim and Zhang, 2000; Fan et al, 2004; Kushchu, 2005; Karthik, Marikkannan, and Kannan, 2008; Snasel, Moravec, and Pokorny, 2005; Tian et al 2006; Bhatia and Khalid, 2007; Kobayashi and Takeda, 2000; Haveliwala et al, 2002; Ashraf, Ozyer, and Alhajj, 2008; Yan et al, 2009; Xu, Deli, and Yu, 2009; Saini, Sharma, and Gupta, 2011). Often, these results are evaluated using precision and recall perspectives. For precision, it measures the percentage of relevant retrieved documents to the total retrieved documents, while recall measures the percentage of relevant retrieved documents to the total relevant documents in search space. In spite of several enhancements achieved in such approaches, still web users encounter two major challenges when trying to retrieve useful information (LEE, 2007; Bhatia and Khalid , 2007; Haveliwala et al 2002; Pathak, Gordon and Fan, 2000); namely; low precision and low recall. Low precision is due to the irrelevance of many of the search results where many of the highly ranked retrieved documents are not related to the user query (Picarougne et al 2002a). On the other hand, the second challenge is the low recall, which is due to the inability to index all the web documents available on the Web and related to the user query, bearing in mind that the aim of the searching engine is to retrieve all relevant documents based on the user query (high recall), and not to retrieve any irrelevant document (high precision).

IRUGRA aims ultimately to produce an IR system that is able to retrieve the relevant documents based on the user query. These documents must satisfy two criteria. The first criterion is that the obtained results must have high recall, i.e. retrieving from the search space as much relevant documents to the user query as possible. The second criterion is that the results must have high precision, i.e. the least possible irrelevant documents from the search space.

II. OVERVIEW OF FITNESS FUNCTION FOR THE IRUGRA PROCESS

IRUGRA consists of two main units. The main purpose of the first unit, namely indexing, is to extract the meaningful keywords from the documents and represent them in a way that makes the process of finding relevant documents efficient. GA is the second unit of IRUGRA and is utilized in this paper as a core of its behavior. This unit compares the user query with the indexed documents to retrieve the relevant set of documents and display them in a descending order according to a relevance measure. More precisely, in order to obtain high quality results, additional units need to be combined with IRUGRA, namely, the query formatting unit and the ranking unit. The IRUGRA are outlined as shown in Figure 1.

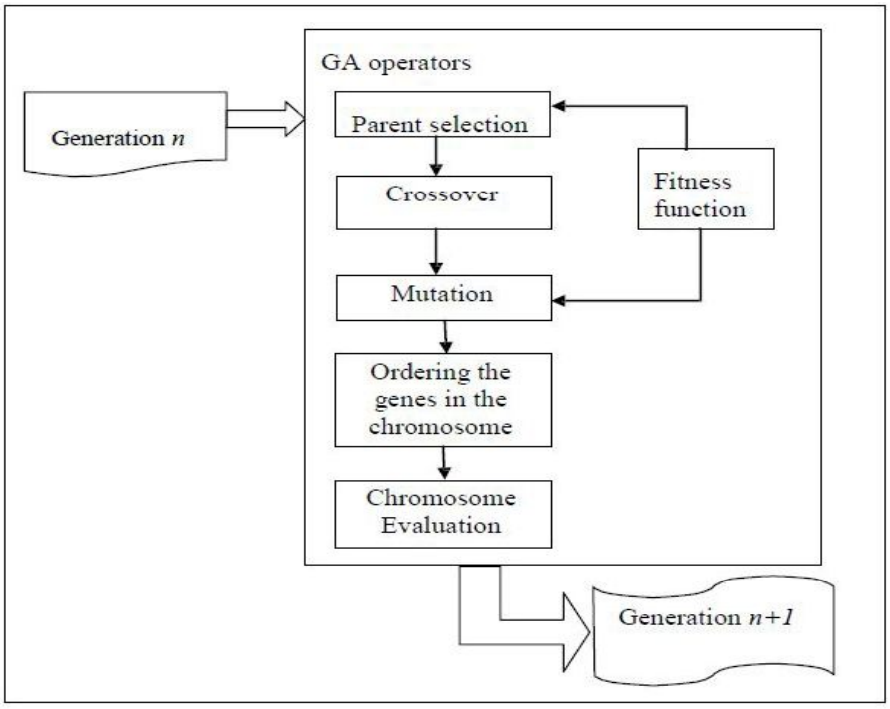

Figure 1: Overview of IRUGRA process showing the effect of fitness functions

III. THE HYBRID CROSSOVER

The proposed crossover operator chosen to be implemented in IRUGRA is a combination of reordering crossover (Vrajitoru, 2000), fusion crossover (Vrajitoru, 1998) and one-point crossover (Marghny and Ali, 2005). When genes within a chromosome are ordered based on their fitness value and the order is important, then the crossover applied to such chromosomes is called a reordering crossover. In fact, the order of genes in the proposed crossover to is important as it represents the ranked documents that will be displayed to the user. If one offspring is to be produced from the crossover process rather than two then it is called a fusion crossover. Combining these two techniques together and applying a one- point crossover on them forms the new crossover suggested in the GA unit of IRUGRA.

In the one-point crossover, GA selects one point randomly to perform exchange of genes. A reordering crossover is applied to chromosomes having their genes ordered based on their fitness value from higher to lower. Since genes are in order within the chromosome then a 2-point crossover could not produce better results as the high quality genes are on the edges while exchange is done for the genes somewhere in the middle.

The rationale behind using the ordered crossover technique over other techniques is the need to inherit the good genes and pass the good building blocks to the resulting offspring.

In fusion crossover (Vrajitoru, 1998) only one offspring is generated from the two selected parents. In this technique, the offspring inherits the genes from one of the parents with a probability according to its performance. The advantage of this technique is that the good genes of both parents are inherited simultaneously to the offspring, producing high quality offspring.

Combining the three techniques of crossover into one process allows fast convergence with high quality offspring. The ordered technique gathers the good genes into one side of the chromosome. Then the one-point crossover copies these gathered genes from the heavy side of both parents to one offspring only. This results in an offspring having the best genes of the parents.

The cross point cp is selected randomly to perform a one-point crossover. In this example it is 3. Because the first gene of x has a greater fitness value than the first gene of y, x's genes along with the fitness values are considered as the first three genes of O. To complete the genes values of O, the other three genes are copied starting from the leftmost position of y. Then a competition between the genes in both x and y is done to complete the creation of O. Because the gene at position cp+1 in y has a greater value than that of x, then y's genes are copied into O (step C in Figure 2). Once all positions in the offspring are populated with genes, these genes are

ordered from higher to lower based on their fitness value (step D in Figure 2). The algorithm of hybrid crossover is illustrated in Algorithm.

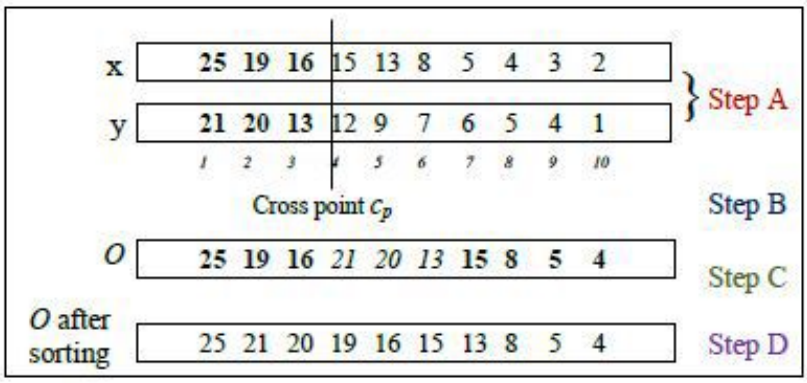

Figure 2: Illustration of the hybrid crossover process

IV. MUTATION

Mutation is the last genetic operator used in the GA unit of IRUGRA. In mutation, one or more genes are selected randomly to be replaced by other genes according to some criteria. It causes the individual genetic representation to be changed according to some probability pm ranging from 0.001 to 0.7. Because of its importance and effect on the generated chromosome, it is applied in this system with probability of 0.7.

An example of the mutation applied in this work is illustrated in Figure 3 where the numbers in this figure represent the fitness value of genes at these positions. The chromosome represented here is a continuation to the one shown in Figure 3. The position of mutation is selected randomly (position 7 in this example – Step B). The gene at this position is replaced by another gene selected randomly from the space such that it has a better fitness value or the same as the replaced one. In this example, the new value is 23 and it is better than the original one: 13 (Step C). This new value is unique within this chromosome; therefore it is exchanged with the original one. Then genes of this chromosome are re-ordered in descending order according to their fitness value to produce the new chromosome (Step D).

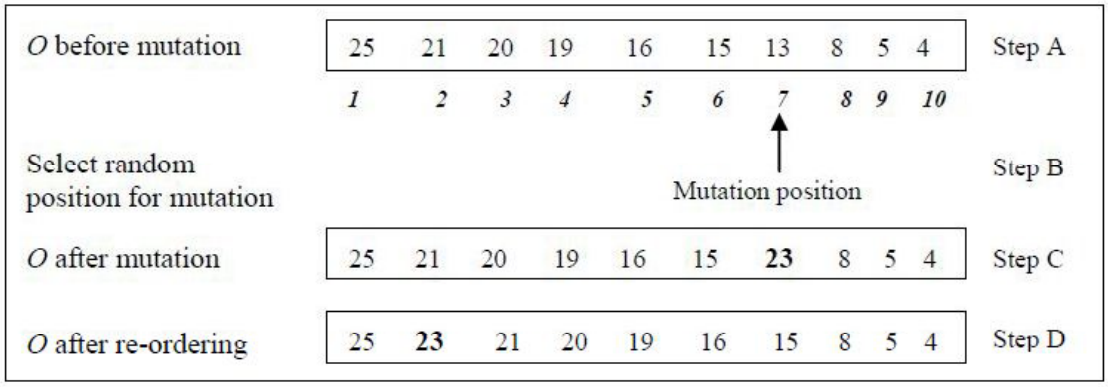

Figure 3: Illustration of the applied mutation in IRUGRA

V. FITNESS FUNCTION

Fitness function is a performance measure or reward function that measures the relevance of the documents to the user query. The decision about whether to accept or reject a document for crossover or mutation depends only on the value computed by the fitness function. This function is used in the GA process to evaluate the documents while selecting parents to perform crossover and mutation. The evolution process results in pushing high quality individuals to survive over lower ones.

From the literature review, it is deduced that the fitness functions can be categorized into three types, namely, the terms weight-based fitness function, similarity-measuring fitness function, and the custom fitness function.

The **first category** uses the term weight as an evaluation function to the document. In this category the document is evaluated by taking the summation of the query term weight (Kim and Zhang, 2003; Billhardt et al, 2002; Cummins and O'Riordan, 2006; Vrajitoru, 2000; Radwan et al 2006; Aly, 2007)

The **second category** is the similarity function which measures the distance between the document and the query vector. However, this method is most suitable for documents indexed using the vector space model, and doesn't fit into the proposed model because it uses the enhanced inverted index model (Klabbankoh and Pinngern, 2008).

The **third category** is the custom fitness function in which the fitness functions are developed using set of factors that best suit each model (Marghny and Ali, 2005; Picarougne et al, 2002a; Fan et al, 2004).

Term Proximity Fitness Function

This function shows much better performance than the multi-term fitness function explained in the previous section and will be used though out this thesis. That is because it has many advantages. These features are:

- It utilizes the term distance.
- It includes only local factors.
- It uses all the three types of factors: statistical, formatting and semantic.
- It has a maximum upper limit; hence a threshold can easily be set to determine the relevant documents.

The TPFF function is defined in formula 3.4 shown below and its terms are explained in Table 1:

$$f(D) = a\frac{\sum_{i=1}^{K} k_{ui}}{K} + b\frac{\sum_{i=1}^{K} k_{ui}-1}{\sum_{i=1}^{K-1} \min(d_{i(i+1)})} + c\frac{1}{avg(\sum_{i=1}^{K} \min(p_i))} + d\log(\frac{\sum_{i=1}^{K} w_i}{\sum_{i=1}^{K} k_i})$$

Table 1: Terms for the above formulas showing their description, domain and type

Terminology	Description	Domain	Type
k_{ui}	Represents the existence of k_{ui} within the documents	Local	Statistical
K	The query length, i.e. the total number of terms in the query	Local	Format – Statistical
$d_{i,i+1}$	Distance between term i and term i+1of the query terms	Local	Semantic
p_i	The offset (position) of term i within the document	Local	Constant
F	Document size (total number of terms in the document)	Local	Statistical
w_i	Weight of term i in the document as per Table 3-1	Local	Format – Statistical - Semantic
a, b, c and d	Weighting factors for each component	Local	Constant

This function is a summation of four components: the first one is the ratio of the existence of the query's keywords within the document, where kui represents the unique existence of keyword i within the document D. In other words, this component reflects how many of the query keywords exist in the document divided by the query size. This factor has a maximum value of one. Further explanation for computing this factor; assume "web data mining" is the requested query that is entered by the user. If D has just two keywords such as "web" and "mining" and K=3 (i.e. query size), then this factor will be equal 2/3. This factor equals 3/3 when D has all the assumed keywords (i.e. "web", "data" and "mining").

The Minimum Term Distance (MTD) between query keywords within document D is used to compute the second component of the evaluation function. However, this component is evaluated by subtracting one from the total number of existence of query's keywords within the document D. Undoubtedly, the summation of the minimum (shortest) distance between query keywords in the document D. The reason for subtracting one here is that the distance between K keywords is K-1. Recall to the above example for the suggested query (i.e., q={"web", "data", "mining"}), MTD equals 2. Indeed, this component will return 1 if all query keywords exist in the document and they appeared adjacent. The third component depends on the position of MTD within the document.

It represents the reciprocal of the average of the minimum distance between query terms

$avg(\sum_{i=1}^{K} \min(p_{t_i}))$.

The highest value of this component is given when the keywords appear right at the beginning of the document, such as in the title, header or in the first sentence of the document. However, the maximum value of this component is one only if the query consists of one word and this word is the first word in the document. Otherwise, the value is always less than one as it considers the average offset of the first appearance of MTD keyword.

VI. TESTING DIFFERENT FITNESS FUNCTIONS

For our ranking technique, the decision about whether to take or reject a document depends only on the value computed by the proposed fitness function. The proposed fitness function is developed based on local factors only to make the evaluation of the document independent of other documents. The local factors are those obtained from the document under consideration such as document size, number of unique terms within the document, and the total number of specific terms within the document.

The performance of term-proximity fitness function will be examined against two well known fitness functions in the IR domain. These fitness functions are the Okapi-BM25 and the Bayesian inference network model functions. These functions are listed in Table 2.

Table 2: List of fitness functions

Fitness method	Fitness Formula
Okapi-BM25	$f(D) = \sum_{T \in Q} \frac{(k1+1) \times tf}{(k1 \times ((1-b) + b\frac{length}{length_{avg}}) + tf} \times log\frac{N - df + 0.5}{df + 0.5}$
Bayesian inference network model	$w_i = (d_t \cdot H + (1 - d_t) \cdot \frac{\log(tf_i + 0.5)}{\log(\max tf_i + 1.0)}) \cdot \frac{\log(\frac{N}{n})}{\log N}$
Term-proximity fitness function	$f(D) = a\frac{\sum_{i=1}^{K} k_{ui}}{K} + b\frac{\sum_{i=1}^{K} k_{ui} - 1}{\sum_{i=1}^{K-1} \min(d_{i,i+1})} + c\frac{1}{avg(\sum_{i=1}^{K} \min(p_i))} + d\log(\frac{\sum_{i=1}^{K} w_i}{\sum_{i=1}^{K} k_i})$

It is obvious from the results shown in Figure 4 that the GA unit of IRUGRA which uses the term-proximity function has the highest average precision of 86% in the first top 10 ranked documents at the moment where the other two models reach only 49% for the Bayesian network inference model and 55% for the Okapi-BM25, which means that the proposed system achieves a 75.27% improvement on average in precision at the top 10 ranked documents over the Bayesian model and 31.78% over the OKAPI-BM25 models. Details of these results are illustrated in Table 3.

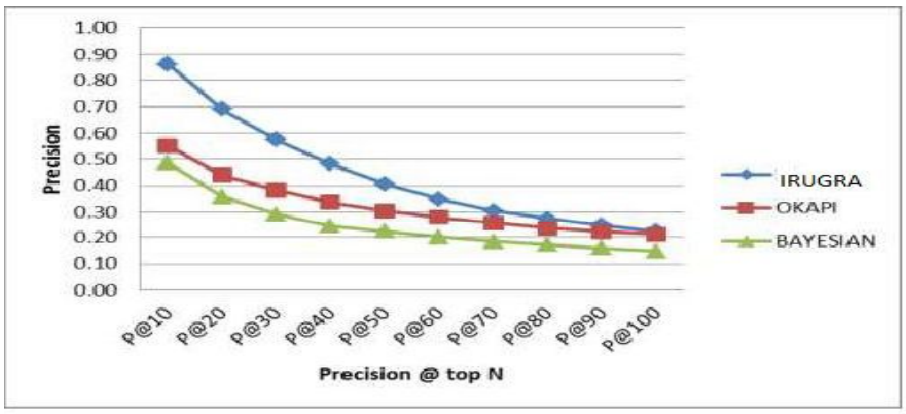

Figure 4: Comparison of P@N for Different Fitness Functions

Table 3: The P@N enhancement percentage of the term proximity fitness function over other fitness functions

Measure	IRUGRA	BAYESIAN	% of improvement	OKAPI	% of improvement
P@0	1.00	0.82	21.95	0.86	18.60
P@10	0.85	0.49	77.73	0.55	56.18
P@20	0.64	0.36	92.66	0.44	57.66
P@30	0.54	0.29	97.62	0.38	49.89
P@40	0.47	0.24	97.79	0.34	44.03
P@50	0.41	0.22	81.95	0.30	33.73
P@60	0.36	0.20	73.06	0.28	24.73
P@70	0.33	0.18	64.94	0.26	18.50
P@80	0.30	0.17	59.37	0.24	15.94
P@90	0.28	0.16	56.19	0.22	11.64
P@100	0.26	0.15	51.42	0.21	5.51
Average	**0.49**	**0.30**	**70.43**	**0.37**	**30.58**

Another measure to be considered here is the recall @ top N measure. The term proximity function in IRUGRA was able to retrieve 84% of related documents at maximum of the top 50 retrieved documents, as shown in Figure 5, whereas the Bayesian network inference model and the Okapi-BM25 reach only 75% and 71% recall respectively for the first 50 retrieved documents. The improvement of the term proximity model is 22.52% over the Bayesian network inference model and 27.55% over the Okapi-BM25 model. Table 4 illustrates the details of these results.

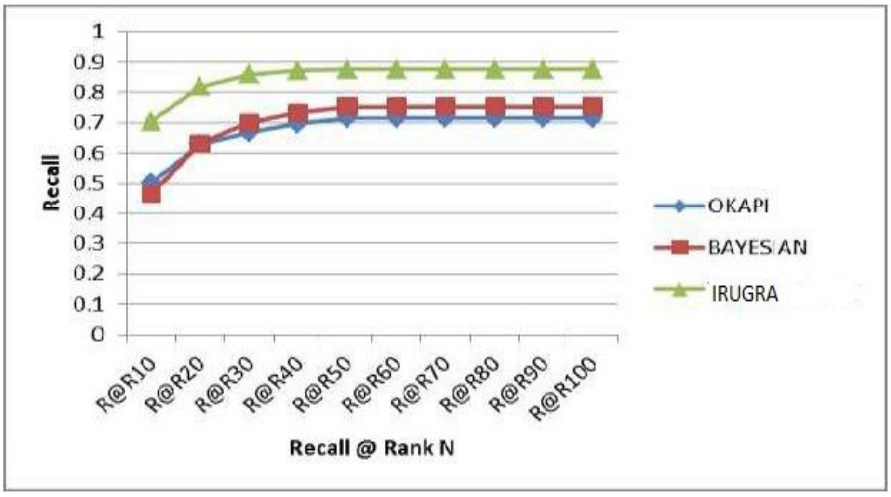

Figure 5: Comparison of R@N for Different Fitness Functions

Table 4: The R@N enhancement percentage of the term proximity fitness function over other fitness functions

Measure	IRUGRA	BAYESIAN	% of improvement	OKAPI	% of improvement
R@10	0.65	0.46	40.05	0.50	29.86
R@20	0.78	0.63	23.73	0.63	24.14
R@30	0.85	0.70	21.25	0.67	27.42
R@40	0.88	0.73	20.04	0.70	26.40
R@50	0.90	0.75	19.60	0.71	26.04
R@60	0.91	0.75	20.93	0.71	27.44
R@70	0.92	0.75	22.26	0.71	28.84
R@80	0.93	0.75	23.58	0.71	30.24
R@90	0.93	0.75	23.58	0.71	30.24
R@100	0.94	0.85	10.22	0.75	24.91
Average	**0.87**	**0.71**	**22.52**	**0.68**	**27.55**

When examining the precision @ recall measure, which is shown in Figure 6, one can notice the high performance of the proximity function in IRUGRA. The precision starts by 1 when the system retrieves 10% of relevant documents and then reduces gradually until it reaches 0.87 when retrieving all relevant documents. This means that until it retrieves 10% of relevant documents, all the displayed documents are relevant. In fact, this score was not achieved by any other technique or model. Moreover, the 0.87 at 100% recall implies that when the system retrieves all the relevant documents, only 13% of those retrieved are not relevant to the user query and they appear in low rank or at the bottom. This result is very close to the user anticipation since he or she is looking to have all top ranked documents as relevant, and most of the relevant documents appear in top position.

These results imply that the term proximity model achieved a 5.16% enhancement compared with the Bayesian inference network model, and achieved an enhancement of 13.17% when compared with the OKAPI-BM25 model. Details of these figures are illustrated in Table 5.

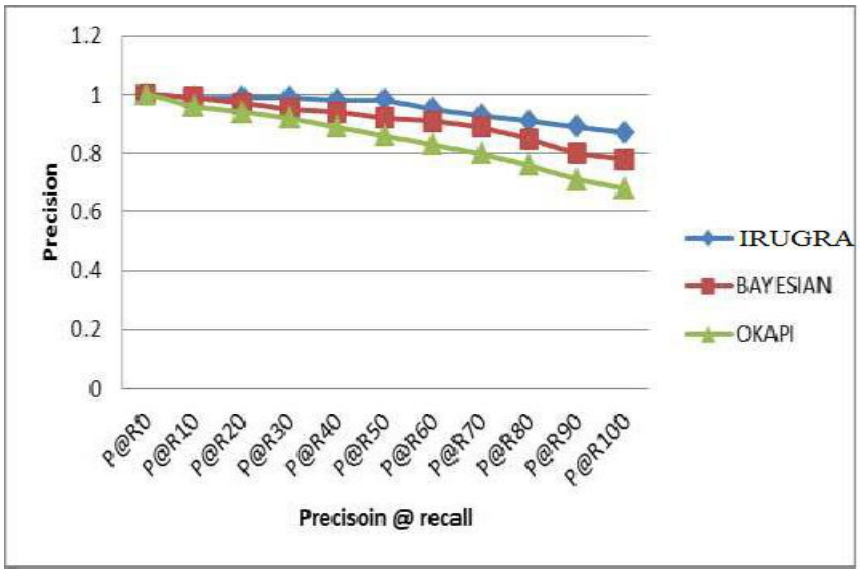

Figure 6: Comparison of P@R for Different Fitness Functions

Table 5: The P@R enhancement percentage of the term proximity fitness function over other fitness functions.

Measure	IRUGRA	BAYESIAN	% of improvement	OKAPI	% of improvement
P@R0	1.00	1.00	0.00	1.00	0.00
P@R10	0.99	0.99	1.01	0.96	4.17
P@R20	0.99	0.97	2.06	0.94	5.32
P@R30	0.99	0.95	4.21	0.92	7.61
P@R40	0.98	0.94	4.26	0.89	10.11
P@R50	0.98	0.92	6.52	0.86	13.95
P@R60	0.95	0.91	4.40	0.83	14.46
P@R70	0.93	0.89	4.49	0.80	16.25
P@R80	0.91	0.85	7.06	0.76	19.74
P@R90	0.89	0.80	11.25	0.71	25.35
P@R100	0.87	0.78	11.54	0.68	27.94
Average	**0.95**	**0.91**	**5.16**	**0.85**	**13.17**

VII. CONCLUSION:

The reason behind high results for the proposed fitness function is that it doesn't depend only on the frequency of terms within the document as other fitness functions do. It also depends on the importance of the term based on the HTML tag and on the position of the terms within the document, in addition to considering the distance between the terms. At the same time it doesn't ignore the term frequency factor. The proposed techniques are compared with the existing ones in terms of the three measures, precision at top N (P@N), recall at top N (R@N) and precision at recall (P@R). Each operator is examined using these three measures and the results are presented graphically and numerically. When compared with the OKAPI-BM25 model and Bayesian network inference model the performance of term proximity fitness function in IRUGRA is good.

REFERENCES

[1] Kofax. (2011). Retrieved 6 2011, 1, from Kofax: http://www.kofax.com/glossary

[2] Collard, P., and Escazut, C. (1995). Genetic operators in a dual genetic algorithm. Proceedings, Seventh International *Conference on Tools with Artificial Intelligence*,

[3] Saini, M. Sharma, D. Gupta, P.K . (2011), Enhancing information retrieval efficiency using semantic-based-combined-similarity-measure. *International Conference on Image Information Processing (ICIIP)*, pp. 1 - 4. IEEE Conference Publications

[4] Kui, F. and Juan, W., (2012), An Optimized Features Extraction Algorithm on VSM, *9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 29- 31 May, pp. 1471 - 1473Green, J.J. (2004). Google PageRank and related technologies, [online]. Available at: http://www.lazworld.com/whitepapers/PageRank-Technologies.pdf [Accessed 22/9/2012]

[5] Guezouli, L. and Kadache, A. (2014), Information retrieval model based on neural networks using neighbourhood, *International Conference on Information Technology and e-Services (ICITeS)*, pp. 1 – 5. IEEE Conference Publications.

[6] Aly, A. (2007). Applying genetic algorithm in query improvement problem. *Information Technologies and Knowledge* , vol.1, pp. 309-316.

[7] Manning, C. D., Raghavan, P., and Schütze, H. (2009). *An introduction to information retrieval*. Cambridge, England: Cambridge University Press.

[8] Marghny, M. H., and Ali, A. F. (2014). Web mining based on genetic algorithm. *AIML 05 Conference*. Cicc, Cairo, Egypt.