ISSN:0975-9646

Relevancy Enhancement of Query with Czekanowski Coefficient by Expanding it Using Genetic Algorithm

Neha Soni#1, Jaswinder Singh#2

#1Student, M.Tech. Department of Computer Science and Engineering, GJUS&T, Hisar, Haryana, India,

#2Assistant Professor, Department of Computer Science and Engineering, GJUS&T, Hisar, Haryana, India

***Abstract*— Genetic Algorithm searches for a good solution to a problem by taking inspiration from the natural selection of living beings. Among their lots of uses, we can consider information retrieval. In this field, the aim of genetic algorithm is to help an information retrieval system to find, in a huge collection of documents, a good reply to a query expressed by the user so as to increase the relevancy of the query. For this, the query is expanded using genetic algorithm. In this paper, Czekanowski coefficient is used during the expansion process to increase the efficiency of information retrieval.**

***Keywords*— information retrieval, similarity measure, genetic algorithm, query expansion**

I. INTRODUCTION

The amount of information on the web is growing rapidly. The number of queries that search engine can handle has grown incredibly. So it is possible sometimes that the document which is not most related to the user's query is presented to the user by the search engine. Junk results often wash out any results that a user is interested in. In this situation, the retrieval of documents relevant to the user's query is of utmost importance.

II. INFORMATION RETRIEVAL (IR)

Information Retrieval is the study of how to determine and retrieve from a corpus of stored information, the sections which are responsive to particular information need. Given the user query, the key goal of information retrieval system is to retrieve information which may be relevant to the user.

A. Three Basic Components of Information Retrieval System

1) Query Subsystem: It is a system that allow users to formulate their queries and then present the relevant document retrieved by the system for user's query.

2) Matching Function: Both the query and documents in database are compared using the matching function which gives a value that measures the similarity between query and document.

3) Document Database: It is the storage space where all the documents in the database are stored.

B. Models of Information Retrieval System

1) Boolean Model: It is most common type of model. It is based on Boolean Logic (AND, OR, NOT). Here retrieval of documents is based on whether or not the documents contain the query terms.

2) Vector Space Model (VSM): Here documents and queries are represented as vectors of weights in multidimensional space. Each weight denotes the importance of the corresponding keywords respectively in the document or in the query.

3) Probabilistic Model: This model is specially based on the probability ranking principle, which states that an information retrieval system is supposed to rank the documents based on their probability of relevance to the query.

III. SIMILARITY MEASURE

It is a function used to measure as to what amount the query and documents are similar to each other. It gives a value which decides the degree of similarity. Some of the measures are:

$$\text{Cosine}(Q, D_i) = \frac{\sum_{j=1}^{t} w_{q,j} d_{i,j}}{\sqrt{\sum_{j=1}^{t}(w_{q,j})^2 \sum_{j=1}^{t}(d_{i,j})^2}}$$

$$\text{Jaccard}(\text{Q}, \text{D}_\text{i}) = \frac{\sum_{\text{j}=1}^{\text{t}} \text{w}_{\text{qj}} \text{d}_{\text{ij}}}{\sum_{\text{j}=1}^{\text{t}}(\text{d}_{\text{ij}})^2 + \sum_{\text{j}=1}^{\text{t}}(\text{w}_{\text{qj}})^2 - \sum_{\text{j}=1}^{\text{t}} \text{w}_{\text{qj}} \text{d}_{\text{ij}}}$$

$$\text{Dice} = \frac{2\sum_{\text{i}=1}^{\text{n}} \text{A}_\text{i} \cdot \text{B}_\text{i}}{\sum_{\text{i}=1}^{\text{n}}(\text{A}_\text{i})^2 + \sum_{\text{i}=1}^{\text{n}}(\text{B}_\text{i})^2}$$

$$\text{Czekanowski} = \frac{2\sum_{\text{i}=1}^{\text{d}} \min\,(\text{P}_\text{i}, \text{Q}_\text{i})}{\sum_{\text{i}=1}^{\text{d}}(\text{P}_\text{i} + \text{Q}_\text{i})}$$

IV. GENETIC ALGORITHM (GA)

Genetic algorithms are generally used for optimization problems. Through operations based on natural selection, they search for the best solution to the problem.

The GA starts with an initial population containing a number of individuals and representing the generation number. Given an old generation, a new generation is built from it according to the following steps.

A. Reproduction

This step selects some individuals from the old generation, according a better chance to individuals presenting a better performance.

B. Crossover.

This operation perform the mating of two chromosomes that gives birth to two new offspring. Crossover happens with one parameter that is known as probability of crossover Pc.

C. Mutation.

This step involves changing one bit of a chromosome from 0 to 1 or vice versa. This is performed with a probability of mutation Pm.

V. QUERY EXPANSION

The explosive growth of www is making it difficult for a user to locate information that is relevant to user's interest. The average length of queries by the user is less than two or three keywords. Short queries and the incompatibility between the terms in user queries and documents strongly affect the efficiency of relevant document retrieval. Query expansion is a technique to increase the effectiveness of information retrieval. It is the process of supplementing additional terms to the original query to improve the retrieval performance.

VI. RELATED WORK

Bangorn Klabbankoh and Ouen Pinngern [1], presented an online information retrieval using GA in order to increase the information retrieval efficiency. The experiment indicated that precision and recall were invert to each other. *J. Usharani, K Iyakutti* [2], proposed a method that used GA for finding similarity of web documents based on cosine similarity. The query was expanded and it was observed that average relevancy was increased after expansion. *Manoj Chahal, Jaswinder Singh* [3], tested horng and yeh coefficient to measure the similarity between query and documents. It was observed that relevancy was increased after expanding the query using this coefficient. *Eman Al Mashagba, Feras Al Mashagba and Mohammad Othman Nassar* [4], studied different similarity measures in VSM and for each similarity measure, investigated ten different GA approaches based on different fitness functions, different mutation and crossover strategies to find the best one for Arabic data. *Priya I. Borkar and Leena H. Patil* [5], presented a Hybrid Genetic Algorithm-Particle Swarm Optimization (HGAPSO) model for the retrieval of web information. *Noor Ali Ameen Albayaty and Nushwan Yousif Baithoon* [6], devoted several contributions, the first was: query improvement using GA, and the second was: building an intelligent search system based on VSM that used the new queries and compared their results with the original. *Jose R. Perez-Agiiera* [7], invented a new GA used to change the set of terms that the query contained, without user's supervision with the use of morphological thesaurus. *Suhail S. J. Owais, Pavel Kromer, and Vaclav Snasel* [8] investigated the GA's use in IR for optimizing a Boolean query. *Abdelmgeid Amin Aly* [9], reformulated the query based on query expansion method and experimented a test on data collection which showed that the improvement increases with the size of collection and with the number of additional search terms that expanded the original query.

M. Shamim Khan and Sebastian Khor [10], described a scheme that analysed an initially retrieved documents to automatically expand the user's query. *Hazra Imran and Aditi Sharan* [11], addressed the basic issues related to QE and automatic QE strategies. *Claudio Carpineto and Giovanni Romano* [12], presented a unified view of recent approaches to automatic QE. *Jose R. Perez-Aguera and Lourdes Araujo* [13], studied the two approaches, co-occurrence and probabilistic distributed analyses, for query expansion. *Bhawani Selvaretnam, Mohammed Belkhatir* [14], identified the factors influencing the performance of QE methods. *Ashish Kishor Bindal and Sudip Sanyal* [15], presented a stochastic method for optimizing the query vector without user involvement with the use of particle swarm optimization approach. *Yogesh Kakde* [16], discussed the important work done on QE between the period 1970 to 2012. *Bodo Billerbeck and Justin Zobel* [17] introduced an alternative methods for reducing query evaluation costs and developed a new method based on keeping a brief summary of each document in memory. *Claudio Carpineto, Renato de Mori, Giovanni Romano and Brigitte Bigi* [18], presented a method that assigned scores to candidate expansion terms for query reweighting. *Mohammed Otair, Ghassan Kanaan and Raed Kanaan* [19], used combination of the expansion techniques that optimized the Arabic queries.

VII. RESEARCH TOOLS

A. Text Analyser Tool.

It is a powerful tool which gives statistics about a text including word count, unique words, number of sentences, average words per sentence, lexical density. This tool can also be used to analyse the links on web pages. This tool help us to find the top keywords from relevant document. These keywords are actually used for making the chromosomes which is the backbone of GA, in which the research implementation is carried out.

B. MATLAB (MATrix LABoratory).

It is a numerical computing environment which implements fourth generation programming language. It is developed by MathWorks. MATLAB allows matrix manipulations, functions and data plotting, algorithmic implementations, user's interface creation, and interfacing with programs written in other programming languages.

VIII. EXPERIMENTAL SETUP AND RESULTS

This Section discuss about how experiment is conducted and results obtained during the experiment. The web documents are encoded into strings of 0's and 1's. The documents have been obtained using some search query. In our experimental setup, 10 documents are retrieved for each of the 10 queries. This set of 10 documents serves as a document database.

A. *Process of Experiment is as follow*

1) Query is input to the Google search engine.
2) Top keywords of each retrieved documents are extracted using text analyser tool, making a list of n keywords that are most related to the query.
3) Generate initial population by encoding retrieved documents to chromosomes for each query.
4) For making the chromosomes, first of all, the n keywords related to the query are arranged in alphabetic order. These document chromosomes are then encoded into binary form by setting the keyword$_i$ to 0, if the keyword$_i$ is not present in the document and setting it to 1, if it is present in the document. The length of chromosome depends on the number of keywords of document retrieved from the user query.
5) Calculate initial relevancy of query using similarity coefficient.
6) Initial population now consists of these encoded documents. Pass this initial population to GA.
7) At the end, the document chromosome having the best value of fitness function is find out and the words that are found to be turned 1 from 0 are nominated for expanding the query. The words that is most related to the query is selected manually and the query is expanded with that word and requery once.
8) Now find out the average relevancy of the query after expansion using similarity measure and compare it with the initial relevancy and percentage improvement in relevancy is noted.

B. *Experimentation*

The research work conducted the test for 10 different queries and 10 documents for each query. The length of chromosome taken is 25. The similarity measure chosen is Czekanowski coefficient [20].

$$\text{Czekanowski} = \frac{2\sum_{i=1}^{d}\min(P_i, Q_i)}{\sum_{i=1}^{d}(P_i + Q_i)}$$

where P_i is document vector, Q_i is query or document vector and d is the number of documents which is 10 here. The similarity measure works as a fitness function for GA. A complete MATLAB code has been written with roulette wheel selection operator. In our experiment the GA procedure for query expansion is repeated with three types of crossovers: Single Point Crossover (SPC), Two Point Crossover (TPC) and Uniform Crossover (UC) and percentage improvement in average relevancy is calculated in each case and then compared to each other. Experiment conducted with various GA parameters as: probability of crossover P_c=0.2, 0.5, and 0.8 each with probability of mutation (P_m=0.01, 0.005, and 0.001) to do the performance analysis of GA based retrieval system. GA is run for 1000 iterations. The efficiency parameter is average relevance.

C. *Results*

1) *Tabular Representation of Average Relevancy*

Table I shows the average relevancy of queries initially and after expansion with the words in each case of crossover: SPC (Single Point Crossover), TPC (Two Point Crossover), and UC (Uniform Crossover). The increase in average relevancy in each case of crossover is shown in table I. The $P_c$taken is 0.5 and P_m is 0.01 for this whole procedure.

TABLE I
INITIAL AVERAGE RELEVANCY AND RELEVANCY AFTER EXPANSION OF QUERY

Query	Initial Relevancy	Relevancy With SPC	Relevancy With TPC	Relevancy With UC
Q1	0.7475	0.7911	0.8193	0.8117
Q2	0.6477	0.7850	0.8598	0.8215
Q3	0.7465	0.7521	0.7720	0.7836
Q4	0.6532	0.7224	0.7383	0.7314
Q5	0.7229	0.7630	0.7840	0.7840
Q6	0.7703	0.8605	0.8651	0.8651
Q7	0.6508	0.7112	0.8022	0.7795
Q8	0.6123	0.6999	0.7369	0.7519
Q9	0.6231	0.6866	0.7120	0.7029
Q10	0.7152	0.7265	0.9019	0.7834

2) *Graphical Representation of Relevancy*

Fig. 1 shows the graphical representation of average relevancy of queries initially and after expansion in case of SPC (Single Point Crossover), TPC (Two Point Crossover) and UC (Uniform Crossover).

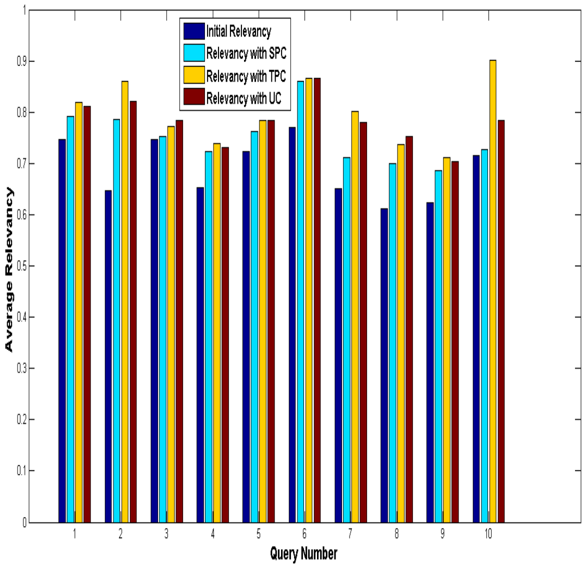

Fig. 1 Graphical Representation of Queries initially and After Expansion

3) *Percentage Improvement in Relevancy*

TABLE II shows the percentage improvement in average relevancy of queries after expansion with new keywords in case of SPC (Single Point Crossover), TPC (Two Point Crossover) and UC (Uniform Crossover).

TABLE II
PERCENTAGE IMPROVEMENT WITH SPC, TPC AND UC

Query	**% improvement (SPC)**	**% improvement (TPC)**	**% improvement (UC)**
Q1	5.83	9.60	8.58
Q2	21.19	32.7	26.8
Q3	0.75	3.41	4.96
Q4	10.59	13.02	11.97
Q5	5.54	8.45	8.45
Q6	11.70	12.30	12.30
Q7	9.28	23.26	19.77
Q8	14.30	20.34	22.79
Q9	10.19	14.26	12.80
Q10	1.57	26.10	9.5

We can see from TABLE II that TPC performs better than SPC for all the ten queries and UC is comparable to TPC as in some cases UC perform a little bit better than TPC and in other cases equally well as TPC.

4) *Required Number of Generations for Convergence*

TABLE III shows the required number of generations so as to converge the query on average value, when run for 1000 iterations for the three cases of crossover: SPC, TPC and UC with Pm=0.01,0.005 and 0.001 (and (Pc=0.2,0.5,0.8)).

Actually it shows the effect of mutation and crossover over the chromosomes for the three cases of crossover: SPC (Single Point Crossover), TPC (Two Point Crossover) and UC (Uniform Crossover). It is observed that, with mutation probability Pm=0.01, (Pc=0.2, 0.5, 0.8) although all the queries converge at one but required more number of generations for convergence. With Pm=0.005, (Pc=0.2, 0.5, 0.8) also all the queries converge at one but in less number of generations as compared to Pm=0.01. With Pm=0.001, (Pc=0.2, 0.5, 0.8) all the queries converge at one in least number of generations. All this shows that less mutation rate is best for these queries.

IX. CONCLUSION AND FUTURE WORK

This paper tested a similarity measure named as Czekanowski coefficient in query expansion process. At first, relevancy of the query is measured with old keywords and then GA is applied with SPC (Single Point Crossover), TPC (Two Point Crossover) and UC (Uniform Crossover) to find the keywords in each case, so as to expand the query. After expanding, relevancy of the query is measured and compared with the query with old keywords. The percentage improvement in relevancy is noted and it is observed that TPC performs better than SPC and UC is comparable to TPC and it is also observed that at lowest mutation rate, all the chromosomes converge into one in lesser number of generations. So lowest mutation rate is best for these queries. It is observed that the usage of GA increases the relevancy of retrieved documents.

As a part of future work, the effect of adjusting the values of various parameters of GA such as mutation probability, crossover probability, size of population, can be studied and the same similarity measure (Czekanowski) can be used with weighted vector form. Also different types of crossovers and mutations can be applied.

ACKNOWLEDGEMENT

I would like to thank to all the people those who have helped me to give the knowledge about these papers and I am thankful to my guide with whose guidance I would have completed my research work and make it to published, finally I like to thank to all the websites which I have gone through and have refer to create my research paper successful.

TABLE III
GENERATIONS NEEDED TO CONVERGE THE QUERY ON AVERAGE VALUE

Query	**Generations needed to converge the query on average value when run for 1000 generations with Pm=0.01, 0.005 and 0.001 (and Pc=0.2,0.5,0.8) is shown below as: Average Relevancy , Generations Needed when Pc=0.2, when Pc=0.5, when Pc=0.8**								
	Pm=0.01			**Pm=0.005**			**Pm=0.001**		
	SPC	**TPC**	**UC**	**SPC**	**TPC**	**UC**	**SPC**	**TPC**	**UC**
Q1	1,312,738,559	1,178,121,284	1,241,229,53	1,207,68,539	1,12,22,25	1,12,16,37	1,7,23,9	1,4,5,5	1,7,7,10
Q2	1,59,844,393	1,44,417,259	1,147,286,256	1,186,48,165	1,8,5,13	1,5,17,14	1,4,10,18	1,7,4,6	1,14,11,11
Q3	1,466,719,14	1,28,324,269	1,234,903,130	1,70,181,216	1,6,31,9	1,16,15,17	1,7,7,17	1,9,9,8	1,4,6,8
Q4	1,896,292,366	1,155,405,120	1,403,369,318	1,35,117,249	1,12,18,3	1,11,35,25	1,6,16,11	1,5,8,6	1,6,11,16
Q5	1,495,499,23	1,143,192,107	1,283,76,447	1,125,198,159	1,14,15,31	1,27,29,30	1,4,7,21	1,6,12,8	1,8,7,6
Q6	1,349,770,255	1,95,22,86	1,72,534,237	1,20,69,63	1,19,17,29	1,22,44,21	1,10,16,17	1,8,7,11	1,7,12,4
Q7	1,98,191,182	1,65,162,58	1,53,923,619	1,100,73,97	1,25,17,6	1,14,11,41	1,8,14,11	1,7,10,5	1,6,17,6
Q8	1,714,764,160	1,48,95,179	1,143,314,593	1,14,102,24	1,25,14,14	1,21,32,19	1,10,8,19	1,7,7,5	1,9,21,11
Q9	1,897,741,613	1,82,197,187	1,374,207,959	1,89,58,366	1,24,14,12	1,21,21,34	1,10,21,15	1,7,6,8	1,5,11,13
Q10	1,585,345,57	1,303,115,86	1,327,450,459	1,75,42,120	1,10,13,29	1,18,34,30	1,14,9,17	1,6,10,7	1,4,15,15

REFERENCES

[1] B. Klabbankoh and Q. Pinngern, "Applied genetic algorithms in information retrieval," Faculty of Information Technology, *King Mongkuts Institute of Techology* Ladkrabang, 2000.

[2] J. Usharani, K Iyakutti, "A Genetic Algorithm based on Cosine Similarity for Relevant Document Retrieval," *International Journal of Engineering Research & Technology*, Vol. 2 Issue 2, February- 2013.

[3] Manoj Chahal, Jaswinder Singh "Effective Information Retrieval Using Similarity Function: Horng and Yeh Coefficient," *International Journal of Advanced Research in Computer Science and Software Engineering*, Volume 3, Issue 8, August 2013.

[4] E. Al Mashagba, F. Al Mashagba, and M.O Nassar, "Query Optimization Using Genetic Algorithms in the Vector Space Model," *International Journal of Computer Science Issues (IJCSI)*, vol. 8(5), 2011.

[5] P. I. Borkar and A. P. L. H. Patil, "A model of hybrid genetic algorithm-particle swarm optimization (hgapso) based query optimization for web information retrieval," *IJRET*, vol. 2(1), pp. 59 – 64, 2013.

[6] N. A. A. Albayaty and N. Y. Baithoon, "File Search with Query Expansion in a Network System(s)," *Information and Knowledge Management*, vol.3, pp. 23– 30, 2013.

[7] L. Araujo, H. Zaragoza, J. R. Pérez-Agüera, and J. Pérez-Iglesias, "Structure of morphologically expanded queries: A genetic algorithm approach," *Data & Knowledge Engineering*, vol. 69(3), pp. 279–289, 2010.

[8] S. S. Owais, P. Krömer, and V. Snaˇsel, "Query optimization by Genetic Algorithms," *in DATESO*, vol. 129, pp. 125–137, 2005.

[9] A. Abdelmgeid Amin, "Using a Query Expansion Technique to Improve Document Retrieval," 2008.

[10] M. Shamim Khan and S. Khor, "Enhanced web document retrieval using automatic query expansion," *Journal of the American Society for Information Science and Technology*, vol. 55(1), pp. 29–40, 2004.

[11] H. Imran and A. Sharan, "Thesaurus and query expansion," *International journal of computer science & information Technology (IJCSIT)*, vol. 1(2), pp. 89–97, 2009.

[12] C. Carpineto and G. Romano, "A survey of automatic query expansion in information retrieval," *ACM Computing Surveys (CSUR)*, vol. 44(1), 2012.

[13] J. R. Pérez-Agüera and L. Araujo, "Comparing and combining methods for automatic query expansion," 2008.

[14] B. Selvaretnam and M. Belkhatir, "Natural language technology and query expansion: issues, state-of-the-art and perspectives," *Journal of Intelligent Information Systems*, vol. 38(3), pp. 709–740, 2012.

[15] A. K. Bindal and S. Sanyal, "Query Optimization in Context of Pseudo Relevant Documents," *in 3rd Italian Information Retrieval (IIR) workshop*, 2012.

[16] Y. Kakde, "A Survey of Query Expansion until June 2012," *Indian Institute of Technology*, Bombay, 2012.

[17] B. Billerbeck and J. Zobel, "Techniques for efficient query expansion,*" in String Processing and Information Retrieval*, pp. 30–42, 2004.

[18] C. Carpineto, R. De Mori, G. Romano, and B. Bigi, "An information-theoretic approach to automatic query expansion," *ACM Transactions on Information Systems (TOIS)*, vol. 19(1), pp. 1–27, 2001.

[19] Otair, Mohammed, Ghassan Kanaan, and Raed Kanaan, "Optimizing an Arabic Query using Comprehensive Query Expansion Techniques," *International Journal of Computer Applications* pp. 42-49, 2013.

[20] S.H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," City, vol. 1, no. 2, p. 1, 2007.