Genetic Algorithm-Based Query Expansion for Improved Information Retrieval

Pragati Bhatnagar and Narendra Pareek

Abstract This paper is focused toward query expansion, which is an important technique for improving retrieval efficiency of an information retrieval system. In particular, the paper proposes an evolutionary approach for improving efficiency of pseudo-relevance feedback-based query expansion (PRFBQE). In this method, the candidate terms for query expansion are selected from an initially retrieved list of documents, ranked on the basis of co-occurrence measure of the terms with the query terms. Top *n* selected terms create a term pool. From this term pool, genetic algorithm (GA) is used to select a thematically rich combination of terms, which provides the terms for expanding the query. We call this method as genetic algorithm-based query expansion (GABQE). The experiments were performed on standard CISI dataset. The results are quite motivating, and one can clearly observe the difference in the result when GA is not used and when GA is used. The paper uses GA for improving pseudo-relevance feedback (PRF)-based query expansion, but at the same time, it can also be generalized and tested for other types of query expansions, where terms may be selected in a different way, but a good combination of expansion terms can be obtained using GA.

Keywords Information retrieval · Query expansion · Genetic algorithm

1 Introduction

Query expansion has been widely investigated as a method for improving the performance of information retrieval system. Though a lot of work has been done in this area, obtaining a proper expansion of query is still an unsolved problem. Different researchers are coming up with different techniques of query expansion.

P. Bhatnagar (✉) · N. Pareek
Department of Computer Science, M.L. Sukhadia University, Udaipur, Rajasthan, India
e-mail: pragatibhat@gmail.com

L.C. Jain et al. (eds.), *Intelligent Computing, Communication and Devices*,
Advances in Intelligent Systems and Computing 308,
DOI 10.1007/978-81-322-2012-1_6

A popular type of query expansion is **pseudo-relevance feedback-based query expansion (PRFBQE)**. The most important concern in query expansion is the source for the expansion terms and criteria for selecting and ranking the expansion terms. Xu and Croft [1, 2] provided an efficient co-occurrence-based measure for ranking the query expansion terms. However, Cao et al. [3] even question the basic notion of goodness of a term. They argue that a goodness criterion, which is based on the frequency of terms in PRF-based documents or their distribution in corpus, is itself not appropriate. The authors then propose to integrate a term classification process to predict the usefulness of expansion terms. Some work has been done for using genetic algorithm (GA) for information retrieval and query expansion. Most of the work has been done to tune the weights of query terms or matching functions. Pathak et al. [4] have used GA for improving the efficiency of matching function of an information retrieval system. Horng [5] used GA to tune the weight of retrieved query terms. The experiment has been done on Chinese data collection. Araujo [6] have used GA for query expansion based on stemming and morphological variations. Cecchini [7] has used GA along with the notion of thematic context to improve query expansion. The proposed techniques place emphasis on searching for novel material that is related to the search context.

The above papers are somewhat related to our work; however, our work is different in the sense that it has been used to achieve an improved ranking of query expansion terms obtained using PRFBQE. In PRFBQE, terms are ranked independently of each other. We observed that this leads to a serious problem, as the terms have dependence over one another. A proper combination of these terms, which is cohesive, can improve the result dramatically. Thus, given n candidate terms, we are interested in selecting a suitable subset of the terms that optimize precision/recall of the query. This makes the problem as an optimization problem, and since the search space is very large, we cannot have a polynomial time algorithm for solving the problem. Thus, it is appropriate to use GA for finding a cohesive selection of expansion terms that optimize the performance of query. Based on these notions, we present the idea of genetic algorithm-based query expansion (GABQE). After explaining our approach, we provide an algorithm for performing GABQE. We performed experiments on standard CISI dataset (benchmark dataset for information retrieval). The results are quite motivating, and one can clearly observe the difference in the result when GA is not used and when GA is used.

2 Proposed Approach

We have tried to improve the performance of PRFBQE by using GA. We call it **GAQBE**. This helps us to provide thematically rich collection of expansion terms. GABQE approach is divided two parts: construction of term pool and selection of expansion terms from the term pool. In order to present our approach, in Sect. 2.1, we discuss about the construction of term pool, in Sect. 2.2, we discuss the GA-based approach for selection of expansion terms.

2.1 Construction of Term Pool

In order to construct the term pool, we first retrieve top n documents for the query using a matching function. In our problem, a query is selected and its Okapi measure is used as a matching function. The Okapi measure is given by following equation:

$$\text{Okapi}(Q, D_i) = \sum_{T \in Q} w \frac{(k_1 + 1)tf}{K + tf} \times \frac{(k_3 + 1)qtf}{k_3 + qtf} \tag{1}$$

Q is the query that contains words T.
k_1, b, k_3 are constant parameters (k_1 = 1.2, b = 0.75, k_3 = 7.0)

$$K \text{ is } k_1(1 - b) + \left(b.\frac{dl}{avdl}\right)$$

tf is term frequency of term in document D_i
qtf is term frequency in query Q

$$w \text{ is } \log \frac{(N - n + 0.5)}{(n + 0.5)}$$

N is number of documents, n is number of documents containing the term.
dl and $avdl$ are document length and average document length.

All documents and sorted on the basis of Okapi measure. All the unique terms of top N documents are selected and are ranked on the basis of their co-occurrence with query terms. Top m terms co-occurring with original query terms are selected as candidate terms for expansion. For our experiments, we have used well-known Jaccard coefficient as a co-occurrence measure, which is given as:

$$\text{Jaccard\_co}(t_i, t_j) = \frac{d_{ij}}{d_i + d_j - d_{ij}} \tag{2}$$

where t_i and t_j are the terms for which co-occurrence is to be calculated and d_i and d_j are the number of documents in which terms occur, respectively, and d_{ij} is the number of documents in which t_i and t_j co-occur.

We can apply this coefficient to measure the similarity between the query terms and terms in the documents. Incorporating inverse document frequency and applying normalization, we define degree of co-occurrence of a candidate term with a query term as follows:

$$\text{co\_degree}(c, t_i) = \log_{10}(\text{co}(c, t_i) + 1) * (idf(c)/\log_{10}(D)) \tag{3}$$

$$idf(c) = \log_{10}(N/N_c) \tag{4}$$

where

N	number of documents in the corpus
D	number of top ranked documents used
c	candidate term listed for query expansion
t_j	jth term of the document
Nc	number of documents in the corpus that contain c
Nco(c, t_j)c	number of documents in the corpus that contain c

Above formula can be used for finding similarity of a term c with individual query term. To obtain a value measuring how good c is for whole query Q, we need to combine its degrees of co-occurrence with all individual original query terms t_1, t_2, t_3…. So, we use

$$\text{Suitability for } Q = f(c, Q) = \prod_{t_i \text{ in } Q} (\delta + \text{co\_degree}(c, t_i))^{idf(t_i)} \tag{5}$$

Above equation provides a suitability score for ranking the terms co-occurring with entire query. The terms of the document are ranked on the basis of similarity value obtained and top m terms form a term pool.

2.2 Genetic Algorithm for Selecting Expansion Terms

We have discussed the approach for developing the term pool. The term pool contains the good candidate terms that may be suitable for query expansion. Now, we have to select an optimal combination of a subset of these terms, which are cohesive among themselves and are better suited for query expansion. In order to apply GA, we require a proper fitness function. Moreover, the performance of GA is very much dependent on proper representation of chromosome, proper selection, and tuning of crossover and mutation operators.

2.2.1 Representation of Chromosome

We have used a chromosome representation where each gene represents a specific candidate term. One particular combination of expansion terms represents a chromosome. Considering number of terms as 10, chromosomes are represented in following way

t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}

where each t_i represents a term index.

2.2.2 Fitness Function

Fitness function is based on the suitability or goodness of the query in retrieving the relevant documents, which is measured by recall or precision. We have used recall of the retrieved result as a fitness function. Recall is given by:

$$\text{Recall} = \frac{|R_a|}{|R|} \tag{6}$$

R Set of relevant documents retrieved
R Set of all relevant documents

Selection, crossover, and mutation are the GA operators that are applied to the above chromosomes. Standard single-point crossover was used here. The algorithm for GAQBE is as follows

2.2.3 Algorithm for GABQE

Final algorithm for GABQE is presented in Table 1.

3 Experiments and Results

We tested the algorithm on CISI dataset. This dataset provides a benchmark for testing efficiency of an information retrieval system. CISI data consist of 1,460 abstracts from information retrieval papers and 112 queries. In order to perform PRFBQE, two important parameters need to be set: n (number of top documents to be retrieved), m (number of expansion terms). After extensive experiment on corpus, the values were set as $n = 10$ and $m = 10$.

For setting GA parameters, chromosome length was set to 10, as number of expansion terms was 10. Other parameters were fixed after extensive experimentation. Population size and final number of generations were 40 and 50, respectively. Crossover and mutation rates were kept as: 0.7 and 0.03.

The results were evaluated and compared for: without query expansion, PRFBQE and GABQE. The improvement in the result can be observed from recall precision curve as shown in Fig. 1. The effect of GA was observed for all the queries. Figure 1a shows 10-point recall precision for query number 28. It can be observed that GABQE is showing improvement over standard PRFBQE. Figure 1b gives average recall precision curve for all the queries. Again, it is observed that GABQE improves the results on average.

The effect of GABQE can be observed from generationwise average fitness curve. Fitness is measured by recall (Eq. 6). Figure 2a shows such graph for query number 28. Similar graph is presented for generationwise average of average fitness for all the queries in Fig. 2b. As it can be observed, average recall is

Table 1 Algorithm developed for selecting expansion terms using GABQE

```
Algorithm:   GABQE
Input : Document corpus D
        Query collection Q
Process:
                Select the  query q from Q
                Preprocess the documents in document collection D
                Calculate similarity measure of each document  d in D w.r.t
                query q using equation 1
                Sort documents in D according to their similarity measure with
                q
                Retrieve top n documents giving document collection R
                Find all unique terms of top n retrieved documents giving term
                collection T
                Find candidate expansion terms giving term collection C
                    (a)Calculate co-occurrence  between each query term qi and
                       each term ti in T using jaccard similarity (equation 2)
                    (b)Calculate similarity of entire query Q with each term ti
                       in T using equation 3
                    (c)Calculate the suitability score of each term ti using
                       equation 5
                    (d) Sort the terms in T on the basis of suitability score
                    (e )Retrieve top m terms of T giving candidate expansion
                       Collection  C
                Perform following to select expansion terms by applying  GA
                   Generate initial population randomly  from the term pool.
                   Repeat
                      Form new population using selection, crossover, mutation
                      operation(in   pair of 2)
                      Expand the original query by adding terms of the individual
                      population  member.
                      Retrieve the initial set of documents using tentatively
                      expanded       query
                      Calculate the fitness of the expanded query using recall based
                      measure  (equation 6)
                    Until the population converges or for maximum number of generation
                Return the terms obtained in final generation of GA as final set of
                expansion terms.

Output :  Set of expansion terms
```

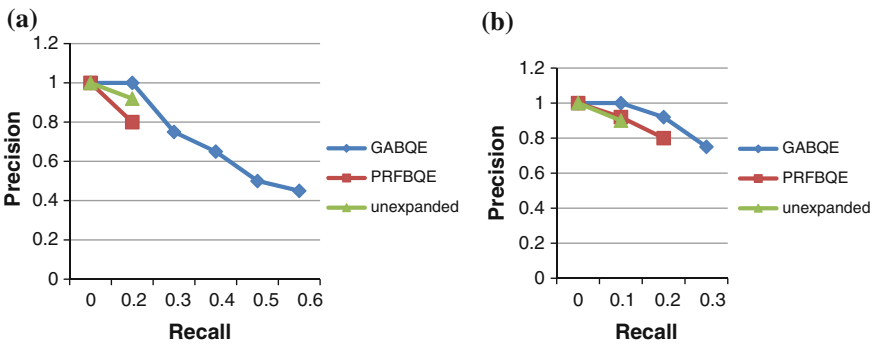

Fig. 1 **a** Recall precision graph for query 28. **b** Average recall precision graph

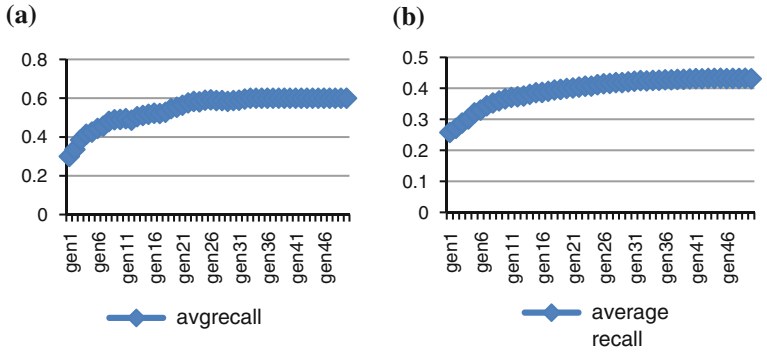

Fig. 2 **a** Generationwise average recall for query no. 28. **b** Generationwise average recall for all queries

increasing and slowly reaches to convergence. This shows that GA is able to improve the fitness (recall); hence, efficiency of information retrieval is increased.

In order to analyze the result we observed the expansion terms obtained without GA and with GA, we observed and analyzed expansion terms for individual query. For almost all the queries, GA-based expansion is providing better terms. Table 2 shows original query, query expansion terms obtained with PRFBQE, and query expansion terms obtained after applying GABQE. Due to space limitation, the result is presented for query number: 2, 11, 12, 23, and 28. The term in bracket indicates recall obtained. Highest recall is given in bold. Last row indicates average recall of all queries. For all the queries listed below as well as on average, our approach performs better. We observe that in case of expanding the query simply with PRFBQE, without applying GA, recall is increasing in some queries, while it is decreasing marginally in some cases. When applying GABQE, recall remains same or is increasing in almost all queries. However, for some queries, performance is detoriating as the term pool itself does not contain good candidate expansion terms. In general, we can see that GA effects query expansion positively; however, the actual effect may vary from query to query.

It can be observed that, in query 2, terms obtained from GABQE: 'search, semantic, retrieval citation, and file' are more focused and useful for expanding the query. Similarly, in query 28, 'asca' and 'sdi' are more related to query and are more focused terms. We observed that these terms have many meanings, but in this context, 'asca' is a terminology related to scientific classification taxonomy, whereas 'sdi.' comes from 'sdi biomed,' providing high-quality laboratory testing products used with laboratory chemical analyzers. Such observations can be made for other queries. So, we can say that application of GABQE helps in expanding the query in such a manner that it provides a better selection of thematically rich expansion terms and hence improves retrieval efficiency.

Table 2 Table showing query expansion terms and recall

Query no.	Actual query (recall)	Expansion terms and (recall) for PRFBQE	Expansion terms and (recall) for GABQE
2	How can actually pertinent data, as opposed to references or entire articles themselves, be retrieved automatically in response to information requests? (0.0769)	Available search sources produced retrieval methods source basis file (0.377)	File journals citation journal designed source semantics exact retrieval search **(0.385)**
11	What is the need for information consolidation, evaluation, and retrieval in scientific research? (0.1890)	System scientist methods science user document documents literature discussed analysis (0.1957)	Described user knowledge technical national subject scientists new analysis available **(0.2598)**
12	Give methods for high-speed publication, printing, and distribution of scientific journals. (0.4615)	Year citations papers work including articles able total abstracting primary (0.0769)	Journal citation multiple abstracting past coverage references highly source national **(0.6514)**
23	Amount of use of books in libraries. Relation to need for automated information systems. (0.2917)	Circulation journal designed people even available copies american articles stack (0.375)	Large, useful, material classification circulation, requirements longer universities available **(0.3958)**
28	Computerized information systems in fields related to chemistry. (0.2833)	Title considerably similarities estimating synthesis mathematics english citation cited alternative (0.2333)	Easily included asca chemical title sdi, estimating, alternative file compounds **(0.6)**
All queries	0.15	0.224	**0.326**

4 Conclusion

This paper suggests used of GABQE in order to improve retrieval efficiency of an information retrieval system. The experiments have been done on standard CISI collection. The comparison of the result has been done on the basis of recall. The results were compared for unexpanded query, PRFBQE and GABQE. It was observed that GA is providing a more cohesive and better selection of expansion terms. The improvement of the result can be observed from the graph. Further, we have also analyzed the result by observing the better expansion terms obtained by using our approach.

References

1. Xu, J.: Solving the Word Mismatch problem through Text analysis. Ph.D. Thesis, vol. 11, University of Massachusetts, Department of Computer Science, Amherst, USA (1997)
2. Xu, J., Croft, W.B.: Improving the effectiveness of information retrieval with local context analysis. ACM Trans. Inf. Syst. **18**(1), 79–112 (2000)
3. Cao, G., Nie, J.Y., Gao, J.F., Robertson, S.: Selecting good expansion terms for pseudo relevance feedback. In: 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 243–250 (2008)
4. Pathak, P., Gordon, M., Fan, W.: Effective information retrieval using genetic algorithm based matching functions adaption. In: Proceedings 33rd Hawai International Conference on Science (HICS), Hawaii, USA (2000)
5. Horng, J., Yeh, C.: Applying genetic algorithms to query optimization in document retrieval. Inf. Process. Manage. **36**, 737–759 (2000)
6. Araujo, L., Aguera J.P.: Improving query expansion with stemming terms: a new genetic algorithm approach. In: 8th European Conference on Evolutionary Computation in Combinatorial Explosion, pp. 182–193, Springer-Verlag Berlin, Heidelberg (2008)
7. Cecchini, R.L., Lorenzetti, C.M., Maguitman, A.G., Brignole, N.B.: Using genetic algorithms to evolve a population of topical queries. Inf. Process. Manage. **44**, 1863–1878 (2008)