International Journal of Information Retrieval Research
Volume 12 • Issue 2

A Firefly Algorithm-Based Approach for Web Query Reformulation

Meriem Zeboudj, SIMPA Laboratory, Computer Science Department, USTOMB, El Mnaouar, Bir El Djir, Oran, Algeria

Khaled Belkadi, SIMPA Laboratory, Computer Science Department, USTOMB, El Mnaouar, Bir El Djir, Oran, Algeria

ABSTRACT

A major difficulty in using a web-based information retrieval system is the choice of terms to be used for expressing and processing a query. The user has to examine a lot of data to find the necessary documents or information. The problem that often appears in this situation is that the query is incorrect and does not express those needs. Researchers have come up with various solutions to overcome this problem among them the use of query reformulation. This paper presents an approach called FA-QR based on this technique using the Firefly metaheuristic. This algorithm was applied to frequent itemsets generated by frequent- pattern growth (FP Growth). The algorithmic solution allowed the user to select the best path among all the possible solutions for the initial query. Experimentally, the results demonstrated that our proposed approaches achieved a significant improvement over other different methods on TREC and FIRE datasets.

KEYWORDS

Firefly Algorithm, Information Retrieval, Query Reformulation, Web-Search

INTRODUCTION

The information search on the Web engages the users in a questioning process about the choice of search engines. Besides, if the queries results do not express their needs or out of their objectives, this implies that some information is not well formulated. This leads us to ask two important questions: How can more pertinent documents be found to a given query? And, how can the user's query be better expressed to better meet one's needs?

The reformulation concept is both an iterative and an interactive process between the user and the search engines to achieve satisfactory results (Lu, Wei, Sun, Li, Wen, & Zhou, 2018). The retrieved results play a significant role in the reformulation strategy. The accuracy of the query suggestion phase is entirely based on either the results (of the documents or the URLs) and the new extracted keywords. This concept has been studied by many researchers to become one of the most known concepts in the fields of Information Retrieval (IR) since it has been the subject of a lot of works that provided solutions to users according to their information needs. The idea of web search improvement by query modification was studied in (Efthimiadis, 1996). Some improvement techniques involve

DOI: 10.4018/ijirr.299939

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

either adding to these queries the existing terms from linguistic resources as it is in the WordNet (Azad & Deepak, 2019), or building resources from the collections (Aminu, Oyefolahan, Abdullahi, & Salaudeen, 2019).

Another widely used and one of the most popular techniques is the one called Pseudo Relevance Feedback (PRF). It is based on the assumption that the ranked top-k documents are considered relevant to the query (Vaidyanathan, Das, & Srivastava, 2016; Xu, Lin, Lin, Yang, & Xu, 2018). For this reason, several papers were proposed to improve the classification of the document compared to a first search (Arampatzis, Peikos, & Symeonidis, 2021; Khennak & Drias, 2017a; Valcarce, Parapar, & Barreiro, 2019). In (Keikha, Ensan, & Bagheri, 2018), the authors have chosen Wikipedia as a source for extracting the relevant articles and used supervised and unsupervised methods for selecting the candidate expansion terms. Researchers in the paper (Azad & Deepak, 2019) used also Wikipedia combined with WordNet as data sources for expansion terms. The approach considers the individual terms and phrases as the expansion terms. This combination of the two data sources gave good results when compared to the two methods individually.

Nature-inspired Optimization Algorithms are well-reputed techniques that can solve efficiently difficult problems. However, some researchers investigated nature-inspired optimization approaches for query reformulation (Zeboudj & Belkadi, 2020), such as Singh, Garg, and Kaur (2016) that relied on them to provide a query suggestion by similarity assessment using the term graph. The graph was a built-in function of words in a user's query relevant documents. The association between the terms graph based on similarity considers the fitness values for the Genetic Algorithm. Veningston and Shanmugalakshmi (2014) used the same steps of the previous approach except at the level of the metaheuristic where they chose the Ant Colony algorithm. As well, at the traverse phase, the graph according to navigation history over web pages is retrieved for the user's query. Bhatnagar and Pareek (2015) used an approach called Genetic Algorithm-based query expansion (GABQE) to improve the Pseudo Relevance Feedback efficiency where the relevant terms are generated by the selection from an initially retrieved list of documents while the GA is applied to generate term combinations. Another approach (Al-Khateeb, Al-Kubaisi, & Al-Janabi, 2017) used the Genetic Algorithm to return more relevant results to the user through query reformulation. The approach is based on WordNet to select the proper meaning of keywords, Google search engine, and GA to select the best result from every generation. In (Sharma, Pamula, & Chauhan, 2019), the authors used cuckoo search and accelerated particle swarm optimization (PSO) to find the most relevant expanded query. They also used the Fuzzy logic technique to improve the performance of accelerated PSO by controlling various parameters. Another approach in this field is proposed by (Deulkar & Narvekar, 2015) who found the capacity of the Improved Memetic Algorithm (IMA) to retrieve the most relevant snippets just for the complex queries. The IMA is based on the hybrid-selection method where the random population is generated with the help of the initial population, and the chromosomes are also the same; however, the order is completely different in both. Consequently, the best was selected from them. Khennak and Drias applied swarm intelligence (SI) algorithms and Data mining techniques for web queries (Khennak & Drias, 2018). The SI used were the FireFly algorithm (Khennak & Drias, 2016), Bat Inspired Algorithm (Khennak & Drias, 2017b), and Accelerated Particle Swarm Optimization (Khennak & Drias, 2017a). This algorithm was used to extract the best expansion terms and select the best relevant documents which were carried out on the MEDLINE database. Approaches based on SI for solving the documents information retrieval (DIR) problem are also studied in papers (Bhopale & Tiwari, 2020; Djenouri, Belhadi, & Belkebir, 2018; Thirugnanasambandam, Anitha, Enireddy, Raghav, Anguraj, & Arivunambi, 2021).

Reformulating web queries is not a new idea in the Information Retrieval domain. Previous works in this discipline are very numerous, and they all tried to better find the pertinent documents. The significance of this work is the application of metaheuristic to solve the problem of query reformulation in this domain and produce an effective query in a shorter time by facilitating navigation tasks to Internet users. The proposed query reformulation approach called FA-QR is built essentially on the

FireFly Algorithm (FA) using new suggestions. With this algorithm, there is a guarantee to find the best reformulation keywords in a considerably shorter time. This algorithm is applied to the frequent itemsets generated by FP_Growth (frequent-pattern Growth), and each path (query) is considered to be a Firefly.

The paper carries on by first, presenting the basic Firefly Algorithm whereas the proposed query reformulation approach is then explained the process as well. Next, the results are provided illustrating the performance of the proposed approach with other methods, to finally conclude with a summary of all the work discussing some future directions.

BASIC FIREFLY ALGORITHM

Fireflies are small insects in the winged beetle family that help with their abdomen to produce a cool flashing light for mutual attraction. Females can mimic the flashing lights of other species to attract, capture or devour males. Fireflies have a capacitor-like mechanism that slowly discharges until it reaches a certain threshold and then releases energy in the form of light. This phenomenon is repeated cyclically (Yang, 2010).

The FireFly algorithm (FA) introduced by Xin-She Yang at Cambridge University in 2007 was inspired by the flashing behavior of firefly insect distance and mutual attraction; however, he considered all fireflies as unisex (Yang, 2010). This algorithm was proved to be effective to solve multimodal optimization problems (Yang, 2020). There have been significant developments since its emergence about ten years ago (Yang & He, 2018). Many research works were based on this algorithm to solve their problems. FireFly was used to solve the mono-processors hybrid flow shop problem (Dekhici & Belkadi, 2017), flexible operation scheduling (Fuyu, Weining, & Yan, 2018), facial expression recognition (Mistry, Zhang, Sexton, Zeng, & He, 2017), home care scheduling (Dekhici, Redjem, Belkadi, & Mhamedi, 2019), biomedical engineering (BME), healthcare (Nayak, Naik, Dinesh, Vakula, & Byomakesha, 2020), and also for image analysis (Dey, Chaki, Moraru, Fong, & Yang, 2020). Some researchers stated that FA was a powerful algorithm for solving even some of the NP-complete problems (Kumar & Kumar, 2021). The Firefly algorithm pseudo-code is given in Algorithm 1. Xin-She Yang took into consideration the following three rules:

- All fireflies are unisex that will be attracted to others regardless of their sex;
- The attraction of a firefly is proportional to the intensity of adjacent fireflies and decreases once the distance between two fireflies increases. If a firefly is not attractive except for a particular firefly, that firefly will move randomly;
- The light intensity of a firefly represents the objective function that determines the best path.

The FireFly algorithm takes into account the following parameters:

Attractiveness: The form of this function takes any monotone decreasing function, such as the following general form:

$$\beta_{i,j} = \beta_0^{*} e^{-\gamma r_{i,j}^{m}} \tag{1}$$

with β_0 representing the attractiveness at *r = 0*, *r* which represents the distance between two fireflies, and γ represents the constant coefficient of light absorption.

Distance: The distance between 2 fireflies i and j at x_i and x_j is determined by the Cartesian distance as follows:

$$r_{i,j} = \sqrt{\sum_{k=1}^{q} \left(x_{i,k} - x_{j,k} \right)^2} \quad (2)$$

Movement: The movement of firefly by another more luminous firefly j is defined by:

$$x_i = \left(1 - \beta_{i,j}\right) x_i + \beta_{i,j} x_j + \alpha \left(rand - \frac{1}{2} \right) \quad (3)$$

With the first and the second terms in the equation due to the attraction so that the third term is randomization. α is the random parameter (can be constant) and “rand” is a uniform random number generator distributed in the interval [0, 1].

Algorithm 1. FireFly Algorithm

```
Define the light absorption coefficient γ and parameter α
Generate an initial population of fireflies
Determine the light intensities $I_i$ of $x_i$ by $f(x_i)$
While (t < Max number of iteration) do
    For i = 1 to  n  do   // all fireflies
    For j=1 to n do      // all fireflies
        If ($I_i$ < $I_j$) then
            Vary the attractiveness $\beta_{i,j}$ according to the distance $r_{i,j}$
            Move firefly i to j with attractiveness $\beta_{i,j}$
        Else move I randomly
        End If
        Evaluate the new solution
        Update Intensity $I_i$
        Check if firefly i is the best
    End j, End i
end While
```

FIREFLY ALGORITHM BASED QUERY REFORMULATION

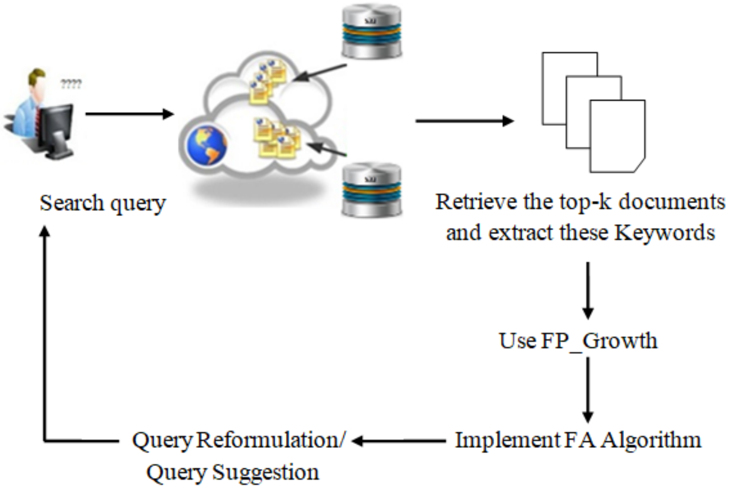
FireFly Algorithm-based query reformulation approach (FA-QR) is provided for users to facilitate their navigation tasks while optimizing them. The proposed approach is inspired by the flashing behavior of firefly insects called the Firefly Algorithm (FA).

As regards the arrival of a user’s query to the web-based engine, the first step is to retrieve relevant documents (document, URL, etc.). A preprocessing phase that removes all StopWord (noisy or irrelevant words) is used later. As a result, the useful terms will be annotated with information on the type of words (names, verbs, infinitives, and participles) and lemmatization information using treeTagger for taking root form. In the second step, the extracted terms (keywords) of these documents are associated with FP_Growth (frequent-pattern Growth) to generate frequent itemsets. Each of these itemsets is considered to be an FA path. The third step allows the algorithm to converge to top-k optimal paths, and the keywords present in the path will be suggested for the user in the last step. The steps of this approach are illustrated in Figure 1.

FP_Growth

Based on the extracted terms of top_k documents, the terms are associated with FP_Growth to build the frequent itemsets. The FP_Growth uses the ‘divide-and-conquer’ strategy in order first to compress

Figure 1. Proposed FireFly Algorithm-based query reformulation (FA-QR)


????
Search query
Retrieve the top-k documents and extract these Keywords
Use FP_Growth
Implement FA Algorithm
Query Reformulation/ Query Suggestion

the database (DB) representing the frequent itemsets in a compact structure called FP-tree (Frequent Pattern tree). These branches have all possible associations of items. Each of these associations can be broken down into fragments (fragment patterns) that establish the frequent itemsets. The FP-Growth algorithm is divided into three main steps (Afuan, Ashari, & Suyanto, 2019), namely:

The phase of generating a conditional pattern base: the entire database will be compressed into a smaller data structure (FP-tree), and then the conditional pattern base will be generated. Each conditional pattern of this base admits a prefix path and a suffix pattern.

The phase of generating a conditional FP-Tree: this phase is intended for the generation of the conditional FP-tree to mine frequent itemsets. The support count of each item in each conditional pattern base is added together. Each item that has a number of supports counting greater or equal to the minimum support count (defined by the user) will be generated with a conditional FP-Tree.

The phase of searching frequent itemsets: in this last phase, if Conditional FP-Tree is a single trajectory, and then frequent itemsets will be obtained by combining all items for each conditional FP-Tree. If it is not a single path, then a recursive generation of FP-Growth will be carried out.

FP_Growth algorithm is used to generate all paths containing frequent itemsets. To minimize computation time and give reformulation query more adequately, the authors selected only paths admitting up to five terms (Frequent pattern). These paths are considered to be an entry for the Firefly algorithm.

Application of the Firefly Algorithm

During query Reformulation, the central decision is the determination of the suitable representation scheme for candidate reformulation terms (solution). The solution is represented by a terms vector (keywords), which is defined as:

$$Q_F = \left\langle t_1, t_2, \ldots t_n \right\rangle \tag{4}$$

where:

$1 \leq \text{n} \leq 5$

The size of the population determines the number of solutions or the size of the search space (R) whose goal is to direct the search to the best location (best query). In our case, the search space contains terms related to information (frequent items generated by FP_Growth) as well as their frequencies. For p items, one has 2^p possible itemsets.

Each firefly ‘i’ in this population is a keywords vector ($Q_I^{'} = \left\langle t'_1, t'_2, \ldots t'_n \right\rangle$ where I is the number of frequent items container at most 5 items) randomly selected from the set of paths obtained by FP_Growth. Therefore, the firefly position can be represented by $x_{i,n} = \{w_{i,1}, w_{i,2,} \ldots w_{i,n}\}$, where $w_{i,n}$ is the query weight ‘i’(these keywords) in search space.

For each firefly in the population, a value indicating its light intensity (fitness) will be associated. The efficiency of the algorithm in terms of the relevance of the solution expressed by query reformulation and the calculation time depends mainly on this value. For this OkapiBM25 was chosen as fitness which is a method of weighting terms in documents and queries according to the probabilistic relevance model developed by Robertson and Sparck Jones in the ‘Okapi’ information system at the University of London (Robertson & Walker, 1994). Therefore, for the set query $Q_I^{'}$, equation 5 is used to measure its fitness:

$$w_i^{BM25}\left(Q_I^{'}, t_n, D_k\right) = \log\frac{K - n_l + 0,5}{n_l + 0,5} \times \left[\sum_{t\in Q_I^{'}} \frac{\left(k_1 + 1\right)tf}{k_1\left(1 + \frac{b\left(dl - avgl\right)}{avgl}\right) + tf}\right] \tag{5}$$

where:

- *K* is the number of documents in the set collection;
- n_l is the number of documents containing t_n;
- *tf* is the frequency of the term t_n in each document D_k;
- k_1 *and* b, are constants;
- *dl* is the document length;
- *avdl* is the average document length.

Following the experiments carried out on the TREC-7 collection (Robertson, Walker, & Beaulieu, 1999), ZF109, CR93H (other Sub-collection of TREC), and CACM corpus “Communications of the Association for Computing Machinery”, the values of the BM25 constants are defined as $k_1 = 1{,}2$ and $b = 0{,}75$.

The best query reformulation Q_b is presented as the firefly with maximum light intensity in all the fireflies:

$$f\left(Q_F\right) = \max\left(w_i^{BM25}\left(Q_I^{'}, t_n, D_k\right)\right) \tag{6}$$

After comparing the brightness of each firefly $Q_I^{'}$, with all other fireflies, the positions of the fireflies were updated based on displacement rules and their neighbors. These rules are usually the initial position, the distance between two fireflies, and a random move.

Table 1. Term frequency of $Q_1^{'}$ and $Q_2^{'}$

	t'_1	t'_2	t'_3	t'_4	t'_5
$Q_1^{'}$	1	1	0	1	1
$Q_2^{'}$	0	0	1	1	1

To adapt an efficient FA to the problem of query reformulation, some modifications to these rules were made. At the level of attractiveness (equation 1), the Manhattan distance was used to find the distance between two fireflies $Q_1^{'}$ and $Q_2^{'}$ by equation 7. The choice of this distance is very important because once its value decreases, attractiveness increases:

$$ManhattanDist = \sum_{n=1}^{I} \left| x_{i,n} - x_{j,n} \right| \tag{7}$$

$x_{i,n}$ is term weighting n in $Q_i^{'}$ and $x_{j,n}$ is term weighting n in $Q_j^{'}$. The result value is possible to be more than 1.

An example that calculates this distance between $Q_1^{'} = \left\langle t'_1, t'_2, t'_4, t'_5 \right\rangle$ and $Q_2^{'} = \left\langle t'_3, t'_4, t'_5 \right\rangle$ is the following:

$$ManhattanDist\left(Q_1^{'}, Q_2^{'}\right) = \left|1-0\right| + \left|1-0\right| + \left|0-1\right| + \left|1-1\right| + \left|1-1\right| = 3$$

During the two-to-two comparison loop, the best solution $(Q)_b$ was iteratively updated. The peer comparison process has been repeated until the stop condition is satisfied. This condition corresponds in our case to a maximum number of iterations at the start or when the results of the algorithm become stable, i.e. the query does not change to avoid wasting time.

The main steps of the proposed FireFly algorithm for query reformulation are shown in Algorithm 2.

Algorithm 2. Proposed FireFly Algorithm

```
Define the light absorption coefficient g and parameter $\alpha$
Generate an initial population of fireflies $Q_I^{'}$
Determine the light intensities by Equation 6
While ( t < Max number of iteration and the best-reformulated
query still changes) do
    For i = 1 to  N  do   // all fireflies
    For j=1 to N do      // all fireflies
        If (f($Q_i^{'}$ )<f ($Q_j^{'}$)) then
            Vary the attractiveness $b_{i,j}$ according to the distance
            $r_{i,j}$ (Equation 7)
            Move query $Q_i^{'}$ to $Q_j^{'}$ with Equation 1
        Else move I randomly
        End If
        Evaluate the new solution
        Update Intensity $I_i$
        Check if query $Q_I^{'}$ is the best-reformulated query $Q_b$
    End j, End i
end While
```

EXPERIMENTS AND DISCUSSION

Datasets

The experiment was performed on two different English test collections: Text Retrieval Conference TREC-3 (disks 1 and 2) (Test Collections, 2020) and Forum for Information Retrieval Evaluation FIRE 2011 Ad-hoc (Data, 2020). Those collections are used to have sufficient training data because they contain a very large number of topics (each collection contains 50 topics) and a set of documents on which IR is done. The detailed descriptions of both datasets are presented in Table 2.

Specifically, the TREC-3 benchmark datasets include newswire articles from various sources, such as Wall Street Journal, Federal Register, Association Press, and Financial Times. These sources are judged as high-quality text data with minimum noise. The FIRE ad hoc dataset is a medium-size collection containing newswire articles from two sources named The Telegraph and BD News 24 offered by the Indian Statistical Institute, Kolkata, India.

Evaluation Metrics

Generally, any Information Retrieval System (IRS) has two main purposes: finding all relevant documents and ignoring any irrelevant ones. To better approach these objectives, an approach based on the techniques of relevance feedback was used to formulate the query by providing a new suggestion.

The most known performance measurement criteria are Recall (R) and Precision (P). Searching was done using the expanded query with the help of Google Search API (API Google Web Service) (Google Developers, 2017). The performance measures of precision and recall were depicted in equations 8 and 9:

$$R = \frac{Number\ of\ relevant\ documents\ given\ by\ the\ system}{Total\ number\ of\ relevant\ documents} \tag{8}$$

$$P = \frac{Number\ of\ relevant\ documents\ given\ by\ the\ system}{Total\ number\ of\ documents\ retrieved} \tag{9}$$

The recall (R) measures the proportion of relevant documents returned by the system among all those relevant to the query, and the precision (P) measures the proportion of relevant documents among all those returned by the system. (P@X) measure the precision when X documents are retrieved. X

Table 2. Details of alls used datasets and query numbers

Datasets	Task	Size	Query Numbers	Docs
TREC-3	Ad hoc	6 Gb	151-200	7,41,856
FIRE 2011	Ad hoc	1.76 Gb	126-175	3,92,577

denotes the proportion of relevant documents in the top X documents in the returned list for given queries. X is set to 10, 20, and 30, respectively.

The average precision (AP) is used to evaluate the performance of a search system in information retrieval. The AP measures the area underneath the entire recall precision.

Also, the F-measure is found which is a measure that combines precision, recall, and weighting. This measure assesses the overall IRS performance and is calculated as follows:

$$F = \frac{2 \times P \times R}{P + R} \tag{10}$$

The purpose of this work is to present a simple form of information to the user to select the adequate terms for query reformulation. For this reason, a new approach for web search query reformulation using FireFly metaheuristics is proposed. To evaluate the performance of this proposed approach, it is recommended to compare it with recent query reformulation approaches based on the same algorithms or methods. Despite using the same data collection and the same approaches, contradictions in results were detected preventing a fair comparison due to the use of a large variety of configuration parameters like stop words filtering, ranking models, etc.

Therefore, the proposed method FA-QR was compared with the baseline approach and Lucene algorithms and also for the different metaheuristics, to name particle swarms optimization (PSO-QR), genetic algorithms (GA-QR), and Bat algorithm (Bat-QR).

Parameter Settings

Based on these preliminary experiments, the values of the FA-QR parameters used in all the tests to solve the web query reformulation problem are given in Table 3.

RESULTS AND DISCUSSION

Tables 4 and 5 display the retrieval performance of FA-QR in all top retrieved documents (P@X) in terms of average precision and average recall on FIRE and TREC datasets based on the 100 queries used (each collection contains 50). X denotes the proportion of relevant documents in the top X documents in the returned list for given queries. X is set to 10, 20, and 30, respectively.

Table 3. FA-QR parameter settings

	Parameters	Values
FP_Growth	Minimum support $\min\_\sup$	1
Firefly Algorithm	Light absorption coefficient γ	1
	Initial randomization parameter α	0,20
	Population size	50
	Number of generations	25
	Fitness: $'OkapiBM25'$	
	K1 b	1,2 0,75

Table 4. Comparison of different query reformation methods in terms of average precision for the TREC-3 dataset

Methods	P@10		P@20		P@30	
	Average precision	Average recall	Average precision	Methods	Average precision	Average recall
FA-QR	0,1836	0,1707	0,1986	FA-QR	0,1836	0,1707
Bat-QR	0,1694	0,1671	0,1720	Bat-QR	0,1694	0,1671
PSO-QR	0,1634	0,1571	0,1826	PSO-QR	0,1634	0,1571
GA-QR	0,1653	0,1669	0,1750	GA-QR	0,1653	0,1669
Lucene	0,1399	0,1392	0,1419	Lucene	0,1399	0,1392
Baseline	0,1314	0,1244	0,1376	Baseline	0,1314	0,1244

Table 5. Comparison of different query reformulation methods in terms of average precision for the FIRE-2011 dataset

Methods	P@10		P@20		P@30	
	Average precision	Average recall	Average precision	Methods	Average precision	Average recall
FA-QR	0,17488	0,17433	0,18151	FA-QR	0,17488	0,17433
Bat-QR	0,16932	0,15925	0,17725	Bat-QR	0,16932	0,15925
PSO-QR	0,16801	0,16015	0,17451	PSO-QR	0,16801	0,16015
GA-QR	0,16571	0,15880	0,16297	GA-QR	0,16571	0,15880
Lucene	0,13251	0,12924	0,13801	Lucene	0,13251	0,12924
Baseline	0,12778	0,11837	0,13266	Baseline	0,12778	0,11837

Tables 6 and 7 tabulate the different values of the mean average precision (MAP) and the improvement rate on the TREC and FIRE datasets, respectively. For each query, the MAP and the improvement rate compared to the baseline (MAP-Gain) were calculated.

Figures 2 and 3 show the significant improvement by Precision, Recall, and F-measures the values of all individual approaches datasets for X=10, 20, and 30 on both FIRE and TREC datasets, respectively.

From the results, the authors determine that the performance of our proposed FA-QR had an important improvement over the other different techniques and on every baseline approach. The proposed approach performed better in retrieving relevant results. On the TREC-3 dataset, it achieved 42,56% improvements as compared to Bat-QR with 31,32%, PSO-QR with 30,45%, GA-QR with 28,83%, and Lucene with 5,75%. Also compared to Lucene where 21,82%, 23,34%, and 24,17% were achieved on GA-QR, PSO-QR, and Bat-QR, respectively as compared to our proposed query reformulation FA-QR with 34,80% results. On the FIRE-2011 dataset, FA-QR achieved 38,47% improvements as compared to Bat-QR with 33,48%, PSO-QR with 31,69%, GA-QR with 28,42%, and Lucene with 3,85%. Again as compared to Lucene where 17,16%, 20,15%, and 21,78% were achieved on GA-QR, PSO-QR, and Bat-QR, respectively as compared to our proposed query reformulation FA-QR with 26,33% results.

Also, the authors noticed that the improvements achieved by the proposed approach on TREC-3 are a little bit greater than the FIRE 2011 dataset unlike for PSO-QR and Bat-QR. This is probably due to the fact they include news articles commonly considered as high-quality text data with less

Table 6. Comparison of different query reformulation methods to the baseline on the TREC-3 datasest

Methods	MAP	MAP-Gain
FA-QR	0,1937	42,565%
Bat-QR	0,1784	31,325%
PSO-QR	0,1773	30,451%
GA-QR	0,1751	28,837%
Lucene	0,1437	5,757%
Baseline	0,1359	

Table 7. Comparison of different query reformulation methods to the baseline on the FIRE-2011 dataset

Methods	MAP	MAP-Gain
FA-QR	0,1815	38,470%
Bat-QR	0,1750	33,482%
PSO-QR	0,1727	31,695%
GA-QR	0,1684	28,424%
Lucene	0,1362	3,854%
Baseline	0,1311	

noise. On the contrary, the FIRE ad hoc dataset includes news as well as web collections that are more challenging and contain multiple sources of a heterogeneous set of documents as well as more noise. The improvement rate on the TREC-3 dataset is 42,56% than on the baseline; on the other hand, on FIRE 2011 it is 38,47%.

According to the results obtained in the comparative tables (Table 6 and Table 7), there is an improvement in accuracy with FireFly compared to the other algorithms, i.e. on the TREC-3 dataset the FA-QR provides 0,193 in comparison to other approaches; while in the second dataset, it provides 0.181.

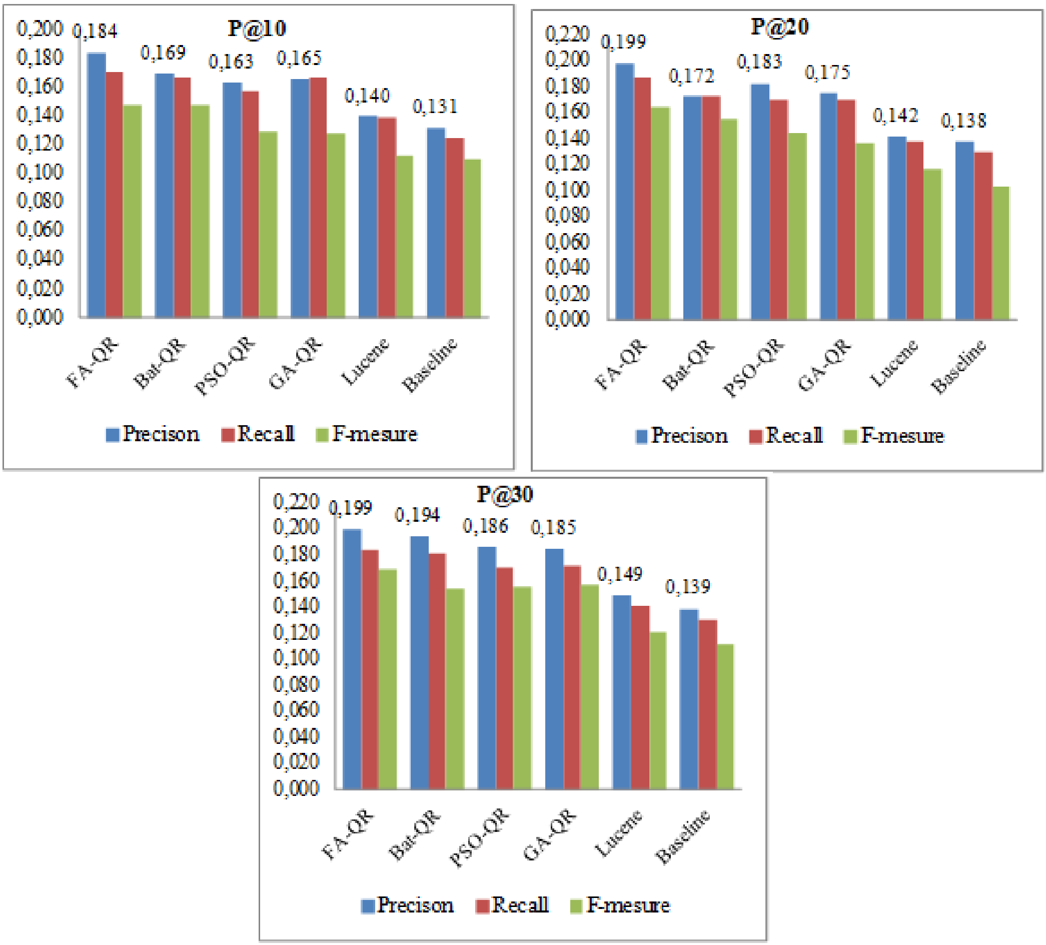
Moreover, for metaheuristics methods, the results obtained have some similarities on the Recall and F-measure (see Figures 2 and 3). However, Precision to some extent has little difference. The FireFly algorithm is better at retrieving relevant results. It improves search performance by achieving 0,199 as compared to the Bat method with 0,194, PSO with 0,186, and GA with 0,185 on TREC-3 for p@30, and the same for the FIRE dataset. On the other hand, the recall archives 0,180 on FireFly as compared to the Bat method with 0,180, PSO and GA with the same value 0,171. The F-measure archives 0,169 on FireFly as compared to the Bat methods, PSO, and GA with 0,154; 0,155 and 0,156 respectively.

The obtained results also showed that the system accuracy can be enhanced when taking into consideration the top 30 returned results compared to 10 and 20 on all approaches on both datasets, and it probably comes up that when there are a lot of document means, there are a lot of terms which allow the query enrichment.

Consequently, as shown with the results obtained above, the use of the Firefly algorithm for web-query reformulation improves retrieval effectiveness.

Globally, the proposed FireFly Algorithm based approach for query reformulation approach carries the following properties:

Figure 2. Precision, Recall, and F-measures values of all individual approaches for all top retrieved documents on the TREC dataset

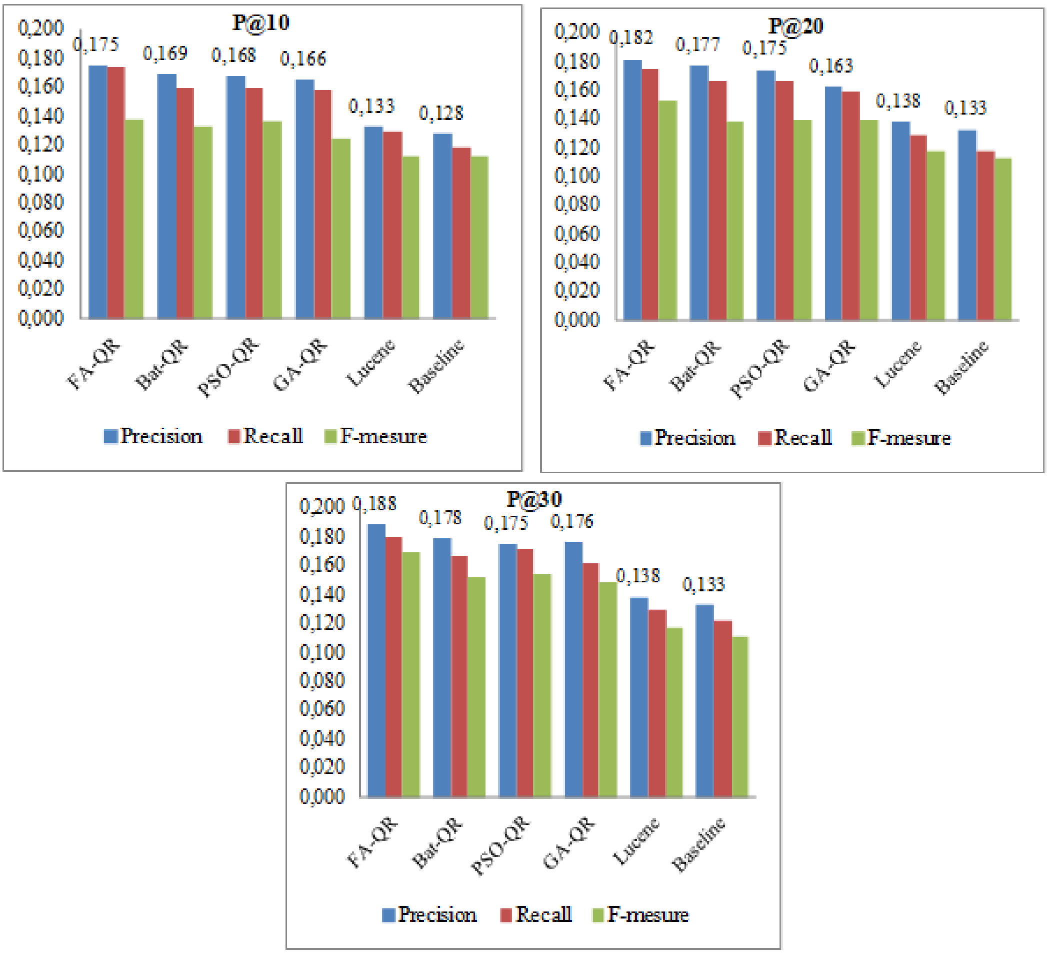


- Query precision is one of the main factors used to measure query quality.
- The FP_Growth algorithm retrieves all frequent itemsets which support exceeding pre-fixed thresholds (min_sup). Its first phase is the most expensive in terms of execution time which may cause a disadvantage for users because the number of these frequent items depends exponentially on the number of items handled (for p items, one has 2^p possible itemsets). For this step to be executable, it is sometimes necessary to increase the support threshold which gives a central role to the support condition making the extraction of items more difficult.
- The Google APIs are very useful for testing but remain insufficient in the evaluation phase which would be more subjective with the other test collections. Nevertheless, the evaluations taken into account remain credible to prove the effectiveness of results in the context of Information Retrieval on the web.

CONCLUSION

In this work, the authors presented an approach for query reformulation based on the FireFly Algorithm by providing a new suggestion called FA-QR. This algorithm is applied to frequent itemsets generated

Figure 3. Precision, Recall, and F-measures values of all individual approaches for all top retrieved documents on the FIRE dataset



by the FP_Growth. The authors explained and worked on the performance of the FA-QR method compared to the baseline approach and Lucene algorithms, also for the different metaheuristics: particle swarms optimization, genetic algorithms, and Bat algorithm. The experiments of the proposed approach were performed on two datasets: FIRE and TREC. The results showed that there is an improvement at the level of means average precision. For future work, the authors are trying to integrate ontology on query reformulation and the user's profile. Moreover, the proposed approach will be entered for a dedicated project specifically for visually impaired people.

ACKNOWLEDGMENT

The authors would like to thank Yamina Houti -Senior lecturer - English teacher at Higher School of Electrical and Energetic Engineering of Oran (ESGEE) for her help during this research paper.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Afuan, L., Ashari, A., & Suyanto, Y. (2019). Query Expansion in Information Retrieval using Frequent Pattern (FP) Growth Algorithm for Frequent Itemset Search and Association Rules Mining. *International Journal of Advanced Computer Science and Applications*, *10*(2), 263–264. doi:10.14569/IJACSA.2019.0100235

Al-Khateeb, B., Al-Kubaisi, A. J., & Al-Janabi, S. T. (2017). *Query reformulation using WordNet and genetic algorithm. In New Trends in Information & Communications Technology Applications (NTICT)*. IEEE.

Aminu, E. F., Oyefolahan, I. O., Abdullahi, M. B., & Salaudeen, M. T. (2019). Enhanced query expansion algorithm: Framework for effective ontology based information retrieval system. i-Manager's. *Journal of Computational Science*, *6*(4), 1–11.

Arampatzis, A., Peikos, G., & Symeonidis, S. (2021). Pseudo relevance feedback optimization. Information Retrieval Journal. doi:10.1007/s10791-021-09393-5

Azad, H. K., & Deepak, K. (2019). A New Approach for Query Expansion using Wikipedia and WordNet. *Information Sciences*, *492*, 147–163. doi:10.1016/j.ins.2019.04.019

Bhatnagar, P., & Pareek, N. (2015). Genetic algorithm-based query expansion for improved information retrieval. Intelligent Computing. *Communication and Devices*, *308*, 47–55.

Bhopale, A. P., & Tiwari, A. (2020). Swarm optimized cluster based framework for information retrieval. *Expert Systems with Applications*, *154*, 1–16. doi:10.1016/j.eswa.2020.113441

Data. (2020). *Forum for Information Retrieval Evaluation*. http://fire.irsi.res.in/fire/static/data

Dekhici, L., & Belkadi, K. (2017). A Firefly Algorithm for the Mono-Processors Hybrid Flow Shop Problem. *International Journal of Advanced Computer Science and Applications*, *8*(12), 424–433. doi:10.14569/IJACSA.2017.081256

Dekhici, L., Redjem, R., Belkadi, K., & Mhamedi, A. E. (2019). Discretization of the Firefly Algorithm for Home Car. *Canadian Journal of Electrical and Computer Engineering*, *42*(1), 20–26. doi:10.1109/CJECE.2018.2883030

Deulkar, K., & Narvekar, M. (2015). An Improved Memetic Algorithm for Web Search. *Procedia Computer Science*, *45*, 52–59. doi:10.1016/j.procs.2015.03.083

Dey, N., Chaki, J., Moraru, L., Fong, S., & Yang, X. (2020). Firefly Algorithm and Its Variants in Digital Image Processing: A Comprehensive Review. In N. Dey (Ed.), *Applications of Firefly Algorithm and its Variants* (pp. 1–28). Springer. doi:10.1007/978-981-15-0306-1_1

Djenouri, Y., Belhadi, A., & Belkebir, R. (2018). Bees swarm optimization guided by data mining techniques for document information retrieval. *Expert Systems with Applications*, *94*, 126–136. doi:10.1016/j.eswa.2017.10.042

Efthimiadis, E. N. (1996). Query Expansion. Annual Review of Information Systems and Technology, 31, 121-187.

Fuyu, W., Weining, L., & Yan, L. (2018). Variable neighborhood improved firefly algorithm for flexible operation scheduling problem. *U.P.B. Sci. Bull., Series C*, *80*(2), 41–56.

Google Developers. (2017). *Custom Search JSON API, Custom Search*. https://developers.google.com/custom-search/json-api/v1/overview

Keikha, A., Ensan, F., & Bagheri, E. (2018). Query expansion using pseudo relevance feedback on wikipedia. *Journal of Intelligent Information Systems*, *50*(3), 455–478. doi:10.1007/s10844-017-0466-3

Khennak, I., & Drias, H. (2016). A Firefly Algorithm-based Approach for Pseudo-Relevance Feedback: Application to Medical Database. *Journal of Medical Systems*, *40*(240), 240. doi:10.1007/s10916-016-0603-5 PMID:27679449

Khennak, I., & Drias, H. (2017a). An accelerated PSO for query expansion in web information retrieval: Application to medical dataset. *Applied Intelligence*, *47*(3), 793–808. doi:10.1007/s10489-017-0924-1

Khennak, I., & Drias, H. (2017b). Bat-Inspired Algorithm Based Query Expansion for Medical Web Information Retrieval. *Journal of Medical Systems*, *41*(2), 34. doi:10.1007/s10916-016-0668-1 PMID:28054196

Khennak, I., & Drias, H. (2018). Data mining techniques and nature-inspired algorithms for query expansion. In *International Conference on Learning and Optimization Algorithms: Theory and Applications (LOPAL '18).* Association for Computing Machinery. doi:10.1145/3230905.3234631

Kumar, V., & Kumar, D. (2021). A Systematic Review on Firefly Algorithm: Past, Present, and Future. *Archives of Computational Methods in Engineering*, *28*(4), 326–329. doi:10.1007/s11831-020-09498-y

Lu, J., Wei, Y., Sun, X., Li, B., Wen, W., & Zhou, C. (2018). Interactive Query Reformulation for Source-Code Search With Word Relations. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 75660–75668. doi:10.1109/ACCESS.2018.2883963

Mistry, K., Zhang, L., Sexton, G., Zeng, Y., & He, M. (2017). *Facial expression recognition using firefly-based feature optimization. In IEEE congress on evolutionary*. IEEE.

Nayak, D. J., Naik, B., Dinesh, P., Vakula, K., & Byomakesha, P. (2020). Firefly Algorithm in Biomedical and Health Care: Advances, Issues. *SN Computer Science*, *1*(6), 1–36. doi:10.1007/s42979-020-00320-x PMID:33063057

Robertson, S., & Walker, S. (1994). Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In *17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval* (pp. 232-241). Springer-Verlag. doi:10.1007/978-1-4471-2099-5_24

Robertson, S., Walker, S., & Beaulieu, M. (1999). Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive track. *The 7th Text Retrieval Conference (TREC-7),* 253.

Sharma, D. K., Pamula, R., & Chauhan, D. S. (2019). A hybrid evolutionary algorithm based automatic query expansion for enhancing document retrieval system. *Journal of Ambient Intelligence and Humanized Computing*. Advance online publication. doi:10.1007/s12652-019-01247-9

Singh, V., Garg, S., & Kaur, P. (2016). Efficient Algorithm for Web Search Query reformulation Using Genetic Algorithm. *Springer India 2016, Computational Intelligence in Data Mining, part of the Advances in Intelligent Systems and Computing book series (AISC, 410), 1*, 459-470.

Test Collections. (2020). *In-Text Retrieval Conference (TREC)*. https://trec.nist.gov/data/test_coll.html

Thirugnanasambandam, K., Anitha, R., Enireddy, V., Raghav, R. S., Anguraj, D. K., & Arivunambi, A. (2021). Pattern mining technique derived ant colony optimization for document information retrieval. *Journal of Ambient Intelligence and Humanized Computing*. Advance online publication. doi:10.1007/s12652-020-02760-y

Vaidyanathan, R., Das, S., & Srivastava, N. (2016). Query Expansion based on Central Tendency and PRF for Monolingual Retrieval. *International Journal of Information Retrieval Research*, *6*(4), 30–50. doi:10.4018/IJIRR.2016100103

Valcarce, D., Parapar, J., & Barreiro, Á. (2019). Document-based and term-based linear methods for pseudo-relevance feedback. *Applied Computing Review*, *18*(4), 5–17. doi:10.1145/3307624.3307626

Veningston, K., & Shanmugalakshmi, R. (2014). Efficient Implementation of Web Search Query Reformulation Using Ant Colony Optimization. *International Conference on Big Data Analytics*, 80-94. doi:10.1007/978-3-319-13820-6_7

Xu, B., Lin, H., Lin, Y., Yang, L., & Xu, K. (2018). Improving Pseudo-Relevance Feedback With Neural Network-Based Word Representations. *IEEE Access : Practical Innovations, Open Solutions*, *6*, 62152–62165. doi:10.1109/ACCESS.2018.2876425

Yang, X. (2010). Firefly Algorithm. In X. Yang (Ed.), *Nature-Inspired Metaheuristic Algorithms* (pp. 81–89). Luniver Press.

Yang, X., & He, X. S. (2018). Why the firefly algorithm works? In Nature-Inspired Algorithms and Applied Optimization (pp. 245-259). Springer International Publishing AG.

Yang, X. S. (2020). Firefly Algorithm: Variants and Applications. In Swarm Intelligence Algorithms (p. 12). CRC Press.

Zeboudj, M., & Belkadi, K. (2020). Web Query Reformulation Using FireFly Algorithm. In *2020 Second International Conference on Embedded & Distributed Systems EDiS* (pp. 87-90). Oran, Algeria: IEEE. doi:10.1109/EDiS49545.2020.9296463

Meriem Zeboudj is PhD student in Computer Engineering specialty at the USTO-MB (University of Sciences and Technology of Oran, Mohamed Boudiaf). His research is based on information retrieval, web query reformulation, optimization, metaheuristics, and web accessibility.

Khaled Belkadi is professor at the USTO-MB (University of Sciences and Technology of Oran, Mohamed Boudiaf). He obtained a doctorate in Computer Science in Clermont-Ferrand (France). He obtained a PhD in Computer Science in Oran (Algeria). He has been a member of the Scientific Council of the USTO-MB School of Mathematics and Computer Science. K. Belkadi has contributed extensively through his many publications to the fields of modeling, simulation, optimization and performance evaluation of manufacturing systems and in particular of hospital systems.