SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2018, Vol. 61 122102:1–122102:16
https://doi.org/10.1007/s11432-017-9343-5

LionRank: lion algorithm-based metasearch engines for re-ranking of webpages

P. VIJAYA* & Satish CHANDER

Waljat College of Applied Sciences, Seeb 121*, Sultanate of Oman*

Received 18 October 2017/Revised 28 November 2017/Accepted 22 May 2018/Published online 20 November 2018

Abstract Due to the rapid growth of the web, the process of collecting the relevant web pages based on the user query is one of the major challenging tasks in recent days. Hence, it is very complicated for the users to know the most relevant information even though various search engines are widely employed. To deal with users' trouble in identifying the relevant information from the web, we have proposed a meta-lion search engine to capture and analyze the ranking scores of various search engines and thereby, generate the re-ranked score results. Accordingly, LionRank, a lion algorithm-based meta-search engine is proposed for the re-ranking of the web pages. Here, different features like text based, factor based, rank based and classifier based features are used by the underlying search engines. In classifier based feature extraction, we have used the fuzzy integrated extended nearest neighbor (FENN) classifier to include the semantics in feature extraction. Moreover, an intelligent re-ranking process is proposed based on the lion algorithm to fuse the features scores optimally. Finally, the results of the proposed LionRank is analyzed with the web page database collected through four benchmark queries, and the quantitative performance are analyzed using precision, recall, and F-score. From the results, we proved that the proposed LionRank obtained the maximum F-score of 81% as compared with that of existing search engines like QuadRank, Outrank, Google, Yahoo, and Bing.

Keywords web technology, meta-search engine, feature extraction, lion algorithm, FENN classifier

Citation Vijaya P, Chander S. LionRank: lion algorithm-based metasearch engines for re-ranking of webpages. Sci China Inf Sci, 2018, 61(12): 122102, https://doi.org/10.1007/s11432-017-9343-5

1 Introduction

The size of the world wide web (WWW) is extremely huge, which contains billions of openly observable web documents [1] that can be spread over the millions of servers worldwide. Therefore, the quantity of information available in the web is growing at a very quick rate [2]. Due to the increasing growth of the web, most search engines are incapable of directing a huge sufficient portion of the obtainable web pages. So, the main challenge is to find the desired data on the web in a timely and cost effective way. To deal with this challenge, more efficient web page retrieval algorithms are used by several researchers [3]. The process of retrieving information from the web pages is called document retrieval and its main aim is to recover the applicable documents from the massive collection of documents based on the user query. To retrieve the relevant web pages, various web page retrieval methods are used. The search engines are used to collect the relevant information from the web with respect to the user query.

A metasearch engine is a viable solution for the retrieval of a document from the extended list of documents. Meta-search engines are used by various researchers to combine a number of search results

* Corresponding author (email: pvvijaya@gmail.com)

from multiple sources and produce a major development in searching efficiency. In addition, the meta-search engines [4,5] are used to provide improved coverage of the web than any individual search engine. They calculate ranking scores [6] of various search engines for every query corresponding to the precedent retrieval knowledge of using those terms in the user query. In the web page retrieval process, the information collected from the individual rankers is plentiful. Rank aggregation [6] is a commonly used technique in metasearch engine applications [7–9], due to the frame work of WWW. Re-ranking [10–14] is performed in the meta-search engine [15, 16] by combining the results of underlying search engines.

This paper aims to design and develop a meta-lion search engine based on the lion algorithm [17]. Here, the proposed meta-lion search engine is used to collect the ranking scores of various search engines and generate the re-ranked results as output. When the user gives the query, a large number of web pages are developed by various search engines. We have proposed the hybrid feature extraction process, in which the features are extracted based on various methods such as text based, factor based, rank based semantic based and classifier based. Then, the re-ranking measure is performed based on the extracted features using the proposed lion based optimization algorithm.

• The major contribution of this paper is to design and develop a meta-lion search engine for re-ranking of the web pages based on the lion algorithm. The proposed meta-lion search engine collects the ranking scores of various search engines, like Google, Yahoo, and Bing and generates the re-ranked results as output.

• Here, the hybrid feature extraction process is proposed in which the text based, factor based, rank based and classifier based features are extracted to improve the performance of the meta-search engine. Then, the re-ranking measure is performed based on the extracted features using the proposed lion based optimization algorithm.

The paper is organized as follows. Section 2 presents the motivation and the problem statement based on the LionRank metasearch engine. Section 3 provides the detailed explanation of the proposed methodology. Section 4 illustrates the experimental results and Section 5 concludes the paper.

2 Motivation for research/objective

Challenges. Building a good metasearch engine can be difficult, as different query languages are needed to access various engines and, furthermore, the engines use undisclosed ranking algorithms. Moreover, most of the popular metasearch engines need to pay for bandwidth and negotiate with the primary engines for continued high volume access [4].

Re-ranking search results to obtain the most relevant documents at the top by adapting to the user's interests is useful and a well-known problem in the area of information retrieval [18].

Web meta-searching is a more complex problem than rank aggregation. Individual rankings might be noisy, incomplete or even disjoint. Hence they should not be the only parameter affecting the ranking. Further processing is required to filter the results and allow the final result of the metasearch engine to be free of unwanted, devious, and unfairly highly ranked web-pages.

Because commercial interests might frequently and unpredictably affect the results of searching, the user is not clearly protected against the interests of individual search engines. Therefore, the ranking algorithm employed by a real metasearch engine should be able to provide results that are as free as possible from paid listings and links.

Problem statement. Assume that $Wg_1, Wg_2, \ldots, Wg_n$ are the search results of search engine SE1 and $Wy_1, Wy_2, \ldots, Wy_n$ be the search results of SE2; similarly, let $Wb_1, Wb_2, \ldots, Wb_k$ be the web results of the search engine SE3. The objective here is to remove duplicate web-pages among these three results and find the rank of the unique webpages.

Challenges covered in developing metasearch engine. The development of a new search engine poses the following challenges.

• Selection of search engine. Metasearch engine requires multiple search engines and its interfaces to connect with a new metasearch engine. The result of the metasearch engine completely depends on the

results of the search engines that are considered. Thus, the right selection of the multiple search engines plays a major role in the development of a metasearch engine.

• Bringing semantic richness. Before integrating the search results, the chief problem to be addressed is how to obtain the semantic results from the user query or reformulated query even though the intent of keywords is not presented in the user query.

• Designing of merging strategy and algorithm. The important challenge to be considered in the metasearch engine is how to integrate all the results of the different search engines and the ranking of those results. This includes the process of selecting the search engine results of important webpages, removing the search engine web results of unnecessary webpages, and ranking webpages. These challenges pose a problem in designing an aggregate ranking algorithm for metasearch engine.

• Visualization of merged results. The final step is to visualize the ranked web pages to user-friendly interface to easily read and analyze the retrieved information.

Objective. The main objective of this exploration work is to develop powerful techniques for an effective metasearch engine, with the aim of covering a much bigger web space while simultaneously retrieving most of the state-of-the-art and more applicable records than existing web crawlers and metasearch engines.

3 LionRank: lion algorithm-based metasearch engines for re-ranking of webpages

The block diagram representation of the proposed method is shown in Figure 1. The steps involved in the proposed metasearch engine aggregation process is discussed as follows: First, the user sends the query to various underlying searching engines like Google, Yahoo and Bing. Thereafter, a number of webpages are generated by the selected underlying search engines based on the given user query. From the meticulous set of generated web pages, 40 web pages are selected from corresponding search engines for the analysis of the proposed LionRank method [17]. The filtered webpage is then obtained by removing the webpages that are generated by the same URLs. The pre-processing steps such as HTML page to word conversion, stop word removal and the stemming process are applied to reduce the seeking time of the user and thereby, various keywords are selected from the generated webpages at the end of pre-processing. From these selected keywords, the top five keywords are extracted for further processing. The features are then extracted from these keywords using text-based, factor-based, rank-based semantic-based and classifier-based techniques. Using these extracted features results, the re-ranking measure is performed by making use of Lion-based optimization algorithm. The weight values accountable for ranking of respective webpages are calculated using the Lion algorithm. Here, the documents that have been selected from the different search engines are re-ranked and developed based on the requirement of the user.

3.1 Pre-processing

The webpage search results of a user query are a descriptive collection of information from various fields. Thus, the process of retrieving the relevant information based on the user query is one of the challenging tasks for various search engines. Before performing the process of information retrieval, the pre-processing steps are applied to webpages to filter the webpage information and thus, reduce the seeking time of the user. First, the input query is applied to three participating three search engines such as Google, Yahoo and Bing. Let us assume that the input database, which contains a number of webpages stored in HTML format, is defined as follows:

$$D_s = \{W_{is},\ 0 \leqslant i \leqslant N,\ 0 \leqslant s \leqslant 3\}, \tag{1}$$

where the input database is denoted as D_s and the number of stored webpages is represented as N. Here, the number of selected search engines is defined as s. After selecting the web-page information from the corresponding database, the pre-processing steps are applied to get the filtered output. The filtered outputs are unique pages that are selected based on the URL analysis concept. Once a filtered page is generated, the HTML page is converted into words. In order to perform the HTML conversion process,

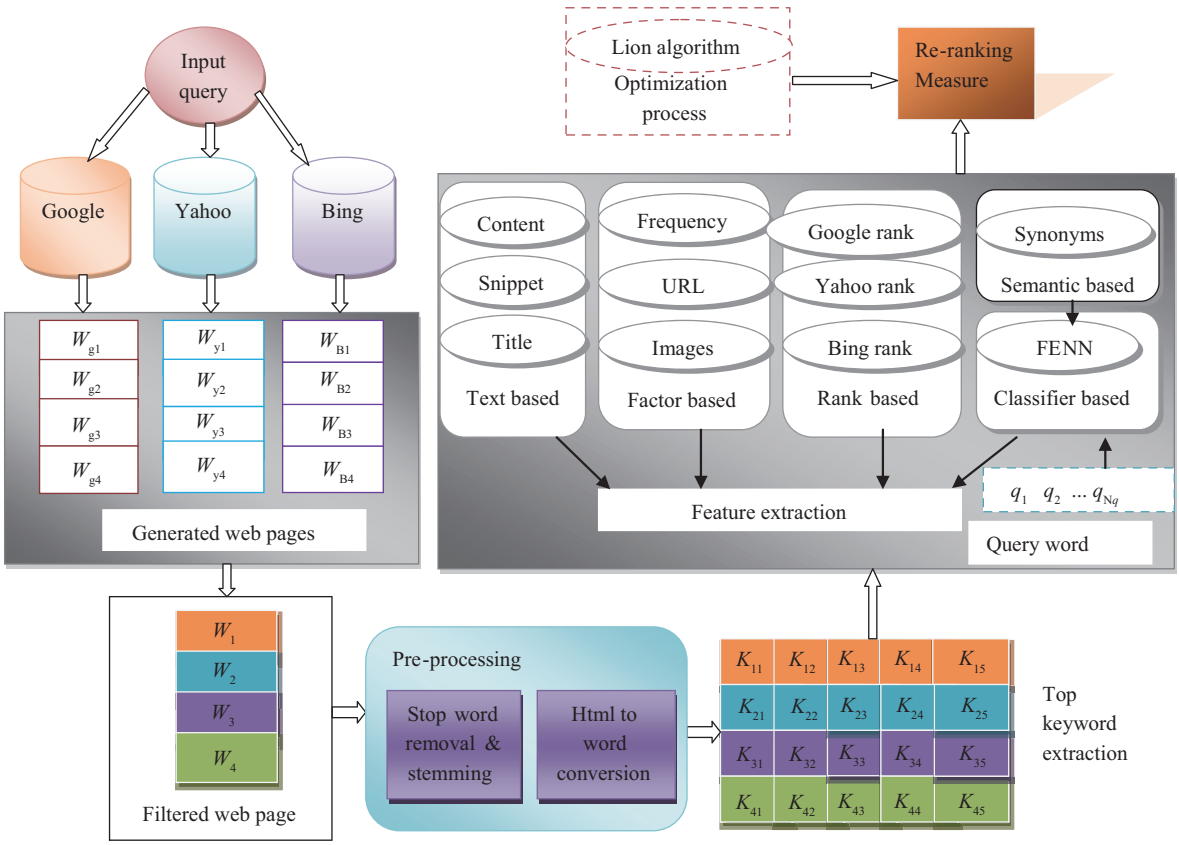

Figure 1 (Color online) Proposed system for re-ranking of webpages.

the HTML webpage is read and the tags are removed from the webpages. The stop word removal process is then applied to remove the stop words such as 'a', 'an', 'the', 'as', 'on', and 'like' from the webpages. After the stop word removal process, the stemming algorithm is applied to the webpages to convert the keywords to their root format. As a result of the pre-processing steps, the information stored in the webpage is converted into keywords. Finally, the keywords are extracted from the webpage and are then considered for the feature extraction process. After performing the pre-processing step, the generated webpage information is represented as follows:

$$W_{is} = \{d_{ij},\ 0 \leqslant i \leqslant N,\ 0 \leqslant j \leqslant M\}, \tag{2}$$

where d_{ij} is the information of j-th keyword from the i-th webpages, N is the number of webpages in the input database and M is defined as the generated keywords. All the three sets of webpages selected from the three databases are then analyzed to remove duplicate webpages. The database representation of the search engines Google, Yahoo and Bing are expressed as D_1, D_2, and D_3, respectively. The combined information is then stored in a temporary database P:

$$P = \{D_1, D_2, D_3\}. \tag{3}$$

Time complexity reduction and reliability upgrade are possible by selecting the top words from the unique webpages. Accordingly, the unique webpages are selected from the three search engines and the final webpage is generated based on the unique information. Here, the information is stored in the form of keywords.

$$P_d = \{W^l,\ 0 \leqslant l \leqslant U\}, \tag{4}$$

where W^l is defined as the unique webpages, in which the range of l can be varied based on the number of unique webpages U. Thus, P_d is defined as the database created based on the number of unique

webpages. Here, the top keywords that are considered for the evaluation process can be expressed as follows:

$$W^l = \left\{K_y^l,\ 0 \leqslant y \leqslant W^t\right\}, \tag{5}$$

where W^t is denoted as the number of selected top keywords and K_y^l is defined as the y-th keyword of the l-th webpage.

3.2 Extraction of features

This subsection presents the feature extraction process for re-ranking the search results in the developed metasearch engine. The feature is extracted from the keywords of the unique key pages. Features of top keywords are extracted based on the following traditions such as text-based, factor-based, semantic-based, classifier-based and rank-based types of feature extraction.

3.2.1 *Text-based feature extraction*

Basically, the text document retrieval process is performed based on the collection of words in each document. To extract the text-based features, we mainly consider the content, snippet and title of the webpages.

(i) Content. When the textual query is applied to the input of the search engine, the initial search is carried out using the content-based feature extraction. Here, the content-based information is named as query log, which contains rich information to analyze the requirement of the user. Initially, the first unique webpage is considered for the feature extraction. Here, the selected keywords are extracted based on the content-based features. The sorting process is then performed in the results of content-based extracted keywords in descending order. The ranking score for the extracted results is then calculated based on the resulting sorting order in order to find the location of the information. After assigning the rank, the content-based correlation (CBC) is calculated between the ranking scores of all keywords in the particular unique webpage. This process is continued for the selected number of unique webpages, which can be shown as follows:

$$\mathrm{CBC} = \frac{1}{U}\left\{\sum_{i=1}^{U} CP_i^d\right\}, \tag{6}$$

$$CP_i^d = \frac{N\sum(w_x w_y) - \left(\sum w_x\right)\left(\sum w_y\right)}{\sqrt{\left[N\sum w_x^2 - \left(\sum w_x\right)^2\right]\left[N\sum w_x^2 - \left(\sum w_y\right)^2\right]}}, \tag{7}$$

$$R_1 = f(\mathrm{CBC}), \tag{8}$$

where U is defined as the number of unique webpages, $f(\mathrm{CBC})$ is the sort function that sorts the 'CBC' values of every webpage in descending order. This means that every webpage obtains the score values of 'CBC' based on the content which is then sorted in descending order. Additionally, $f(\mathrm{CBC})$ denotes the sorted content-based correlation function in descending order, which is stored in the vector format of R_1. The value N denotes the number of pairs of ranking scores selected from the retrieved webpages and the sum of the product of paired scores of the webpages is denoted as $\sum(w_x w_y)$.

(ii) Snippet. This subsection presents the textual snippet feature extraction from the web document to perform the rank aggregation. A small fragment of the text document is taken as the feature for the snippet-based feature extraction, which is used to find the relevance based on the applied query. Based on the snippet-based feature extraction, a segment of the raw code is inserted in the selected webpage. Basically, the snippets often contain HTML code to add a sorting table and text block. First, the snippet-based feature extraction is performed for the top keywords that were present on the first webpage. Thereafter, we find a correlation between the all extracted results.

$$\mathrm{SBC} = \frac{1}{U}\left\{\sum_{i=1}^{U} SP_i^d\right\}. \tag{9}$$

Once the result is extracted, it will be sorted in descending order. Based on this sorted order, the rank is generated to find the weight of the applied query.

$$R_2 = f(\text{SBC}), \tag{10}$$

where R_2 is the generated rank score based on the snippet - based feature extraction and SBC denotes the snippet-based correlation process.

(iii) Title. Basically, the title-based correlations (TBC) are considered for the text-based feature extraction process. Instead of analyzing the whole webpage result, the title of the page is selected for the feature extraction to save space and seeking time. The title-based correlation can then be established into the individual unique webpages and the correlation results are sorted in descending order to generate the ranking score.

$$\text{TBC} = \frac{1}{U}\left\{\sum_{i=1}^{U} TP_i^d\right\}, \tag{11}$$

$$R_3 = f(\text{TBC}), \tag{12}$$

where TBC is defined as title-based correlation. The function R_3 is defined as the generation of ranking score which is sorted in descending order of text-based correlation results.

3.2.2 *Factorbased feature extraction*

This subsection presents the factor-based feature extraction process, in which the attributes are selected based on the frequency, URL and images.

(i) Frequency. To extract the frequency-based feature extraction, we must calculate the ratio of both query frequency and total number keywords in the webpage. The frequency of the input query is calculated by counting the number of occurrence of the query in the collected webpages. Here, the frequency-based extraction is based on the frequency of the query keyword within the webpage and the similarity of the documents containing the query keyword.

$$F = \frac{f_Q}{T_{\text{word}}}, \tag{13}$$

$$R_4 = f(F), \tag{14}$$

where f_Q is denoted as query frequency the total number of words present in the webpages are represented as T_{word}, $f(F)$ is defined as the frequency-based sorting function in descending order and the generated rank is denoted as R_4.

(ii) URL. Based on the input query, the metasearch engine creates multiple query expansions. When the user gives the query, the high possibility of visitation can be extracted using an HTML-based feature extraction process. Basically, the URL based information is used to store the viewers' counting details and timestamp. Here, the selected URL based information is based on the ratio of the number of internal links to the number of external links,

$$\text{URL} = \frac{\text{number of IL}}{\text{number of EL}}, \tag{15}$$

$$R_5 = f(\text{URL}), \tag{16}$$

where the uniform resource locator is denoted as URL, IL denotes the internal links, and EL represents the external links. While performing the webpage retrieval process, a study of webpage relationships is carried out based on the link analysis. Basically, the link-based feature extraction of the search engines is used to generate a higher quality of the results based on the query input. The internal link connects one webpage to different webpages on the same website. The selected webpages can be linked to other websites from external links such as twitter and face book.

(iii) Images. Here, the image-based feature extraction is used to generate the information relevant to the user query. The selection of images based on the user query becomes more complex when there is a large-scale collection of information.

$$I = \frac{\text{number of image}}{\text{maximum number of image}}, \tag{17}$$

$$R_6 = f(I). \tag{18}$$

In addition, the pair-wise similarity (PS) drab also occurs during the image-based feature extraction. The image-based feature extraction process is defined as the ratio of the number of given query image to the maximum number of images presented on the webpage.

3.2.3 *Rank-based features*

When the query is applied to the various underlying search engines the webpages are generated based on the user's information. The ranking score of the search engines is then calculated based on the webpage information. Thereafter, the calculated ranking score of underlying search engines Google, Yahoo and Bing can be represented as G, Y, and B, respectively. Thereafter, the ranking score of each selected search engines are combined and used to generate the final ranking score based on the ascending order.

$$r = \frac{1}{3}(G + Y + B), \tag{19}$$

$$R_7 = g(r), \tag{20}$$

where $g(r)$ is the sort function which sorts the 'r' values in ascending order. This means that every webpage obtains the score values of 'r' based on the rank measure which is then sorted in ascending order. Also, $g(r)$ is defined as the rank based sort function in ascending order thus, the location based on the assigned rank value can be calculated.

3.2.4 *Semantic classifier based features*

To find the influence level of semantics using the rankings of search engines, the semantic-based statistics are extracted. Semantic mark-ups are mainly expressed by the number of possible synonyms corresponding to the user query. Thereafter, the generated synonyms are compared with the keywords which have been generated by the underlying search engines. The feasible synonym generated based on the user query information is represented as follows:

$$S_v = \{Q_a,\ 0 \leqslant a \leqslant h\}, \tag{21}$$

where the number of possible synonyms that can be obtained based on the user query is represented as h, and the a-th synonym of the input query is defined by Q_a.

Fuzzy integrated extended nearest neighbor (FENN) classifier is used to retrieve the relevant documents of the input query by matching it with the semantic-based extracted feature [19]. To find the relevant text of the user query, the neighbors of the input query and neighbors of the neighbors of the input query are utilized. The distance calculation between the semantic-based vector and the top keywords generated from the webpages of the search engines can be represented as follows:

$$S_b(S_v, W_{is}) = \frac{\alpha \cdot Y_b + \beta \cdot \mathrm{YY}_b}{\alpha + \beta}, \tag{22}$$

where α and β are the weighted constants, Y_b is the neighbor of the original input query and the neighbors of the neighbors of the original query sample are referred to as YY_b, which is related to the b-th class. The query result of both Y_b and YY_b are then weighted using the fuzzy score value. The score values of every document are obtained based on the availability of both Y_b and YY_b values. To obtain the final

score value of every webpage document the selected score values of both Y_b and YY_b values are weighted with α and β values.

$$Y_b = \frac{1}{n \times t} \sum_{r=1}^{t} D_r\left(Q_a, d_{ij}\right), \quad D_r\left(Q_a, d_{ij}\right) = \begin{cases} 1, \text{ if } Q_a \in d_{ij} \ \&\& \ n_r(S_v, W_{is}) \in d_{ij}; \\ 0, \text{ otherwise}, \end{cases} \tag{23}$$

$$\mathrm{YY}_b = \frac{1}{n_b t} \sum_{Q_a \in d_{ij}} \sum_{\mathrm{rr}=1}^{t} D_{\mathrm{rr}}\left(Q_a, d_{ij}\right), \quad D_{\mathrm{rr}}\left(Q_a, d_{ij}\right) = \begin{cases} 1, \text{ if } Q_a \notin d_{ij} \ \&\& \ n_r(S_v, W_{is}) \in d_{ij}; \\ 0, \text{ otherwise}, \end{cases} \tag{24}$$

where the number of data samples present in the input database is represented as n, n_b is the number of data samples belonging to the b-th class and the number of neighbors of the query sample is denoted as t. The values of both $D_r(q, v)$ and $D_{\mathrm{rr}}(q, v)$ are computed based on the neighbors of the original query and the neighbors of the neighbors of the original query Y_b and YY_b, respectively. Here, the values of both $D_r(q, v)$ and $D_{\mathrm{rr}}(q, v)$ are calculated by incrementing the possessions of query and neighbors of the neighbors. Once the score value is calculated from the webpages, the sorting has the maximum value taken as the value for the input query. Finally, the documents based on the sorting order are retrieved from the database and is given to the users. Thereafter, the ranking score is generated from the sorting results of semantic-based extracted features based on the FENN classifier which has been stored in ascending order and that result is denoted as R_8:

$$R_8 = g(S(\mathrm{Sv}, \mathrm{Dwi})), \tag{25}$$

where Sv is the semantic-based feature extracted results and Dwi denotes the database based on the user query.

Finding of neighbors. To calculate the neighbors of the input query, the document similarity measure [20] is used, in which the matching process is done between the input query and the feature library. Because of the consideration of both frequency and similarity of the nearest documents, the similarity measure is applied to find similar values. In this paper, we have used similarity measure to calculate the occurrence of the query keyword on the webpage and the resemblance of the documents having the query keyword. To confine the data range, these parameters are combined with the logarithmic function. The similarity measured based on the input query is represented as follows:

$$e_Q^{w_l} = \log\left(1 + \sum_{x=1}^{n_x} f_{q_x}^{e_l} \times \mathrm{idf}_{q_x} \times B_{e_l}\right), \tag{26}$$

where idf_{q_x} represents the inverse document frequency of the query keyword q_x, the similarity of the query is represented as Q, and w_l denotes a webpage. The number of keywords determined based on the input query is represented as n_x and $f_{q_x}^{e_l}$ is the occurrence of the query keyword q_x on the webpage e_l. The inverse document frequency is calculated as the ratio of the total number of documents in the database to the number of the document that have the query keyword q_x.

$$\mathrm{idf}_{q_x} = \log \frac{n}{1 + |W \in v : v_{ij} \in W|}, \tag{27}$$

where the number of documents present in the input database is defined as n and the parameter of B_{e_l} can be represented as follows:

$$B_{e_l} = \frac{\sum_{a=1}^{n_r} d(e_a, e_l)}{n_r - 1}, \tag{28}$$

where the Euclidean distance between the neighbor documents is denoted as $d(e_a, e_l)$, and the number of documents selected based on the query keyword is represented as n_r.

3.2.5 *Re-ranking measure*

This subsection presents the process of re-ranking, in which the ranking scores are collected from the various underlying search engines and the final re-ranking process is done based on the lion optimization

algorithm. In re-ranking process, the ranking scores of the underlying search engines are combined by the LionRank algorithm Re-ranking of the results is used to select the most relevant results to the user. The proposed re-ranking measure formula is shown as follows:

$$\mathrm{RRM} = \alpha_1 (R_1) + \alpha_2 (R_2) + \alpha_3 (R_3) + \beta_1 (R_4) + \beta_2 (R_5) + \beta_3 (R_6) + \gamma (R_7) + \Gamma (R_8), \tag{29}$$

where α, β and γ values are defined as the weighted constants and RRM represents the re-ranking measure.

3.3 Finding optimal weights for re-ranking

Lions have an interesting social behavior of maintaining their strength in every generation, unlike other cat species. Basically, the lion algorithm is used to find the optimum solution, which is based on the behaviors of two unique lions such as terrestrial defense and terrestrial takeover.

The steps involved in the proposed lion's optimization algorithm are as follows: (i) generating solutions (pride generation); (ii) deriving the new solutions (mating process); (iii) performing the evaluation between the existing and newly derived solution (terrestrial defense); (iv) replacing the worst solution with the developed best solution (terrestrial takeover).

(a) Solution representation. This subsection presents the calculation of the solution vector representation. Initially, the solution vector is generated randomly. The searching process of the lion algorithm mainly focuses on the calculation of the optimal solution. The pride generation initiates the procedure of a lion algorithm. Initially, the pride has the two vector solutions a male and a female. The structure of both male and female vector solutions can be represented as follows:

$$V^m = [v_1^m, v_2^m, v_3^m, \ldots, v_L^m], \tag{30}$$

$$V^f = \left[v_1^f, v_2^f, v_3^f, \ldots, v_L^f\right], \tag{31}$$

where V_m and V_f is defined as the solution vectors corresponding to the both male and female lions, respectively. The size of the generated solution vectors is based on the number of optimization weights which we have considered for the re-ranking process. In this paper, we have used eight optimization weights, which have been represented as α_1, α_2, α_3, β_1, β_2, β_3, γ, and Γ. Accordingly, the length of the solution vector is represented as eight ($L = 8$). The function of the pride value can then be calculated using the objective function of both the male and the female vector with their cubs. Here, the cubs are defined as the derived solutions which are generated based on the existing male and female solutions. The calculation of pride solution can then be represented as follows:

$$V^P = \frac{1}{2\left(1 + \|V^{m\_\mathrm{cu}}\|\right)} \left(f\left(V^m\right) + f\left(V^f\right) + \frac{A_m}{A_{\mathrm{cu}} + 1} \sum_{C=1}^{\|V^{m\_\mathrm{cu}}\|} \frac{f\left(V_c^{m\_\mathrm{cu}}\right) + f\left(V_c^{f\_\mathrm{cu}}\right)}{\|V^{m\_\mathrm{cu}}\|} \right), \tag{32}$$

where the strength of the male and the female cubs are denoted as $V_c^{m\_\mathrm{cu}}$ and $V_c^{f\_\mathrm{cu}}$, respectively. Thereafter, the ripeness age for mating is represented as A_m and the cub's age can be denoted as A_{cu}. Initially, the fertility evaluation is carried out for selecting the suitable solution vectors for evaluation. During matting process, two primary steps are considered such as cross over C_r and mutation M_r. Here, the value of both crossover and mutation probability is randomly selected by the user. To select the appropriate solution vector, the comparison analysis is performed between the male and nomad solution vector along with the pride and the nomad solution vector. Initially, the nomadic vector production is similar to the generation of the male vector, which can be expressed as follows:

$$V^n = [v_1^n, v_2^n, v_3^n, \ldots, v_L^n]. \tag{33}$$

(b) Fitness evaluation. In the lion's social system (pride), the comparison is done based on strength. In this paper, the comparisons between the all three vectors such as male, female, and nomadic vector are made based on the fitness value. Accordingly, the vector corresponding to the best fitness value can

be chosen as the solution vector. Here, the fitness value of the three solution vectors $f(V^m)$, $f(V^f)$ and $f(V^n)$ can be evaluated using

$$\text{Fitness evaluation} = \left(\frac{\text{Precision} + \text{Recall} + \text{F-Measure}}{3}\right). \tag{34}$$

The recall function in the fitness function is the ratio of relevant retrieved documents to the relevant document, whereas the precision function in the fitness function is the ratio of relevant retrieved document to the retrieved document. By using such functions in the fitness function, only the relevant webpages/documents are ranked for the user query in our proposed LionRank metasearch engine.

(c) Lion algorithm searching process. The steps involved in the searching process of the proposed lion algorithm [17] can be represented as follows: (i) Initially, randomly generates the three vectors named as V^m, V^f and V^n. Here, the sizes of these three vectors are based on the number of weighted constant used in this paper. (ii) The fitness value of these randomly generated vectors is calculated using the fitness evaluation formula. (iii) In order to avoid convergence to the local optima, the fertility evaluation process is carried out. Here, the representation of l^r, u^c, G^c and s^r is denoted as Laggardness rate, female update count, female generation count and sterility rate, respectively. At the initial stage of fertility evaluation, the laggardness rate and sterility rate are taken as zero. The sterility rate is used to calculate the acceptance value. Using the calculation of fitness value, the male and female vectors are generated. (iv) Thereafter, the mating process is held between the generated two solution vectors (V^m, V^f). The primary steps of the mating process are carried out by mutation and crossover. Finally, the cubs of the both solution vectors generated by the mating process are represented as $V_c^{m\_\text{cu}}$ and $V_c^{f\_\text{cu}}$. (v) Once the cub growth function is completed, the terrestrial defense and terrestrial takeover process are carried out. (vi) In terrestrial defense, the existing solution vector is compared with the nomadic vector. Initially, the nomadic vector values are assigned randomly having sizes similar to the existing vectors. If a new vector (nomadic vector) is better than the existing vector (male vector), the existing vector is replaced by a new one. Thus the mating process continues. (vii) In the process of terrestrial take over, the best solution vector is selected as V^{best}, which is capable of generating new solution to an assured level and eliminating all solutions in the pride.

4 Results and discussion

This section presents the experimental results of the proposed LionRank search engine and the performance evaluation of the proposed method is compared with various existing search engines such as QuadRank [21], Outrank [22], Google, Bing and Yahoo.

4.1 Experimental set up

The proposed LionRank metasearch engine is implemented in java programming language with JDK 1.7.0. For the experimental analysis, we have four benchmark queries from TREC 2002 web track data[1)]. Similarly, we use the WSJ and AP dataset[2)], to test the performance of the proposed LionRank search engine.

4.2 Queries

In this paper, we have used three bench mark queries that can be represented as follows:

Q1: intellectual property;

Q2: foods for cancer patients;

Q3: federal funding for mental illness.

For experimental validation, the webpages documents related to these queries are collected from three standard search engines like Google, Bing, and Yahoo. These webpage documents are then manually

1) TREC 2002 web track data. http://trec.nist.gov/data/t11.web.html.

2) http://www.cs.cmu.edu/ yiz/research/NoveltyData/.

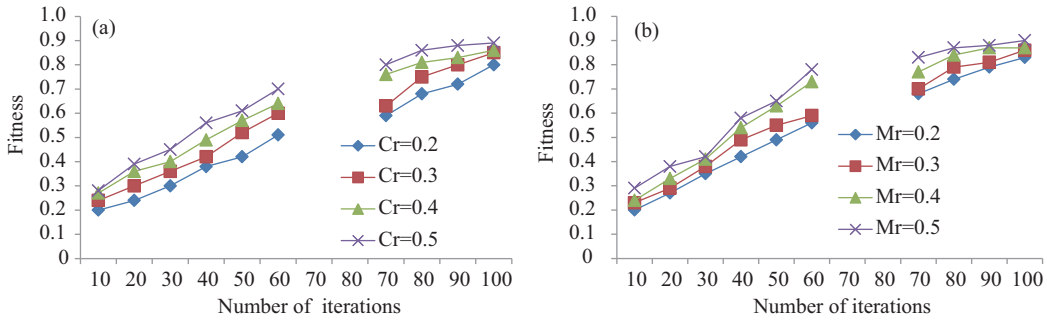

Figure 2 (Color online) Performance analysis of fitness evaluation. (a) Based on the crossover probability; (b) based on mutation probability.

analyzed to generate the ground truth of the webpages. Based on this ground truth the performance of the techniques is analyzed after supplying these queries to the webpage retrieval schemes. The retrieved documents are then analyzed using precision, recall, and F-score.

(a) Convergence analysis. This subsection presents the evaluation of the fitness value by varying the crossover and mutation probabilities. During the mating process, the crossover and the mutation probability measurements are used to perform the evolutionary optimization. By adjusting the values of both crossover and mutation value, the cub's generation process is performed in the lion algorithm. By changing the crossover probability, Figure 2(a) shows the evaluation of fitness value for the proposed Meta lion search algorithm.

When the value of the crossover probability is taken as 0.2, the fitness value is monitored for each iteration. In the 10th iteration, the fitness value of by the proposed LionRank is achieved as 0.2. For the 30th number of iteration, the fitness value is attained as 0.3. Further increasing the number of iterations for the same crossover probability, apparently increases the fitness value. Here, we considered only 100 number iterations. For the final iteration, the fitness value is measured as 0.8, in which the crossover probability is taken as 0.2. Thereafter, the value of crossover probability is changed from 0.2 to 0.3. By increasing the value of crossover probability, the fitness value also gets increased. When the value of crossover probability is taken as 0.3, the fitness value of the proposed method is evaluated as 0.24 for the initial stage of the iteration. In order to increase the fitness value, the number of iterations needs to be increased.

Figure 2(b) shows the fitness evaluation of the proposed LionRank based on the mutation probability. Here, the value of mutation probability is varied from 0.2 to 0.4. When the value of the mutation probability is taken as 0.2, the fitness value is measured as 0.2 for the first round of iteration. The number of iterations is then increased to 100, where the fitness value of the proposed method is measured as 0.83. If the measured fitness does not match with the desired value, re-evaluation of the fitness evolution is carried out based on the varying results of mutation probability. While considering the mutation probability as 0.3, the maximum fitness value obtained is approximately 0.86. To achieve the maximum fitness value, the mutation value is taken as 0.5, at which value the maximum fitness value is obtained as 0.9. From the Figure 2, we can say that the maximum fitness value can be evaluated by using a crossover and mutation probability.

(b) Ranking analysis. This subsection presents the ranking analysis of various search engines such as QuadRank, Outrank, Google, Bing and Yahoo with the proposed method.

The common rank is the matched rank of the webpages in the proposed LionRank search engine and the comparative search engines (QuadRank, Outrank, Google, Bing and Yahoo).

Figure 3(a) shows the ranking analysis of the various search engines based on the proposed LionRank using the first user query. Here, the comparison of the ranking analysis is performed based on the matched rank. When the number of web pages is retrieved as 5, the QuadRank metasearch engine achieves a 45% common rank compared to the proposed LionRank. By considering the ranking order of the proposed method, Outrank and Google search engines achieve common ranks of 42% and 40%,

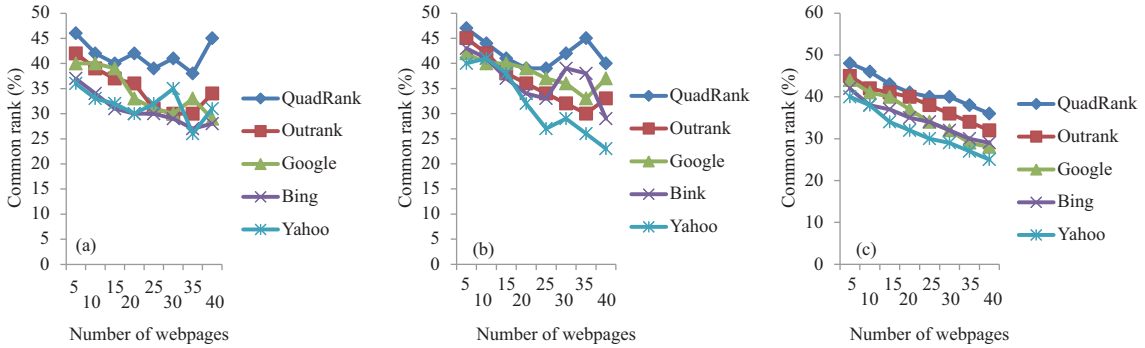

Figure 3 (Color online) Ranking analysis of various search engines based on the user query. Common rank analysis for (a) query 1, (b) query 2, and (c) query 3.

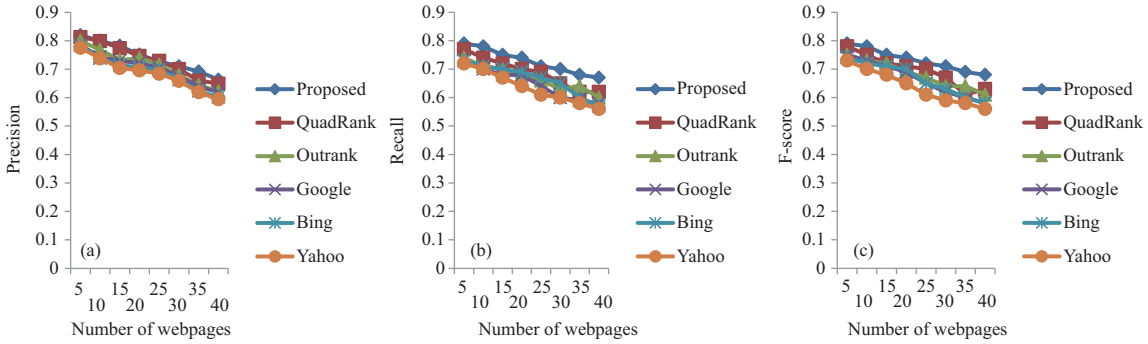

Figure 4 (Color online) Comparative analysis based on the first user query. (a) Precision; (b) recall; (c) F-score.

respectively. By analyzing the 15 retrieved webpages, a 40% common rank is obtained by the QuadRank metasearch engine and the Outrank metasearch engine acquires a 37% common rank. Figure 3(b) shows the common rank analysis of the second query word given by the user. While retrieving the 15 webpages, a 38% common rank is obtained by the Yahoo search engine and a 39% common rank is attained by the Outrank metasearch engine. Thereafter, this common rank analysis is continued for every number of retrieved webpages. Based on this common rank analysis we can analyze the performance of various search engines in tandem with the proposed LionRank.

When the user query is given, the search engines are used to generate the number of webpages based on the user query. After generating the webpages, the performance of search engines is analyzed based on the ranking analysis. Here, the ranking order of the various search engines is analyzed based on the proposed LionRank. Based on the ranking value, we can understand the performance of the various search engines. Figure 3(c) shows the ranking analysis of various search engines based on the third user query. When the third query is applied to the search engines, a 48% common rank is obtained by the QuadRank metasearch engine. However, the common rank of 42% is attained by the Bing search engine and the Google search engine obtains 44%.

4.3 Comparative analysis

(i) Comparative analysis based on Q1. This subsection presents the comparative analysis of the proposed metasearch engine with the other existing search engines such as QuadRank, Outrank, Google, Bing and Yahoo. The performance evolution of various search engines for the first user query is shown in Figure 4.

Figure 4(a) shows the precision measurement of the various search engines results with the proposed LionRank. In essence, the precision value is affected by increasing the number of retrieved pages. Here, 40 webpages are considered for the analysis. When the number of retrieved webpages is 35, the precision value of the proposed method is achieved as 71%. However, the precision values of other existing search

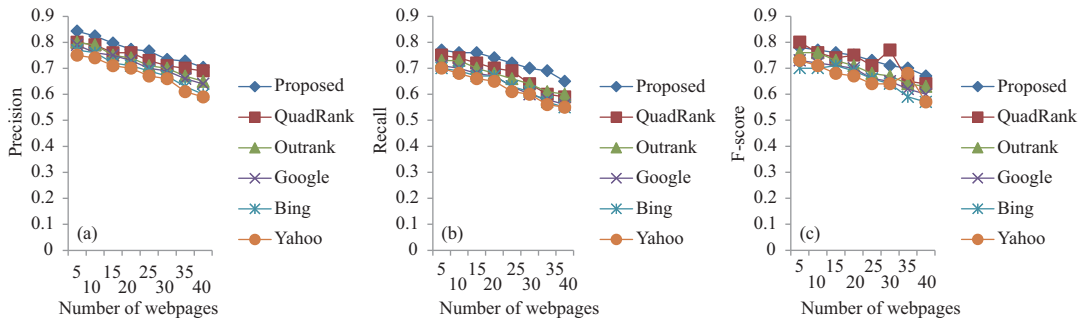

Figure 5 (Color online) Comparative analysis based on the second user query. (a) Precision; (b) recall; (c) F-score.

engines such as QuadRank, Outrank, Google, Bing and Yahoo are approximately 66%, 64%, 61%, 59% and 57%, respectively.

Figure 4(b) shows the recall measurement of the proposed metasearch engine. Here, the measurement is analyzed based on the ratio of intersection results between both relevant and retrieved documents to the total number of relevant documents. For 15 retrieved web pages, the recall measurement of the both Outrank and Bing search engines are achieved as 70% and the measurement result of Google search engine is achieved as 68%. However, the proposed LionRank search engine achieved a 75% recall value which is better than the recall measurement of the other search engines.

Figure 4(c) shows the F-score measurement of the various search engines based on the results of both precision and recall measurement. When the numbers of retrieved webpages are taken as 20, the F-score value of 69% is obtained by both Outrank and Google search engines. At the same time, the maximum F-score value of 74% is acquired by the proposed metasearch engine. The QuadRank obtained a 71% F-score value, when the number of retrieved webpages was 20.

(ii) Comparative analysis based on Q2. Figure 5 shows the comparative analysis of the proposed metasearch engine with the various existing search engines with the second user query based on the evaluation metrics such as precision and recall. Figure 5(a) shows the comparison of the proposed LionRank with the various existing search engines based on the precision measurement.

From the Figure 5(a), we can understand that the proposed LionRank acquired the maximum precision value compared to the other search engines. While retrieving the five pages, the precision value of 84% is obtained by the proposed method. However, an 80% precision is obtained by both the QuadRank and Outrank metasearch engines. The Yahoo search engine acquired the least precision value of 75%. By increasing the number of webpages related to a user query, the performance of all search engines gets affected.

Figure 5(b) shows the performance analysis of proposed LionRank based on the recall measurement for the second query given by the user. Here, the recall measurement of 67% is obtained by the both Google and Bing search engines, when the number of retrieved pages is taken as 25. For the same number of retrieved webpages, the proposed and QuadRank metasearch engines achieved the recall measurement value of 72% and 69%, respectively. In addition, the recall measurement value is increased by increasing the number of retrieved webpages.

The F-score measurement of the various search engines is analyzed and compared with the proposed LionRank which is shown in Figure 5(c). When the numbers of retrieved webpages are taken as 35, the maximum F-score measurement of 70% is obtained by the proposed metasearch engine. At the same time, the minimum F-score value of 59% is acquired by Bing search engine.

(iii) Comparative analysis based on Q3. In this subsection, Figure 6 shows the comparative analysis of various search engines with the proposed method using the third user query, which is shown in Figure 6(a). By taking the number of retrieved pages as 15, the precision measurement of various search engines is analyzed with the proposed metasearch engine. For the 15 number of retrieved webpages, the precision value of 78% is obtained for the proposed LionRank. For the same number of retrieved pages, the precision value of the Quad-rank and Outrank metasearch engines are obtained as 75% and

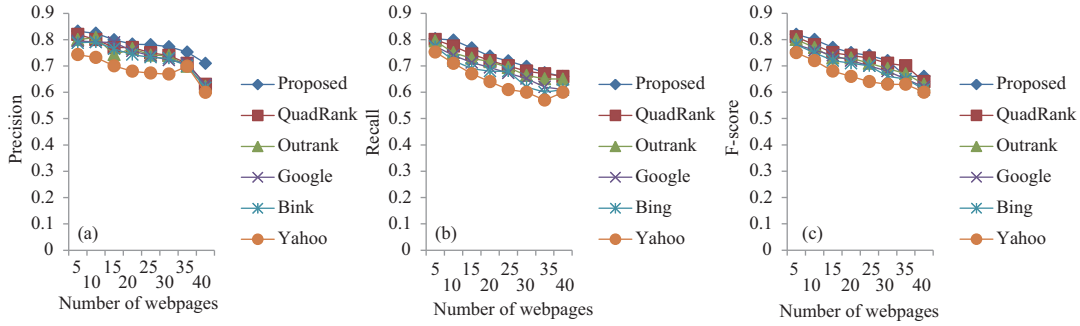

Figure 6 (Color online) Comparative analysis based on the third user query. (a) Precision; (b) recall; (c) F-score.

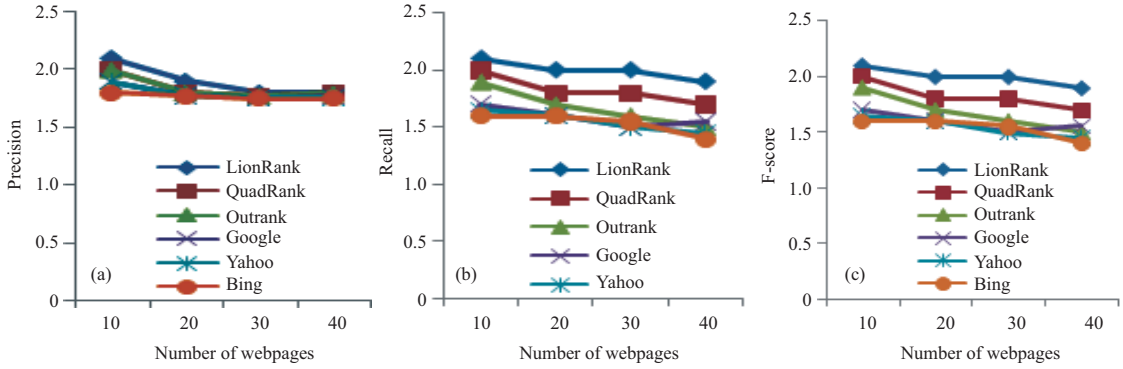

Figure 7 (Color online) Comparative analysis based Precision value. (a) Precision; (b) recall; (c) F-score.

74%, respectively. Here, the proposed metasearch engine maintains the highest precision value while considering lesser number of retrieved webpages.

Figure 6(b) shows the recall measurement of the various search engines with the proposed LionRank. By taking the number of webpages to be 5, the maximum recall value of 80% is acquired by the proposed LionRank. For the same number of retrieved webpages, the recall values of both QuadRank and Outrank metasearch engines are measured as 79% and 76%, respectively. To analyze the recall measurement of the proposed LionRank, the number of retrieved webpages was increased for further processing.

Figure 6(c) shows the performance evaluation of the proposed metasearch engine using the F-score value based on the third user query. Here, the F-score measurement of the various search engines is analyzed based on the number of retrieved webpages. When the number of retrieved webpages is taken as the top 5, the maximum F-score value of 82% is obtained by the proposed met-search engine and 81% of recall value is obtained by the QuadRank metasearch engine. From the analysis, we can understand that the F-score value decreases by increasing the number of retrieved webpages.

4.4 Experimentation results using WSJ and AP dataset

This subsection presents the experimental results of the proposed LionRank metasearch engine using the WSJ and AP dataset[3)].

Figure 7(a) shows the performance evaluation of the proposed metasearch engine using precision value. The analysis is performed by varying the number of retrieved webpages. When the number of retrieved pages is 10, the individual search engines QuadRank, Outrank, Google, Yahoo, and Bing obtained the precision values of 2, 2, 1.9, 1.9, and 1.8, respectively, whereas LionRank obtained the precision value of 2.1. From the analysis, it is clear that precision value decreases with increasing number of retrieved pages, and the proposed LionRank performed well in all cases.

The recall measurement of the various search engines, QuadRank, Outrank, Google, Bing, and Yahoo, are analyzed by comparing them with the recall of the proposed LionRank. The results are presented in

3) http://www.cs.cmu.edu/~yiz/research/NoveltyData/.

Table 1 MAP measure

Method	Mean average precision
LionRank	0.47
QuadRank	0.46
Outrank	0.46
Google	0.45
Bing	0.45
Yahoo	0.44
AdaRank.MAP	0.43
AdaRank.NDCG	0.43
Rank Boost	0.415
Ranking SVM	0.42
BM 25	0.41

Figure 7(b). For all the number of webpages considered for the experimentation, the proposed LionRank performed better compared to the other search engines.

Figure 7(c) depicts the performance evaluation based on F-score. F-score measurement is analyzed by varying the number of retrieved pages. Initially, when the number of retrieved pages is 10, the proposed LionRank obtained the F-score of 2.1, whereas, the search engines QuadRank, Outrank, Google, Bing and Yahoo, obtained the F-score of 2, 1.9, 1.7, 1.65, and 1.6, respectively. With 20 retrieved pages, the proposed LionRank attained the F-score value of 2, which is minimal compared to the F-score value attained when retrieving 10 webpages. However, the performance of LionRank is better than the performance of the existing search engines compared.

Comparative analysis based on MAP. The comparative analysis based on the mean average precision (MAP) value is listed in Table 1. In addition to the comparison made with the search engines, QuadRank, Outrank, Google, Bing and Yahoo, the MAP value of the proposed method is compared with the MAP value of Ada Rank [23].

From Table 1, it is clear that the proposed LionRank obtained the maximal MAP value of 0.47. Moreover, the significance student t-test is performed to analyze the effectiveness of the proposed LionRank metasearch engine. The student t-test is performed using the MAP value attained by the comparative methods, i.e., QuadRank, Outrank, Google, Yahoo, Bing, and AdaRank. The t-value obtained for validating the proposed LionRank and QuadRank is 2.27, LionRank and Outrank is 2.47, LionRank and Google is 3.28, LionRank and Yahoo is 2.61, LionRank and Bing is 2.25, LionRank and AdaRank is 2.34. This shows the significance of the proposed LionRank metasearch engine, demonstrating its statistical significance in webpage retrieval.

4.5 Practical implications

The proposed LionRank algorithm has various practical implications. (i) The findings of this work would be helpful for users to choose effective search engines; (ii) The results offers motivation to vendors of search engines to ensure that their technology is better; (iii) The findings would also be useful for many users of metasearch engines such that they do not use two highly correlated ones as their main search tools at the same time; (iv) Web users should be aware that limiting searches to single search engines results in missing considerable pieces of information ranked highly by other search engines and directories; (v) These findings assist users to select and make use of a metasearch engine such that the developers can design more efficient and effective search engines, and information professionals can identify and retrieve highly relevant documents that meet their information needs.

5 Conclusion

In this paper, we proposed LionRank, which is used to improve the performance of results obtained through various metasearch engines by retrieving webpage documents using the hybrid feature extraction

process such as text-based, factor-based, rank-based and classifier-based feature extraction. In classifier-based feature extraction we used FENN classifier to compute the score value of every document for the user query. Based on the results of extracted features from the webpage documents, the proposed LionRank was used to generate the re-ranked results. Here, the proposed lion algorithm was used to improve the re-ranking process of the proposed metasearch engines. For the experimentation, the webpage documents collected four relevant benchmark queries using various search engines. The performance analysis of the various search engines was then carried out in tandem with the proposed LionRank and the evaluation was performed using precision, recall and F-Score. The maximum F-score of 80% was obtained for the proposed LionRank which was higher than the value obtained by the existing search engines QuadRank, Outrank, Google, Yahoo and Bing. In the future, the performance enhancement of LionRank will be exploited with a hybrid optimization algorithm.

References

1 Naim I, Ali R. Metasearching using modified rough set based rank aggregation. In: Proceedings of International Conference on Multimedia, Signal Processing and Communication Technologies, Aligarh, 2011. 208–211
2 Keyhanipour A H, Moshiri B, Piroozmand M, et al. WebFusion: fundamentals and principals of a novel meta search engine. In: Proceedings of International Joint Conference on Neural Networks, Vancouver, 2006. 4126–4131
3 Sumiya K, Kitayama D, Chandrasiri N P. Inferred information retrieval with user operations on digital maps. IEEE Int Comput, 2014, 18: 70–73
4 Davison B D. The potential of the meta-search engine. In: Proceedings of the Annual Meeting of the American Society for Information Science and Technology, Providence, 2004. 393–402
5 Sun Y Z, Han J W. Meta-path-based search and mining in heterogeneous information networks. Tinshhua Sci Technol, 2013, 18: 329–338
6 Cao Y L, Huang T J, Tian Y H. A ranking SVM based fusion model for cross-media meta-search engine. J Zhejiang Univ Sci C, 2010, 11: 903–910
7 Keyhanipour A H, Moshiri B, Kazemian M, et al. Aggregation of web search engines based on users' preferences in WebFusion. Knowl-Based Syst, 2007, 20: 321–328
8 Desarkar M S, Sarkar S, Mitra P. Preference relations based unsupervised rank aggregation for metasearch. Expert Syst Appl, 2016, 49: 86–98
9 Ma S X, Li S Y, Yang H J. Creative computing for personalisedmeta-search engine based on semantic web. In: Proceedings of the 21st International Conference on Automation and Computing, Glasgow, 2015
10 Keyhanipour A H, Moshiri B, Lucas C. User modeling for the result re-ranking in the meta-search engines via the reinforcement learning. In: Proceedings of the 7th International Conference on Intelligent Systems Design and Applications, Rio de Janeiro, 2007
11 Lange S, Gebert S, Zinner T, et al. Heuristic approaches to the controller placement problem in large scale SDN networks. IEEE Trans Netw Serv Manage, 2015, 12: 4–17
12 Keyhanipour A H, Moshiri B, Lucas C. User modelling for the result re-ranking in the meta-search engines via the reinforcement learning. In: Proceedings of the 7th International Conference on Intelligent Systems Design and Applications, Rio de Janeiro, 2007
13 Mavridis T, Symeonidis A L. Identifying valid search engine ranking factors in a Web 2.0 and Web 3.0 context for building efficient SEO mechanisms. Eng Appl Artif Intel, 2015, 41: 75–91
14 Huang J, Yang X K, Fang X Z, et al. Integrating visual saliency and consistency for re-ranking image search results. IEEE Trans Multimedia, 2011, 13: 653–661
15 Hassanpour H, Zahmatkesh F. An adaptive meta-search engine considering the user's field of interest. J King Saud Univ Comput Inf Sci, 2012, 24: 71–81
16 Vargas-Vera M, Castellanos Y, Lytras M D. CONQUIRO: a cluster-based meta-search engine. Comput Human Behav, 2011, 27: 1303–1309
17 Rajakumar B R. Lion algorithm for standard and large scale bilinear system identification: a global optimization based on lion's social behaviour. In: Proceedings of Congress on Evolutionary Computation, Beijing, 2014
18 Rohini U, Varma V. A novel approach for re-ranking of search results using collaborative filtering. In: Proceedings of the International Conference on Computing: Theory and Applications, Kolkata, 2007. 491–496
19 Iraji M S, Maghamnia H, Iraji M. Web pages retrieval with adaptive neuro fuzzy system based on content and structure. Modern Educ Comput Sci, 2015, 7: 69–84
20 Karisani P, Rahgozar M, Oroumchian F. A query term re-weighting approach using document similarity. Inf Process Manage, 2016, 52: 478–489
21 Akritidis L, Katsaros D, Bozanis P. Effective rank aggregation for metasearching. J Syst Softw, 2011, 84: 130–143
22 Muller E, Assent I, Steinhausen U, et al. OutRank: ranking outliers in high dimensional data. In: Proceedings of IEEE International Conference on Data Engineering Workshops, Cancun, 2008. 600–603
23 Xu J, Li H. Adarank: a boosting algorithm for information retrieval. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and development in Information Retrieval, Amsterdam, 2007. 391–398