International Journal of Data Science and Analytics (2021) 11:37–55
https://doi.org/10.1007/s41060-020-00232-2

REGULAR PAPER

A novel approach for ranking web documents based on query-optimized personalized pagerank

Rajendra Kumar Roul[1] · **Jajati Keshari Sahoo**[2]

Received: 20 March 2018 / Accepted: 1 August 2020 / Published online: 18 August 2020

Abstract
Ranking plays an important role in the search process of web documents on a huge corpus. This not only reduces the searching time but also provides useful documents to the users. In this paper, we extend our earlier query-optimized PageRank approach by combining the TF-IDF and personalized PageRank algorithm to generate a robust ranking mechanism. In our earlier approach, we modeled a ranking scheme by considering the link structures of the documents along with their content. A novel feature selection technique named as 'Term-term correlation-based feature selection' (*TCFS*) is also proposed which removes all noise terms from the document before the ranking process starts. We believe that by incorporating *TCFS* and personalized PageRank of the documents along with their relevance will improve the retrieval results. The aim is to modify the link structure based on the similarity score between the content of the document and the user query. Experimental results show that the proposed feature selection technique can outperform the conventional feature selection techniques, and the performance of the combined TF-IDF and personalized PageRank approach is promising compared to the traditional approaches.

Keywords Correlation · Cosine-similarity · Peronalized PageRank · Silhouette coefficient · TF-IDF

1 Introduction

With the first growing of the internet, everybody experiences a flood of information. It is estimated that the present web contains at least 4.62 billion page[1] which makes it highly difficult for common users to get the desire information on the web. *Ranking* is the most commonly used technique in the field of Information Retrieval (IR) which brings the required documents on top of the retrieved results. Initially, for a given set of documents and a query, a scoring function is computed which finds the degree of relevance of each document with respect to the query. Then a ranking list is generated by sorting the documents based on their relevance score. Modern ranking approach uses different machine learning techniques such as BM25 [37], PageRank [34] etc., to generate such a ranking function and therefore achieve great improvements on the ranking performances [28]. By nature most of the IR problems are ranking problems, such as anti web spam [38] [45], collaborative filtering [7], product rating [14], key term extraction [11,41,44,46], important e-mail routing [10], sentiment analysis [32,43], definition finding [58], text summarization [40,48] etc. Among these ranking problems, document ranking is a common problem that is faced by many search engines. Emails, web documents, news articles, books, and academic papers are some examples of documents. Some ranking scenarios of document retrieval are:

- Documents are ranked as per their relevance to the user query.
- Website structures [12], diversity [61], and relationships of similarity [55] between documents are some of the features that are considered during the ranking process and is known as relational ranking [36].
- Several candidate ranked lists are aggregated to get a better ranked list, known as meta search [4].

[1] http://www.worldwidewebsize.com/

✉ Rajendra Kumar Roul
raj.roul@thapar.edu

Jajati Keshari Sahoo
jksahoo@goa.bits-pilani.ac.in

[1] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, Punjab 147004, India

[2] Department of Mathematics, BITS,Pilani-K.K.Birla Goa Campus, Goa 403726, India

– Finding up to what degree the property of a web document influences the ranking result.

In recent years, ranking has become a very important research direction in the domain of IR, and a large number of ranking models has been proposed that achieved high influence [3, 21,39,57]. All these models can be roughly categorized as

– *Query-dependent model*:
  In this model, documents are retrieved based on the occurrences of the query terms in the documents. Examples are standard Boolean model [9], Vector Space model [50], Latent Semantic Indexing [31], Probabilistic ranking technique such as Binary independence model [23], Latent dirichlet allocation [24].
– *Query-independent model*:
  In this model, documents are ranked based on their own importance such as traditional PageRank algorithm, query-independent learning [13], content-based technique [30] etc.

PageRank is the first algorithm which is used in Google search engine to rank the web pages (or documents). In the PageRank algorithm, the importance of a web document is evaluated by considering the number of quality incoming links to that web document. As the internet is growing rapidly, the PageRank will help to retrieve the required information in the fastest way. In the current PageRank algorithm, the importance or relevance of a web document is a relative concept, and it completely depends on the user query. This is one of the major drawbacks of the present ranking system and will be solved by using the concept of *Personalized* PageRank algorithm. There are many such limitations of the existing PageRank algorithms [22] and some of them are listed below:

– In some of the PageRank algorithms, PageRank is calculated not at the query time but at the indexing time.
– Most of the PageRank algorithms have a problem called *topic drifting* which decreases their efficiency.
– Some PageRank algorithms are judged based on the importance of the web documents, whereas some of them completely ignore the importance of each individual document.
– Content of web documents which play a vital role in PageRank algorithm is ignored sometimes which reduces the performance of the algorithm.

Among the above limitations, the main limitation of the traditional PageRank algorithm is *topic drifting*. This is because it considers uniform link structure i.e., a surfer will jump from one document to the other uniformly. For example, suppose someone is looking for documents related to *computer science*, then those documents have outgoing links to *biological documents* are also incorporated in the computation of PageRank (since some biological documents can be relevant or linked to computer science such as 'prediction of diabetics using machine learning', 'detection of breast cancer', 'recognizing brain tumor', 'finding stress level in the human brain' etc.).

Our earlier query-optimized PageRank approach [42] succeeded in dealing with the limitations of the PageRank algorithm by biasing the next jump to the relevant documents of the user query. The importance of web pages for different users can be better determined if the PageRank algorithm takes into consideration user preferences which is called as *personalized page ranking*. The importance of a page differs for different individuals with different interests, knowledge and backgrounds. So, a global ranking of a web page might not necessarily indicate the importance of that page for individual users. It is important to calculate a personalized view of the importance of the pages.

Hence, to make the query-optimized PageRank approach better (i.e., by making it more user friendly), the proposed technique extended our earlier approach by introducing personalized PageRank that combines with a user query to rank the web documents, which is the main objective of the paper.

The major contributions of the proposed approach are as follows:

i. Incorporating importance of the document with its personalized PageRank is an innovative idea to rank the web documents. It updates the link structure according to the similarity score between the document and the user query, and thereby refine the retrieval results by bringing the required documents on the top.
ii. By using a traditional PageRank algorithm, search engines might return pages that may not give information satisfying user needs and preferences. Hence, by considering the personalized PageRank algorithm for restructuring the links based on the user query would be more beneficial while implementing it on the search engine with datasets that have a lot of citations and have good link structures such as Wikipedia databases, research journal databases, business databases etc.
iii. As the content of the web document is considered along with the personalized PageRank, hence the proposed approach achieved high performance by bringing the required documents on the top of the search results. Here, the content of each web document and the user query are converted into TF-IDF forms and then the similarity score is computed between them. Based on this similarity score, the required documents are retrieved on the top of the search results that reduces the searching time of the user.

iv. By re-ranking the web documents, relevancy of the results is enhanced. Here, the re-ranking is nothing but personalized ranking of our earlier query-optimized PageRank approach. The earlier approach is improved by using the optimization function [18]. The modified link structure is the input to the personalized PageRank algorithm and contains only those output links that are connected to relevant documents. (which here is the non-zero cosine-similarity with the user query).
v. By introducing a novel feature selection technique named as *TCFS*, the noise terms are removed from the corpus before the personalized PageRank starts. This makes the personalized pageranking process more effective.

Although much work has been done for ranking the web documents (as evident from the past literature) but those ranking mechanisms either completely ignore the content of the documents or are fully dependent on the user query. Hence, the realm of personalized PageRank combine with similarity scores between the relevant documents and the user query provides a relatively unexplored pool of opportunities. The proposed algorithm is implemented on different benchmark datasets and, the experimental results show the effectiveness of the proposed feature selection technique and the query-optimized personalized PageRank algorithm.

The remainder of this paper is organized as follows: Sect. 2 discusses the past work done in the ranking domain. The basic preliminaries required for the proposed approach are discussed in Sect. 3. Section 4 discusses the query-optimized personalized PageRank. Experimental work of the proposed approach is analyzed in Sect. 5. Section 6 concluded the work with some future enhancement.

2 Past work

The dynamic web contains a huge volume of digital documents, and it is growing very rapidly. This makes it difficult for the search engine to retrieve relevant results. A search engine needs to rank the documents in such a way that the retrieved results should be most relevant to the user. Among all the existing ranking techniques, *Spatial TF-IDF* is a technique that is used to rank the documents by incorporating spatial and textual features of the documents and is suggested by Ali et al. [25]. The authors have proposed another method named *spatial-keyword Inverted File for Points (SKIF-P)* for web document indexing. They implemented their algorithm on real and synthetic datasets and show that their technique is more efficient than existing ranking techniques. Chahal et al. have discussed a semantic-based new document ranking mechanism [8] where conceptual instances between the keywords are considered by building an ontology. Important relations among the keywords have been analyzed by the authors, and the importance of each web document is decided based on these relations. Experimentally, they have shown that their approach can outperform the existing ranking techniques. Derhami et al. [15] have proposed a Reinforcement Learning (RL) for web documents ranking. They considered each web document as a state, and a technique is developed by combining RL Rank and BM25 (a content-based algorithm) to rank the documents. Experimental results of LETOR and dotIR datasets show that their approach can achieve much better results than PageRank algorithm. Du and Hai [17] have suggested a semantic approach of web documents based on formal concept analysis. Their approach uses a combination of all three types of the web Mining (i.e., web content, usage, and structure). Empirical results show that the returned results are highly efficient and relevant to the user query.

Patterns or similar words of a document are combined to generate a topic. Topic models play a vital role in document ranking. Some of the primary research work has been done in this direction [29,47,52,62]. Bougouin et al. [6] have suggested a graph-based topic ranking mechanism for key-phrase extraction. Their approach generates topics by clustering the candidate key-phrases. Empirical results on benchmark datasets show that their method is better than the existing ranking method. In the similar line, multiple topic tracking which classifies the news articles either interesting or not for a specific user has been developed by Pon et al. [35]. Empirical results justify the performance of their approach compared to the traditional pattern and term-based models. A pattern-based topic model, which is an information filtering model is proposed by Gao et al. [20]. In their work, multiple topics are combined together to generate useful information for ranking the documents. Experimental results of different benchmark datasets justify the efficiency of the proposed work.

The present document ranking structure treats the user query as independent which overlooks the interests of the user. Working in this direction, a cumulative proximity expansion method has been proposed by Vuurens et al. [56]. The authors investigate that occurrences of query terms are very much useful for measuring the document's relevance. They have implemented their work on Newswire and web corpora and showed the effectiveness of their technique. Evi et al. [60] have proposed a quality-biased ranking that incorporates signals from passages based on a novel use of community question answering data. Their approach develops a set of methodologies to improve the term relevance estimates from which answering passages are extracted. Ranking experiments on two web test collections (GOV2 and ClueWeb09B) shows the efficiency of their approach. Fafalios et al. [19] suggested a ranking method which ranks the archive documents for structured queries. Probabilistic and stochastic ranking models are proposed by them which consider the timeliness, relativeness, and temporal

relations among the documents. For experimental purposes, they have used the New York Times annotated corpus which contains 1.8 million query and the results show the effectiveness of their approach. Deep learning architecture named as 'DeepRank' has been used by Pang et al. [33] for relevance ranking of documents. DeepRank captures the query term importance, proximity heuristics, and diverse relation requirements. Empirical results on LETOR4.0 and large clickthrough dataset show that DeepRank model outperforms the existing ranking methods and deep IR models.

The above discussed approaches are either fully dependent on the query or neglect the content of the web documents. Combining the personalized PageRank of documents with their relevance is an innovative idea to rank the web documents. It updates the link structure of documents based on the similarity score with the user query and thereby refine the retrieval results by bringing the required documents on the top. Experimental results on five benchmark datasets show the efficiency of the proposed ranking approach.

3 Basic preliminaries

3.1 TF-IDF

$TF\text{-}IDF$ [53] is a common technique which finds the importance of a term t in a given document d by considering its appearance in the whole corpus and is shown in Eq. 1.

$$TF\text{-}IDF_{t,d} = TF_{t,d} \times IDF_t \tag{1}$$

where,

$$TF_{t,d} = \frac{Number\ of\ t \in d}{|d|}$$

$|d|$ represents the total length of d, and

$$IDF_t = \log_{10}\left(\frac{Number\ of\ d \in P}{Number\ of\ d\ have\ t}\right)$$

3.2 Silhouette coefficient

Silhouette Coefficient [49] of a term t is defined using Eq. 2.

$$silhouette(t) = \frac{s(t) - c(t)}{max\big(c(t), s(t)\big)} \tag{2}$$

where $c(t)$ and $s(t)$ are the cohesion (how close is t to its own cluster) and separation score (how well separated is t from other clusters) of the term t respectively.

3.3 Fuzzy C-means

Fuzzy C-Means (FCM) algorithm [5] distributes a finite collection of n documents into c clusters. It returns a list of c cluster centroids along with a matrix which shows the degree of membership of each document to other clusters. It aims to minimize the following function as shown in Eq. 3.

$$T_m = \sum_{i=1}^{n}\sum_{j=1}^{c} v_{ij}^m ||d_{ij}||^2 \tag{3}$$

where $d_{ij} = x_i - c_j$ is the distance, m is the fuzzy coefficient and generally set to 2, c_j is the centroid(vector) of cluster j, x_i is the i^{th} document. $v_{ij} \in [0, 1]$ is the degree of membership of x_i with respect to c_j and subject to the following conditions:

$$\sum_{j=1}^{c} v_{ji} = 1, \quad i = 1, 2, 3, \ldots, n \quad \text{and}$$

$$0 < \sum_{i=1}^{n} v_{ij} < n, \quad j = 1, 2, 3, \ldots, c$$

One can iteratively find the values of c_j and v_{ij} updated with each iteration by using the Eqs. 4 and 5 respectively.

$$c_j = \frac{\sum_{i=1}^{n} v_{ij}^m - x_i}{\sum_{i=1}^{n} v_{ij}^m} \tag{4}$$

$$v_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{||d_{ij}||}{||x_i - c_k||}\right)^{\frac{2}{m-1}}} \tag{5}$$

3.4 Mutual information judge

The relationship between a class c and a term t is established by using Mutual Information Judge (MI) [26] which mainly focus the information of t in c. The MI is computed using the Eq. 6.

$$MI(t, c) = \sum_{e_t \in \{0,1\}}\sum_{e_c \in \{0,1\}} Prob(e_t, e_c)\log_2 \frac{Prob(e_t, e_c)}{Prob(e_t)Prob(e_c)} \tag{6}$$

where the Bernoulli variables e_t and e_c are defined as

$$e_t = \begin{cases} 1, & \text{if } t \in d \\ 0, & \text{otherwise} \end{cases} \quad \text{and}$$

$$e_c = \begin{cases} 1, & \text{if } d \in c \\ 0, & \text{otherwise} \end{cases}$$

4 Propose approach

This Section discusses our earlier Query-optimized PageRank approach [42] briefly and the current Query-optimized personalized PageRank technique in detail.

4.1 Query-optimized PageRank

The following steps are used for query-optimized PageRank algorithm:

1. Initially, all the documents of a given corpus are pre-processed and converted into vector forms using Step 1 of Sect. 4.2.
2. Top l (l = 1 or 2)[2] terms are selected as the query term whose average TF-IDF values are maximum in the corpus. The reason for selecting the length of the query as one or two terms is, from literature [54] it has been found that most of the queries are very short (i.e., either one or two terms).
3. Cosine-similarity is calculated between each document and the query. The documents which are highly dissimilar (cosine similarity is zero) are discarded from the corpus and then the ranks of the remaining documents are calculated using PageRank algorithm. The main focus of the approach is that a surfer who is searching for a query on the web should only jump to those web documents which are highly correlated with the query.
4. After the link structure of documents gets modified using PageRank algorithm, adjustment of the weights of documents are made by considering the damping factor. Initially all the documents get same importance (i.e., same weight). Next the rank of a web document p_i is updated by adding the importance of the incoming links to the current rank score of p_i. This process is repeated for every document of the corpus. A rank matrix r is created which stores the updated rank of each web document after incorporating the damping factor to r. The following steps are used to compute the PageRank:
   i. Consider a directed graph G of k nodes and $\frac{k(k-1)}{2}$ edges, where each node is a web document and each edge represents the relationship between two documents. When page i refers to document j, then a directed edge will be added from node i to the node j in G.
   ii. Though all the documents that are linked by a single document will get equal importance initially, hence if a node has n outgoing edges, then the importance of each document will be $\frac{1}{n}$. Let A be the transition matrix of the graph G and represented as,

$$A = \begin{bmatrix} x_{11} & x_{12} & x_{13} \ldots x_{1n} \\ x_{21} & x_{22} & x_{23} \ldots x_{2n} \\ \vdots & \vdots & \vdots \ \ddots \ \vdots \\ x_{k1} & x_{k2} & x_{k3} \ldots x_{kn} \end{bmatrix}$$

   where x_{ij} is the link from document i to j.
   iii. Let v is the initial rank vector whose all the entries are $\frac{1}{k}$, because initially all web documents received equal importance. The rank of a web document i will be updated by adding the importance of the incoming links to the current value of i. It is same as multiplying the transition matrix A with the initial rank vector v. Hence, after first the iteration, the new importance vector become $v_1 = Av$. We keep iterating this step and it generates the sequences $v, Av, A^2v, A^3v, \cdots, A^kv$ which is the PageRank of the web graph G.
   iv. Since the experimental dataset is large, the graph G may not be connected. Thus, one requires an unambiguous meaning of the rank of a document for such directed web graph. To overcome this problem, damping factor (p) is used which is a positive constant lying between 0 and 1. The typical value of damping factor is 0.85. Equation 7 is used to compute the PageRank of G.

$$PageRank(G) = (1 - p)A + pB \tag{7}$$

   where,

$$B = \frac{1}{n} \begin{bmatrix} 1 & 1 & 1 \ldots 1 \\ 1 & 1 & 1 \ \ldots 1 \\ \vdots & \vdots & \vdots \ \ddots \ \vdots \\ 1 & 1 & 1 \ldots 1 \end{bmatrix}$$

4.2 Query-optimized personalized pageRank

To improve the retrieved results of the above query-optimized PageRank, we combine the personalized pageRank with the content of the relevant documents. The new ranks of the web documents which are computed using personalized PageRank are discussed using the following steps:

1. *Data acquisition and Pre-processing of the corpus*: Consider a corpus P having q classes. All the documents are pre-processed which includes lexical-analysis, stop-word elimination, removal of HTML tags, stemming [3], and then index terms are extracted. Documents of all

[2] the query either having one top term or two top terms

[3] http://snowball.tartarus.org/algorithms/porter/stemmer.html

q classes are put together, which constitute the dimension of P as $b \times l$, where b and l represent number of documents and terms respectively. Table 1 shows the document-term matrix where t_{ij} indicates the weight of the jth term in ith document.

2. *Document-Term cluster formation*:
   FCM clustering algorithm is run on the corpus P and divided all the terms of different documents of P into k doc-term clusters (dt_p) where $dt_p = \{dt_1, dt_2, \ldots, dt_k\}$ by bringing similar terms into the same cluster. Here, each dt_p is of dimension $b \times n$ (i.e., the number of documents remain same for each cluster but the number of terms get reduced by clustering). The reason for choosing FCM among the existing clustering techniques is that it is a soft clustering algorithm where the fuzziness can be exploited to create a more crisp behaving technique which generates better results, and it is one of the best algorithms compared to other hard clustering algorithms used for text data [5]. Next objective is to select the significant terms from each of the k clusters for maintaining the uniformity without excluding any collection.
3. *Term-Term Correlation based Feature Selection (TCFS)*:
   The following steps discuss how important features are selected from each cluster $dt_p, \forall p \in [1,k]$.

   (i) Frequency-based correlation (CF) calculation:
   First, the frequency-based correlation measure between every pair of terms i and j of each cluster dt_p is calculated using Eq. 8.

$$CF_{ij} = \sum \lim_{m \in p} f_{im} * f_{jm} \tag{8}$$

   where, f_{im} and f_{jm} represent the frequency of i^{th} and j^{th} terms in the m^{th} document of the cluster dt_p.

   (ii) Constructing association matrix:
   An association matrix shown in the Table 2 is constructed where each entry represents the association or frequency-based correlation measure between the terms t_i and t_j.

   (iii) Normalizing CF_{ij}:
   The frequency-based correlation measure CF_{ij} is normalized (named as normalized correlation measure (NCM)) using Eq. 9 which float the correlation values between 0 and 1 as shown in the Table 3. All the diagonal values of NCM_{ij} are 1 as $i = j$.

$$NCM_{ij} = \frac{CF_{ij}}{CF_{ii} + CF_{jj} - CF_{ij}} \tag{9}$$

   (iv) Semantic centroid vector generation:
   For each term t_i (i.e., for each row of NCM), the mean is calculated and all the means generate an n-dimensional vector named as semantic centroid vector sc_p. Each component of sc_p is shown in the Eq. 10.

$$sc_{p_i} = \frac{\sum_{j=1}^{n} NCM_{ij}}{n}, \quad 1 \le i \le n \tag{10}$$

   Each component of the semantic centroid vector is represented as

$$\begin{bmatrix} sc_{p_1} = \frac{(NCM_{11} + NCM_{12} + NCM_{13} + \cdots + NCM_{1n})}{n} \\ sc_{p_2} = \frac{(NCM_{21} + NCM_{22} + NCM_{23} + \cdots + NCM_{2n})}{n} \\ sc_{p_3} = \frac{(NCM_{31} + NCM_{32} + NCM_{33} + \cdots + NCM_{1n})}{n} \\ \vdots \qquad\qquad \vdots \\ sc_{p_n} = \frac{(NCM_{n1} + NCM_{n2} + NCM_{n3} + \cdots + NCM_{nn})}{n} \end{bmatrix}$$

   (v) Selecting important features:
   a. *Calculating silhouette coefficient*:
   The silhouette coefficient ($silhout$) of the term $t_i \in dt_p$ is computed using Eq. 11. Cohesion (coh) measures how cohesive is the term, $t_i \in dt_p$ to the centroid, $sc_p \in dt_p$ and is shown in Eq. 12 and separation (sep) measures how well separated a term, $t_i \in dt_p$ from the semantic centroid of other clusters, $sc_m, \forall m \in [1, k]$ and $m \neq p$ which is shown in the Eq. 13.

$$silhout(t_i) = \frac{sep(t_i) - coh(t_i)}{max\big(coh(t_i), sep(t_i)\big)} \tag{11}$$

$$coh(t_i) = (||sc_p - t_i||) \tag{12}$$

$$sep(t_i) = \min\big(||sc_m - t_i||\big) \tag{13}$$

   where, sc_m is the semantic centroid of the m^{th} cluster.

   b. Finally, the terms are ranked based on their silhouette coefficient scores and among them top 'm%' terms (for experimental work, we choose m = 10 of the total terms it is decided empirically) are selected as the important features for the cluster d_{tp}.

   (vi) By repeating Step 3 (i-v) for all k doc-term clusters, top m% important terms are generated for each doc-term cluster. After generating top $m\%$ important terms for each doc-term cluster, the noise terms are ignored from each cluster. The documents which do not contain any of these important terms are discarded from the cluster.

The details of this feature selection technique are generalized in Algorithm 1 for the implementation purposes.

Algorithm 1: Term-Term Feature Selection

Input: document-term matrix and term frequency of cluster dt_p
Output: $Top[\,] \leftarrow$ important features of dt_p
$CF[\,][\,] \leftarrow \phi$ // correlation measure matrix
$NCM[\,][\,] \leftarrow \phi$ // normalized correlation measure matrix
$Silhoutte[\,] \leftarrow \phi$ // stores silhoutte coefficient score of all the terms
$Top[\,] \leftarrow \phi$
$sc_p \leftarrow \phi$ // semantic centroid of dt_p
for all terms $(i, j) \in dt_p$ **do**
  $sum \leftarrow \phi$
  for all document $k \in dt_p$ **do**
    $sum \leftarrow sum + (f_{ik} * f_{jk})$
  end for
  $CF_{ij} \leftarrow sum$
end for
for all terms $(i, j) \in CF$ **do**
  $NCM_{ij} \leftarrow \frac{CF_{ij}}{(CF_{ii}+CF_{jj}-CF_{ij})}$
end for
for $i \in [1, n]$ **do**
  // n represents total no. of terms
  $sum \leftarrow \phi$
  for $j \in [1, n]$ **do**
    $sum \leftarrow sum + NCM_{ij}$
  end for
  $sc_i \leftarrow \frac{sum}{n}$ // semantic centroid
end for
for $i \in [1, n]$ **do**
  $Silhoutte[i] \leftarrow silhout(sc_p, t_i)$
end for
$Top[\,] \leftarrow$ select top m% terms from $Silhouette[\,]$ after ranking the terms
return $Top[\,]$

4. *Query vector generation:*
   Among the top $m\%$ important terms of each cluster dt_p of the corpus P, top l terms (l = 1 or 2 and the reason for such selection of l values is discussed in Sect. 4.1) based on their silhouette coefficient scores are selected to generate the query $q_p, \forall p \in$ [1,k] for that cluster. As we are working on Bag-of-words model, the order of the terms in the query does not matter.
5. *Computing similarity between the documents and the query:*
   Using Eq. 14, cosine-similarity ($cosine\text{-}sim$) is computed between each document $d_i \in dt_p$ and the query vector q_p. Then the documents of each dt_p are arranged based on their cosine-similarity scores, and those documents are discarded from the corpus P whose scores fall below a threshold of 0.5[4].

$$cosine\text{-}sim(d_i, q_p) = \frac{d_i.q_p}{||d_i|| * ||q_p||} \tag{14}$$

   All the documents of each dt_p are merged together which generates a new corpus P_{new}.

[4] the threshold is decided by the experiment

Table 1 Document-term matrix

	t_1	t_2	t_3	...	t_l
d_1	t_{11}	t_{12}	t_{13}	...	w_{1l}
d_2	t_{21}	t_{22}	t_{23}	...	t_{2l}
d_3	t_{31}	t_{32}	t_{33}	...	t_{3l}
⋮	⋮	⋮	⋮	⋱	⋮
d_b	t_{b1}	t_{b2}	t_{b3}	...	t_{bl}

Table 2 Association matrix

	t_1	t_2	t_3	⋯	t_n
t_1	CF_{11}	CF_{12}	CF_{13}	⋯	CF_{1n}
t_2	CF_{21}	CF_{22}	CF_{23}	⋯	CF_{2n}
t_3	CF_{31}	CF_{32}	CF_{33}	⋯	CF_{3n}
⋮	⋮	⋮	⋮	⋱	⋮
t_n	CF_{n1}	CF_{n2}	CF_{n3}	⋯	CF_{nn}

6. *Computing the personalized PageRank of P_{new}:*
   Personalized PageRank is used to rank the documents of P_{new} assuming that it contains n number of documents and is discussed in the next step. At the beginning, all the documents of P_{new} received the same importance. Next, their ranks are updated by adding the importance of the incoming links to their current rank scores. This technique is repeated for all the documents of P_{new}.
7. *Applying Link-Based Technique on the corpus P_{new}:*
   In Link-based techniques, the personalized PageRank is evaluated for all the web documents having non-zero cosine-similarity of the corpus P_{new} which improved the earlier PagaRank approach. The link-based approach is developed using the following steps:

   (i) Adjacency matrix construction:
   We represented the web by a directed graph $G = \{V, E\}$ where vertices V is considered as the set of web documents and the edges E represents the hyperlink from vertex U to V. The outlink information between web documents have been stored according to the format of dataset (for example purposes, we have shown the link structure of few documents) and demonstrated in Table 4.

The outlink information of web documents can be easily obtained from each row of Table 4. For example, the last row tells us the outlink information of the fifth web document and explained in Table 5 for better understanding.

Table 3 Normalized correlation measure

	t_1	t_2	t_2	$\cdots$	t_n
t_1	NCM_{11}	NCM_{12}	NCM_{13}	$\cdots$	NCM_{1n}
t_2	NCM_{21}	NCM_{22}	NCM_{23}	$\cdots$	NCM_{2n}
t_3	NCM_{31}	NCM_{32}	NCM_{33}	$\cdots$	NCM_{3n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_n	NCM_{n1}	NCM_{n2}	NCM_{n3}	$\cdots$	NCM_{nn}

Table 4 Outlinks to web documents

Web document	Outlinks to
0	1:0, 3:2, 4:1
1	0:2, 2:1, 5:1
2	1:1,3:1, 5:4
3	0:2, 4:1, 5:3
4	0:1,3:3,5:1
5	0:5, 1:3, 2:3

Table 5 Outlink information of fifth web document

Number of outlinks	Destination web document
5	0
3	1
3	2

The adjacency matrix A of the outlinks information is defined as follows

$$A = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \cdots & \beta_{1n} \\ \beta_{21} & \beta_{22} & \beta_{23} & \cdots & \beta_{2n} \\ \beta_{31} & \beta_{32} & \beta_{33} & \cdots & \beta_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta_{n1} & \beta_{n2} & \beta_{n3} & \cdots & \beta_{nn} \end{bmatrix}$$

where β_{ij} is the number of outlinks from document i to document j. To understand the calculation of the adjacency matrix A, we have shown the adjacency computation A' as

$$A' = \begin{bmatrix} 0 & 0 & 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 4 \\ 2 & 0 & 0 & 0 & 1 & 3 \\ 1 & 0 & 0 & 3 & 0 & 1 \\ 5 & 3 & 3 & 0 & 0 & 0 \end{bmatrix}$$

The adjacency matrix is normalized by dividing the row sum to each row. This normalized form is used for personalized PageRank. The normalized forms of A and A' are denoted by H and H' respectively.

$$H = \begin{bmatrix} \frac{\beta_{11}}{\beta_{11}+\beta_{12}+\cdots+\beta_{1n}} & \cdots & \frac{\beta_{1n}}{\beta_{11}+\beta_{12}+\cdots+\beta_{1n}} \\ \frac{\beta_{21}}{\beta_{21}+\beta_{22}+\cdots+\beta_{2n}} & \cdots & \frac{\beta_{2n}}{\beta_{21}+\beta_{22}+\cdots+\beta_{2n}} \\ \vdots & \ddots & \vdots \\ \frac{\beta_{n1}}{\beta_{n1}+\beta_{n2}+\cdots+\beta_{nn}} & \cdots & \frac{\beta_{nn}}{\beta_{n1}+\beta_{n2}+\cdots+\beta_{nn}} \end{bmatrix}$$

$$H' = \begin{bmatrix} 0 & 0 & 0 & 2/3 & 1/3 & 0 \\ 2/4 & 0 & 1/4 & 0 & 0 & 1/4 \\ 0 & 1/6 & 0 & 1/6 & 0 & 4/6 \\ 2/6 & 0 & 0 & 0 & 1/6 & 3/6 \\ 1/5 & 0 & 0 & 3/5 & 0 & 1/5 \\ 5/11 & 3/11 & 3/11 & 0 & 0 & 0 \end{bmatrix}$$

(ii) Calculation of personalized PageRank:
The personalized PageRank for a web document i is evaluated by considering all other web documents' contributions to the PageRank of i. In the graph G, the contribution of a vertex V_1 to the PageRank of another vertex V_2 is described in terms of personalized PageRank [2]. For a row normalized adjacency matrix H, the PageRank PR_i (for document i) is determined as

$$PR_i \leftarrow \alpha * PR_i * H + (1-\alpha) * \vartheta \tag{15}$$

In the Eq. 15, ϑ is the teleportation vector and $\alpha \in [0, 1]$ is scaling parameter. In practice, the scaling parameter is normally assumed as 0.85 [27]. To calculate the personalized PageRank contribution vector for ith web document, the ith bit of ϑ is set to 1 and remaining bits are set as 0. At the beginning, PR_i is set to

$$PR_i = \left(\frac{1}{n}, \frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right)$$

where n is the total number of documents of P_{new} and it updated iteratively using Eq. 15. PR_i then stored in the i^{th} row of personalized PageRank (ppr) matrix i.e., $ppr[i, :] \leftarrow PR_i$. The computation of personalized PageRank (ppr) is described explicitly in Algorithm 2 for implementation purposes. The overview of the query-optimized personalized PageRank technique is shown in Fig. 1.

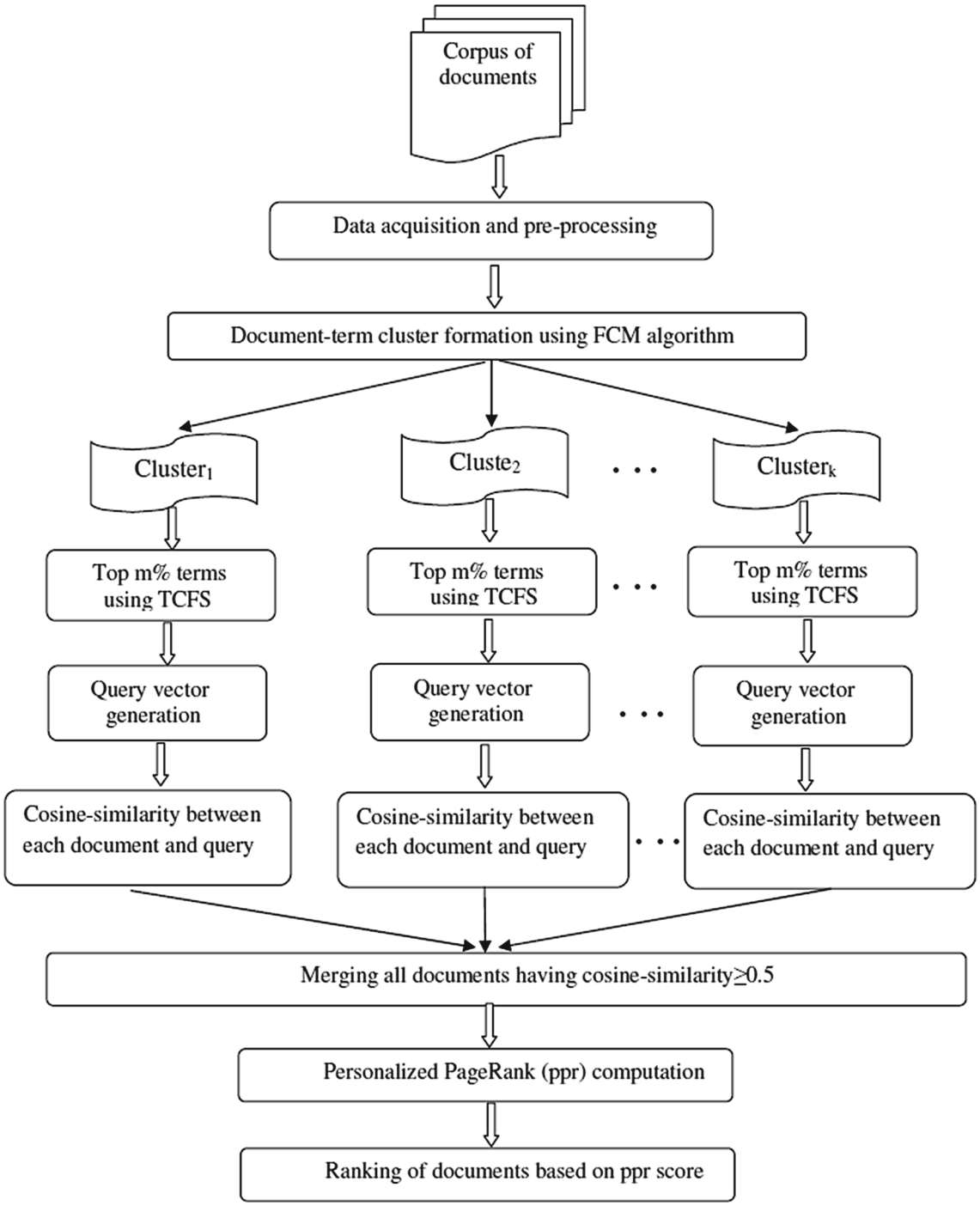

Fig. 1 Query-optimized personalized PageRank

Algorithm 2: Personalized PageRank

Input: $n \leftarrow$ size of the adjacency matrix, P_{new}
$PR_{i0} \leftarrow$ initial $1 \times n$ vector at iteration 0 // set to $\left(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n}\right)$
$\alpha \leftarrow$ scaling parameter
$H \leftarrow$ Row-normalized adjacency matrix
$\beta \leftarrow$ convergence tolerance // set to 0.001 (user dependent)
$\vartheta \leftarrow$ teleportation vector
Output: $ppr \leftarrow$ personalized PageRank matrix of dimension $n \times n$
$\vartheta \leftarrow$ [0,0,$\cdots$,0] //teleportation vector is initially null
for i in 1 to n **do**
  ϑ[i]$\leftarrow$ 1 // set the i^{th} bit of ϑ[i] to 1 to calculate the PageRank contribution vector of i^{th} web document
  $k \leftarrow$ 0 //number of iterations
  residual $\leftarrow$ 1
  $PR_i \leftarrow PR_{i0}$
  while residual $\geq \beta$ **do**
    $PR_{i_{prev}} \leftarrow PR_i$ //assign current value of PR_i to its previous value $PR_{i_{prev}}$
    $k \leftarrow k + 1$
    $PR_i \leftarrow \alpha * PR_i * H + (1\text{-}\alpha) * \vartheta$
    residual$\leftarrow$ norm-1 distance between PR_i and $PR_{i_{prev}}$
  end while
  $ppr[i, :] \leftarrow PR_i$ // values of PR_i stored in i^{th} row of ppr
  ϑ[i] $\leftarrow$ 0
end for

5 Experimental work

For experimental purposes, five benchmark datasets are used (DMOZ [5], Classic4 [6], Reuters [7], 20-Newsgroups [8], and WebKB [9]). A brief description about each dataset is mentioned below:

i. *DMOZ* is one of the largest dictionaries on the web, and it has 14 categories of web documents. Out of 69,068 documents, we have used 38,000 documents for training and 31,068 for testing purposes. Many documents have no content or very less content. Total number of terms is 60320 out of which 39886 are considered for training.
ii. *20-Newsgroups* is a standard machine learning dataset, and it has 11,293 training and 7528 testing documents classified into 20 classes. All the classes are considered for experimental purposes. Some of the documents have no content. Among 52,422 terms, 32,270 are used for training and the rest are used for testing.
iii. *Classic4* is a text mining dataset and it has 4257 training and 2838 test documents classified into four classes - *cisi, med, casm, and cran*, having 1460, 1033, 3204, and 1400 respectively. All the classes are considered in evaluation. The total terms contained in all documents is 21299 and for training documents is 15971.
iv. *Reuters* is a well-known machine learning dataset having eight categories of documents, and all categories are used for the experimental purposes. Among these documents, 5485 are used for training and 2189 are used for testing. The total number of terms used is 17,582, and among them 13,531 are used for training.
v. *WebKB* is a popular machine learning dataset which has four categories of documents from four different college websites. For experiment, all categories of documents are used in which 2803 documents are considered for training and 1396 for testing. The total number of terms of all these documents is 7606 and from that 7522 terms are used for training. Documents having less content are filtered out as discussed in the next paragraph, but they are counted during the ranking process.

For experimental purposes, we have discussed how the adjacency matrix is generated for WebKB dataset and the same technique has been applied for all other datasets. A collection of 4199 different hosts are considered as available on WebKB dataset (Host graph format). Initially, the adjacency matrix of dimension 4199×4199 for the Host graph is computed and normalized as discussed in Step 7 of Sect. 4.2. The dataset of 4199 web documents has been filtered out to the dataset of 994 web documents. The filtering process is done using four steps as mentioned below:

(i) Initially, only those web documents are selected from the dataset where the human assigned labels are available and all other web documents are discarded.
(ii) Among those human assigned labels of web documents, all working links are selected.
(iii) Next, the content of those working links of web documents are extracted and stored in a corpus in text file format.
(iv) At the end, only those web documents from the corpus are selected which have content of at least 1KB. The reason for selecting at least 1KB of web document is to conduct the personalized PageRank smoothly because the content of the web document is more and it is important for experimental purposes.

5.1 Result analysis of query-optimized pageRank

This section discusses the experimentation of the earlier query-optimized PageRank. To implement the PageRank algorithm combine with the content of the document, a

[5] http://www.dmoz.org
[6] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/
[7] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[8] http://qwone.com/~jason/20Newsgroups/
[9] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

Table 6 Query: Agriculture (NFr = 0.333)

Cosine-similarity ranking	Query-optimized PageRanking
Agriculture	Agriculture
Algeria	Africa
Albania	Algeria
Almond	African_union
Accountancy	Albania
Africa	Almond
2005_Atlantichurricane_season	2005_Atlantichurricane_season
Aberdeen	Aberdeen

Table 7 Query: Massachusetts (NFr = 0)

Cosine-similarity ranking	Query-optimized PageRanking
Abu_dhabi	Abu_dhabi
2004_Atlantichurricane_season	2004_Atlantichurricane_season
Alternative_rock	Alternative_rock

Table 8 Query: Roman Empire (NFr =0.661)

Cosine-similarity ranking	Query-optimized PageRanking
1st_century_BC	14th_century
13th_century	13th_century
10th_century	Abacus
5th_century	9th_century
3rd_century	11th_century
11th_century	10th_century
6th_century	5th_century
Abacus	6th_century
14th_century	1st_century
1st_century	Akkadian_Empire
Akkadian_Empire	16th_century
9th_century	Aachen
16th_century	1st_century_BC
Aachen	3rd_century

benchmark research dataset called DBpedia [10] has been chosen. The reason for choosing such a dataset is that it has both content and link structure which are needed for the experimental work. But the limitation of this dataset is that it does not have a sufficient set of relevant documents for any given query, which makes difficult to compute the accuracy of our earlier approach. To handle such problem, we used a method called Spearman's footrule [16]. For checking the efficiency of our earlier approach, we compared the accuracy of query-optimized PageRank with the cosine-similarity ranking of web documents. The query vector is generated in the same way as discussed in Step 4 of Sect. 4.2. Here, we have considered only monogram (l=1) and bi-grams (l=2) queries for experimental purposes and the reason is already discussed in Sect. 4.1. The followings are some of the links of dbpedia dataset used for experimental purposes.

- Links/amsterdammuseum_links
- Links/dailymed_links
- Links/eunis_links
- En/external_links_en
- En/infobox_properties_en
- Links/italian_public_schools_links
- Sv/labels_en_uris_sv
- Pl/long_abstracts_en_uris_pl
- Links/revyu_links
- Links/yago_links

Assuming that dbpedia dataset contains N documents, where the documents are ranked between 1 and N. The Spearman's footrule method is applied to both query-optimized and cosine-similarity ranking techniques for measuring their accuracies. No ties are allowed as the rankings generated by each of the two techniques being compared is basically a permutation of the other. Let's say that the result of the rankings are permutations σ_2 for the ranking based on cosine-similarity and σ_1 for query-optimized PageRank. The ranking results of the top 'k' documents of each of the two techniques is turn out to be over S, where S the set of overlapping results between the two ranking techniques. Equation 16 is used to compute the Spearman's footrule.

$$Fr^{|S|}(\sigma_1, \sigma_2) = \sum_{i=1}^{|S|} |(\sigma_1(i) - \sigma_2(i)| \tag{16}$$

[10] http://wiki.dbpedia.org/Datasets

Table 9 Query: General History (NFr =0.714)

Cosine-similarity ranking	Query-optimized PageRanking
9th_century.txt	12th_century
1st_century_BC.txt	11th_century
12th_century.txt	9th_century
6th_century.txt	13th_century
African_slave_trade.txt	10th_century
Acceleration.txt	20th_century
13th_century.txt	4th_century
Adriaen_van_der_Donck.txt	1st_century_BC
Alfred_the_Great.txt	6th_century
10th_century.txt	African_slave_trade
Acts_of_Union_1707.txt	Acceleration
20th_century.txt	Adriaen_van_der_Donck
4th_century.txt	Alfred_the_Grea
11th_century.txt	Acts_of_Union_1707

Table 10 Query: Mediterranean (NFr = 0.611)

Cosine-similarity ranking	Query-optimized PageRanking
Algiers	Africa
5th_century	Algeria
Africa	Algiers
Akkadian_Empire	5th_century
Akrotiri_and_Dhekelia	19th_century
Albania	Airship
Algeria	2004_Indian_Ocean_earthquake
Almond	Albania
Aircraft_Carrier	Aircraft_Carrie
Albigensian_crusade	Albigensian_crusade
19th_century	Adelaide
Adelaide	Aberdeen
Aberdeen	Alexander_the_Great
Airship	Akkadian_Empire
2004_Indian_Ocean_earthquake	Akrotiri_and_Dhekelia
Alexander_the_Great	Almond

The value of $Fr^{|S|}$ is computed by dividing the obtained result with its maximum value. The achieved value is independent on the size of the overlap S and lies between 0 and 1. Following three conditions are observed based on the value of S:

i. When the ranking lists of both query-optimize and cosine-similarity ranking techniques are equal, then $Fr^{|S|}$ is zero.
ii. When $|S|$ is even, $Fr^{|S|}$ obtained the maximum value of $\frac{1}{2}S^2$.
iii. When $|S|$ is odd, $Fr^{|S|}$ obtained the maximum value of $\frac{1}{2}(|S|+1)(|S|-1)$.

Equation 17 is used to compute the normalized Spearman's footrule (NFr) for $|S| > 1$.

$$\text{NFr} = \frac{Fr^{|S|}}{max\ Fr^{|S|}} \tag{17}$$

Thus, NFr will range between 0 and 1. Ranking using query-optimized PageRank and cosine-similarity for uni-gram and bi-gram queries are shown in Tables 6, 7, 8, 9, 10, 11, and 12. In these Tables, NFr represents the accuracy gained by the query-optimized PageRank over cosine-similarity ranking. Since the retrieved documents are very less, and the link structure could not refine the ranks much based on the cosine-similarity of the documents with

Table 11 Query: Catholic (NFr = 0.704)

Cosine-similarity ranking	Query-optimized PageRanking
9th_century	15th_century
Albigensian_Crusade	Africa
1755_Lisban_earthquake	14th_century
Abbot	Algiers
Albrecht_Rodenbach	Algeria
16th_century	16th_century
15th_century	17th_century
17th_century	9th_century
14th_century	Addis_Ababa
Addis_Ababa	Akbar
Adolf_Hitler	Adolf_Hitler
Aberdeen	1928_Okeechobel_Hurricane
1928_Okeechobel_Hurricane	Albert_Einstein
Algiers	Albigensian
Alfred_Hitchcock	1755_Lisbon
Albrecht_D%C3%BCrer	Abbo
1896_Summer_Olympics	Albrecht_Rodenbach
Akbar	Aberdeen
Algeria	Alfred_Hitchcock
Africa	Albrecht_D%C3%BCrer
Albert_Einstein	1896_Summer_Olympics

Table 12 Query: Civil (NFr = 0.857)

Cosine-similarity ranking	Query-optimized PageRanking
2005_Lake_Tanganyika_earthquake	Algeria
African_Great_Lakes	Algerian_Civil_War
Aircraft	African_Union
15th_century	21st_century
African_Union	1st_century_BC
Abidjan	15th_century
African_American_literature	19th_century
19th_century	17th_century
17th_century	African_Great_Lakes
21st_century	2005_Lake_Tanganyika_earthquake
Alexsandr_Vasilevsky	Aircraft
1st_century_BC	Abidjan
Algerian_Civil_War	African_American_literature
Algeria	Alexsandr_Vasilevsky

the query, hence for the query "Massachusetts" as shown in Table 7, the Spearman coefficient turned out to be 0. A non-zero spearman's score says that the ranking given by the query-optimized PageRank puts forward a new direction of research for the modified PageRank which is query dependent.

5.2 Result analysis of proposed feature selection technique

Table 13 shows different parameters used for the feature selection technique. All top $m\%$ important terms (m = 10) (as discussed in step 3(v) of Sect. 4.2) are combined together to generate the training feature vector for classifiers. Comparison results of the proposed TCFS feature selection technique with other well-known existing techniques

Table 13 Size (approximate) of the input feature vector

Dataset	No. of training documents	No. of testing documents	No. of terms used for training	10% of terms
20-NG	11293	7528	32270	3230
DMOZ	38000	31068	39886	3990
Classic4	4257	2838	15971	1600
Reuters	5485	2189	13531	1350
WebKB	2803	1396	7522	750

Table 14 Comparisons on 20-NG Dataset

Classifier	MI	CHI-2	GINI	IG	TCFS
LinearSVM	0.9328	0.9455	0.9364	0.9465	**0.9536**
G-NB	0.8914	0.8958	0.8981	0.9051	**0.9082**
B-NB	0.9399	0.9101	0.9335	0.9399	0.9138
M-NB	0.9223	0.9347	0.9371	0.9373	**0.9398**
Adaboost	0.8567	0.8620	0.8632	0.8754	0.8668
DT	0.8233	0.8498	0.8608	0.8516	0.8564
RF	0.8929	0.8945	0.8970	0.8929	**0.9051**
ET	0.9326	0.9453	0.9341	0.9464	0.9286

Table 15 Comparisons on DMOZ Dataset

Classifier	MI	CHI-2	GINI	IG	TCFS
LinearSVM	0.9512	0.9591	0.9405	0.9795	0.9620
G-NB	0.9213	0.9359	0.9293	0.9395	**0.9479**
B-NB	0.9261	0.9064	0.9191	0.9019	0.9098
M-NB	0.9385	0.9238	0.9306	0.9456	**0.9468**
Adaboost	0.8736	0.8883	0.8474	0.8581	**0.8986**
DT	0.8428	0.8444	0.8486	0.8555	0.8484
RF	0.9184	0.9182	0.9167	0.9190	0.8955
ET	0.9454	0.9221	0.9478	0.9478	0.9476

(Mutual Information (MI), Chi-square (CHI-2), GINI [51], and Information Gain (IG) [59]) are given in Tables 14, 15, 16, 17 and 18 respectively. Eight classifiers like LinearSVM, Gaussian-Naive Bayes (G-NB), Binomial Naive Bayes (B-NB), Multinomial Naive Bayes (M-NB), Adaboost, Decision Trees (DT), Random Forest (RF), and Extra Trees (ET) are used for classification of document on different datasets. All ensemble classifiers are 10-class classifiers. We have adapted the above techniques to check the performance of the proposed feature selection with respect to other conventional techniques. Equation 18 is used to measure the performance of each classifier. The bold results indicate the highest F-measure obtained by the proposed feature selection technique using the corresponding classifier. From the results, it is observed that the proposed feature selection technique is either comparable or better than the conventional techniques.

$$\begin{aligned} \textit{Precision } (p) &= \frac{(\text{relevant}_{\text{documents}}) \cap (\text{retrieved}_{\text{documents}})}{\text{retrieved}_{\text{documents}}} \\ \text{Recall(r)} &= \frac{(\text{relevant}_{\text{documents}}) \cap (\text{retrieved}_{\text{documents}})}{\text{relevant}_{\text{documents}}} \\ \text{F} - \text{measure(f)} &= 2 * (\frac{p * r}{p + r}) \end{aligned} \tag{18}$$

5.2.1 Tuning hyper-parameters:

Tuning of hyper-parameter of different classifiers are given below:

Table 16 Comparisons on Classic4 Dataset

Classifier	MI	CHI-2	GINI	IG	TCFS
LinearSVM	0.9441	0.9653	0.9759	0.9786	**0.9799**
G-NB	0.9322	0.9495	0.9350	0.9439	**0.9692**
B-NB	0.9202	0.9687	0.9149	0.9155	0.8971
M-NB	0.9509	0.9456	0.9477	0.9535	**0.9697**
Adaboost	0.8541	0.8541	0.8541	0.8456	0.8456
DT	0.8500	0.8479	0.8529	0.8475	**0.8584**
RF	0.9157	0.9283	0.9188	0.9138	0.9173
ET	0.9235	0.9475	0.9563	0.9693	0.9562

i. *LinearSVM:* Cs = [0.001, 0.01, 0.1, 1, 10], gammas = [0.001, 0.01, 0.1, 1], param_grid = {'C': Cs, 'gamma': gammas}
ii. *Naive Bayes:* Prior Probabilities = [0.65, 0.35]
iii. *Adaboost:* n_estimators = 12, max_depth = 5, subsample = 0.5, random_state = 0, number of classifiers used = 10
iv. *Decision Trees:* max_depth = 10, min_sample_splits = 40%, min_samples_leaf = 5
v. *Random Forest:* n_estimators = 11, max_depth = 4, subsample = 0.5, random_state = 0, number of classifiers used = 10
vi. *Gradient Boosting:* learning_rate = 0.1, n_estimators = 280, max_depth = 4, subsample = 0.4, random_state = 0, number of classifiers used = 10

Table 17 Comparisons on Reuters Dataset

Classifier	MI	CHI-2	GINI	IG	TCFS
LinearSVM	0.9546	0.9468	0.9465	0.9489	**0.9671**
G-NB	0.8453	0.8430	0.8421	0.8456	**0.8559**
B-NB	0.7522	0.8428	0.8340	0.8193	0.7945
M-NB	0.9009	0.9057	0.9010	0.9071	0.8893
Adaboost	0.6320	0.6342	0.6348	0.6317	**0.6480**
DT	0.9057	0.8984	0.9060	0.8992	0.8976
RF	0.9291	0.9190	0.9295	0.9192	0.9176
ET	0.8936	0.9467	0.9142	0.9502	0.9456

Table 18 Comparisons on WebKB Dataset

Classifier	MI	CHI-2	GINI	IG	TCFS
LinearSVM	0.8704	0.8838	0.8680	0.8808	**0.8862**
G-NB	0.8038	0.7980	0.7943	0.7915	0.7861
B-NB	0.7038	0.7326	0.7188	0.7059	0.7098
M-NB	0.7854	0.7782	0.7899	0.7905	**0.7963**
Adaboost	0.8114	0.7931	0.7981	0.8097	0.8083
DT	0.7883	0.7978	0.7907	0.7874	**0.8186**
RF	0.8360	0.8370	0.8344	0.8289	**0.8587**
ET	0.8662	0.8876	0.8812	0.8564	0.8423

vii. *Extra Trees:* n_estimators = 20, learning_rate = 1, max_depth = 4, subsample = 0.6, random_state = 0, number of classifiers used = 10

5.3 Result analysis of query-optimized personalized pageRank

This Section discusses the experimental results of the proposed Query-optimized personalized PageRank technique in detail. At the beginning of the work, two sets of documents are formed:

i. One set contains all the original categories of documents of a dataset named as '$original_{category}$'.
ii. The other set contains newly formed clusters named as $new_{cluster}$, where a cluster may have documents that come from other categories of a dataset. New clusters are formed by combining all the documents of different categories of a dataset and then running FCM technique on those documents. The number of clusters formed for each dataset is same as the number of categories the dataset has. For example, 20-NG has seven categories, hence the number of clusters generated by the FCM algorithm is seven. To know which cluster of $new_{cluster}$ belongs to which category of $original_{category}$, a technique is developed and according to it, cluster i belongs to category j iff i contain maximum documents of j. This is done in order to compute the ranking process efficiently.

Table 19 Monogram and Bi-grams query words

Dataset	Monogram	Bi-grams
20-NG	Atheism	Atheism religion
DMOZ	Graphic	Graphic computer
Classic3	Trivial	Trivial unrealistic
Reuters	Bank	Bank rate
WebKB	Subject	Subject coursework

The query-optimized personalized PageRank is applied on $new_{cluster}$ to rank all the documents. The top 's'[11] ranked results of the $original_{category}$ and top 't'[12] ranked results of $new_{cluster}$ are considered for performance measurement of the ranking approach. Equations 19 and 20are used to compute the precision and recall respectively.

$$precision(p') = \frac{a}{b} \tag{19}$$

$$recall(r') = \frac{a}{d} \tag{20}$$

$$F\text{-}measure(f') = 2\left(\frac{p * r}{p + r}\right) \tag{21}$$

where, 'a' is the common documents between the top '$s\%$' documents of $original_{category}$ and the top 't%' documents of the $new_{cluster}$, 'b' is the top 't%' ranked results of $new_{cluster}$ and 'd' is the top '$s\%$' documents of $original_{category}$. For experimental purposes, we have discussed the '$Computer$' category named as $original_{comp}$ of 20-NG dataset which has 1952 documents. The proposed query-optimized personalized PageRank algorithm is run on both the new generated computer cluster named as new_{comp} and $original_{comp}$ to rank the documents of both clusters. We then find how many top 's%' documents of '$original_{comp}$' category match with the top 't%' documents of 'new_{comp}' from which the F-measure is computed as mentioned in Eq. 21.

5.3.1 Comparison of proposed personalized pageRank with cosine-similarity and traditional pageRank algorithm

The monogram and bi-grams queries generated from each dataset (shown in the Table 19) are used for cosine-similarity, PageRank, and proposed query-optimized personalized PageRank approaches. The proposed query-optimized personalized PageRank approach using monogram query is tested on each category of all the datasets, and their performances are shown in Tables 20, 21, 22, 23 and 24 respectively.

[11] decided experimentally

[12] decided experimentally

Table 20 Performance of the ranking approach on 20-NG

Class	No. of documents	Precision	Recall	F-measure
Alt	320	0.7271	0.7068	0.7168
Computers	1952	0.7359	0.7145	0.7250
Miscellaneous	390	0.6981	0.7255	0.7115
Recreation	1590	0.7491	0.7241	0.7364
Science	1580	0.7182	0.7093	0.7137
Social	399	0.7128	0.7245	0.7186
Talk	1297	0.7359	0.7145	0.7250
Overall	7528	0.7325	0.7167	0.7245

Table 21 Performance of ranking on DMOZ

Class	No. of documents	Precision	Recall	F-measure
Arts	1396	0.6765	0.6930	0.6847
Business	3384	0.6895	0.6834	0.6864
Computers	1494	0.6594	0.6633	0.6613
Games	5757	0.6606	0.6960	0.6778
Health	1491	0.6459	0.7145	0.6785
Homes	1405	0.6571	0.6631	0.6601
News	1504	0.6779	0.6723	0.6751
Recreation	1410	0.6883	0.7044	0.6963
Reference	1301	0.6686	0.6456	0.6569
Regional	1307	0.6989	0.6590	0.6784
Science	1390	0.6765	0.6930	0.6847
Shopping	6209	0.6866	0.7060	0.6962
Society	1505	0.6686	0.6705	0.6695
Sports	1515	0.6686	0.6705	0.6695
Overall	31068	0.6734	0.6864	0.6797

Table 22 Performance of ranking on Classic4

Class	No. of documents	Precision	Recall	F-measure
casm	1282	0.8232	0.8154	0.8194
cisi	584	0.8346	0.8532	0.8438
cran	560	0.8326	0.809	0.8206
Med	413	0.7935	0.8167	0.8049
Overall	2839	0.8078	0.8067	0.8072

Comparison of these three ranking techniques using mono-gram and bi-grams query are shown in Figures 2 and 3 respectively. From the figures, it is observed that the performance of query-optimized personalized PageRank is better than the query combined with cosine-similarity and query-optimized PageRank approach. The overall (i.e., average) F-meaure of each dataset shows the stability of the proposed ranking approach.

Table 23 Performance of ranking on Reuters

Class	No. of documents	Precision	Recall	F-measure
acq	696	0.8432	0.8345	0.8388
Crude	121	0.8156	0.8543	0.8345
Earn	1083	0.8323	0.8345	0.8334
Grain	10	0.8634	0.8365	0.8497
Interest	81	0.8543	0.8327	0.8433
Money-fx	87	0.8346	0.8124	0.8233
Ship	36	0.8312	0.8456	0.8383
Trade	75	0.811	0.8256	0.8182
Overall	2189	0.8351	0.8345	0.8348

6 Conclusion

This paper is an extension of our earlier approach which improves the existing PageRank by personalizing it and then combined with query to rank the web documents. In the earlier approach, an efficient ranking model is developed which improves the ranking mechanism by bringing the required

Table 24 Performance of ranking on WebKB

Class	No. of documents	Precision	Recall	F-measure
Project	168	0.7984	0.8094	0.8038
Course	310	0.8152	0.7994	0.8058
Faculty	374	0.8156	0.7823	0.7986
Student	544	0.8234	0.8167	0.8200
Overall	1396	0.8158	0.8027	0.8092

documents on the top of the retrieved results. The current approach further improved the earlier approach using the following points:

i. To improve the ranking mechanism, a novel feature selection technique (*TCFS*) is proposed which removes the noise features from the corpus.
ii. To ensure that the proposed *TCFS* technique is efficient, it is compared with the existing feature selection techniques.
iii. Next, the earlier PageRank algorithm is improved by personalizing it.
iv. The personalize PageRank and the cosine-similarity of the documents with the user-query further enhances the earlier ranking mechanism which was based on the relevance of documents and their outlink to the user query.

Experimental results on five benchmark datasets show the stability and effectiveness of the proposed query-optimized

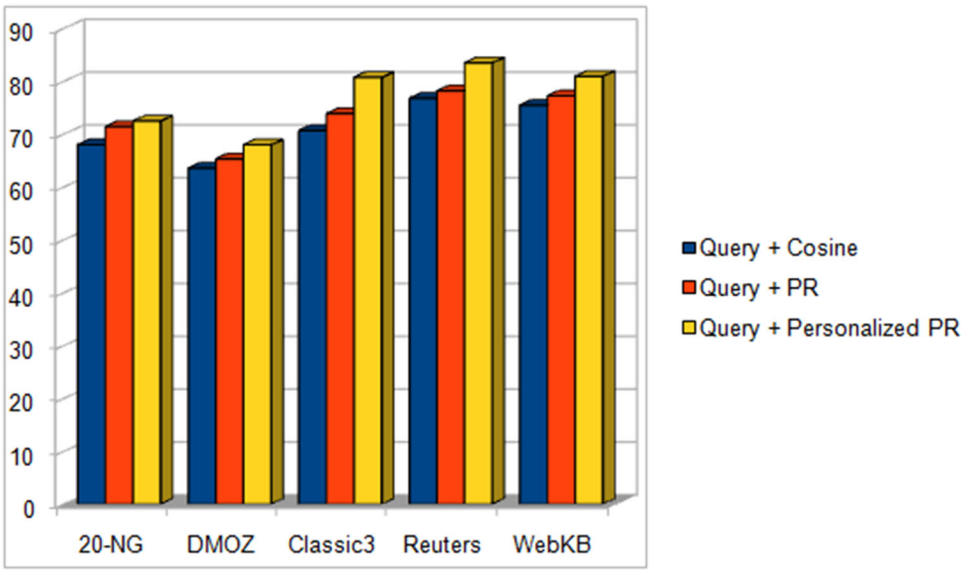

Fig. 2 F-measure (monogram query)

Fig. 3 F-measure (bi-grams query)

personalized PageRank approach. This work further can be extended by considering the following points:

i. The proposed ranking method does not take care of whether the ranking documents are spam or not. Hence, spam detection can be done before the ranking process starts.
ii. Similarly, duplicate documents are a big threat to the search engine which is not detected by the proposed method. Future work for detection of duplicate documents before the ranking process can further improve the proposed work.
iii. By considering each query as a mixture of various topics (generated from documents using latent dirichlet allocation) can further improve the PageRank matrix that receives more relevant and important outlinks.
iv. The proposed approach can be improved by incorporating user behavior signals [1].

Compliance with ethical standards

Conflict of interest The corresponding author states that there is no conflict of interest.

References

1. Agichtein, E., Brill, E., Dumais, S.: Improving web search ranking by incorporating user behavior information. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp 19–26 (2006)
2. Andersen, R., Borgs, C., Chayes, J., Hopcraft, J., Mirrokni, V.S., Teng, S.H.: Local computation of pagerank contributions. In: Algorithms and Models for the Web-Graph, Springer, pp 150–165 (2007)
3. Arun, K., Govindan, V., Kumar, S.M.: On integrating re-ranking and rank list fusion techniques for image retrieval. Int. J. Data Sci. Analytics **4**(1), 53–81 (2017)
4. Aslam, J.A., Montague, M.: Models for metasearch. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp 276–284 (2001)
5. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: the fuzzy c-means clustering algorithm. Computers Geosci. **10**(2), 191–203 (1984)
6. Bougouin, A., Boudin, F., Daille, B.: Topicrank: Graph-based topic ranking for keyphrase extraction. In: International Joint Conference on Natural Language Processing (IJCNLP), pp 543–551 (2013)
7. Breese, J.S., Heckerman, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., pp 43–52 (1998)
8. Chahal, P., Singh, M., Kumar, S.: An efficient web page ranking for semantic web. J. Inst. Eng. India Ser B **95**(1), 15–21 (2014)
9. Chen, L., Kulasiri, D., Samarasinghe, S.: A novel data-driven boolean model for genetic regulatory networks. Front. Physiol. **9**, 1328 (2018)
10. Chirita, P.A., Diederich, J., Nejdl, W.: Mailrank: Using ranking for spam detection. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, ACM, pp 373–380 (2005)
11. Collins, M.: Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp 489–496 (2002)
12. Craswell, N., Hawking, D.: Overview of the trec-2002 web track. In: TREC, pp 78–92 (2002)
13. Dali, L., Fortuna, B., Duc, TT., Mladenić, D.: Query-independent learning to rank for rdf entity search. In: Extended Semantic Web Conference, Springer, pp 484–498 (2012)
14. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th International Conference on World Wide Web, ACM, pp 519–528 (2003)
15. Derhami, V., Khodadadian, E., Ghasemzadeh, M., Bidoki, A.M.Z.: Applying reinforcement learning for web pages ranking algorithms. Appl. Soft Comput. **13**(4), 1686–1692 (2013)
16. Diaconis, P., Graham, R.L.: Spearman's footrule as a measure of disarray. J. R. Stat. Soc. Ser. B Methodological **39**, 262–268 (1977)
17. Du, Y., Hai, Y.: Semantic ranking of web pages based on formal concept analysis. J. Syst. Softw. **86**(1), 187–197 (2013)
18. Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: Collaborative filtering recommender systems. Found. Trends Human-Computer Interact. **4**(2), 81–173 (2011)
19. Fafalios, P., Kasturia, V., Nejdl, W.: Ranking archived documents for structured queries on semantic layers. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries, ACM, pp. 155–164 (2018)
20. Gao, Y., Xu, Y., Li, Y.: Pattern-based topics for document modelling in information filtering. IEEE Trans. Knowl. Data Eng. **27**(6), 1629–1642 (2015)
21. Gugnani, S., Roul, R.K.: Triple indexing: an efficient technique for fast phrase query evaluation. Int. J. Computer Appl. **87**(13), 9–13 (2014)
22. Gugnani, S., Bihany, T., Roul, R.K.: A complete survey on web document ranking. Int. J. Computer Appl. ICACEA **975**, 8887 (2014)
23. Guo, Z., Zhang, L., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. IEEE Trans. Image Process. **19**(6), 1657–1663 (2010)
24. Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., Zhao, L.: Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. Multimedia Tools Appl. **78**(11), 15169–15211 (2019)
25. Khodaei, A., Shahabi, C., Li, C.: Skif-p: a point-based indexing and ranking of web documents for spatial-keyword search. Geoinformatica **16**(3), 563–596 (2012)
26. Kwak, N., Choi, C.H.: Input feature selection by mutual information based on parzen window. IEEE Trans. Pattern Anal. Mach. Intell. **24**(12), 1667–1671 (2002)
27. Langville, A.N., Meyer, C.D.: Deeper inside pagerank. Internet Math. **1**(3), 335–380 (2004)
28. Liu, T.Y., et al.: Learning to rank for information retrieval. Found. Trends® Inf. Retr. **3**(3), 225–331 (2009)
29. Lv, Y., Zhai, C.: Positional language models for information retrieval. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp 299–306 (2009)
30. Meymandpour, R., Davis, J.G.: A semantic similarity measure for linked data: an information content-based approach. Knowl.-Based Syst. **109**, 276–293 (2016)
31. Mirzal, A.: Clustering and latent semantic indexing aspects of the singular value decomposition. Int. J. Inf. Decision Sci. **8**(1), 53–72 (2016)

32. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 115–124 (2005)
33. Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., Cheng, X.: Deeprank: a new deep architecture for relevance ranking in information retrieval. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, pp. 257–266 (2017)
34. Pasquinelli, M.: Google's pagerank algorithm: a diagram of cognitive capitalism and the rentier of the common intellect. Deep Search: The Politics of Search Beyond Google pp. 152–163 (2009)
35. Pon, R.K., Cardenas, A.F., Buttler, D., Critchlow, T.: Tracking multiple topics for finding interesting articles. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 560–569 (2007)
36. Qin, T., Liu, T.Y., Zhang, X.D., Wang, D.S., Xiong, W.Y., Li, H.: Learning to rank relational objects and its application to web search. In: Proceedings of the 17th International Conference on World Wide Web, ACM, pp. 407–416 (2008)
37. Robertson, S.E., Walker, S.: Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Springer, New York, pp. 232–241 (1994)
38. Roul, R.K.: Detecting spam web pages using multilayer extreme learning machine. Int. J. Big Data Intell. **5**(1–2), 49–61 (2018a)
39. Roul, R.K.: An effective approach for semantic-based clustering and topic-based ranking of web documents. Int. J. Data Sci. Analytics **5**(4), 269–284 (2018b)
40. Roul, R.K., Arora, K.: A nifty review to text summarization-based recommendation system for electronic products. Soft. Comput. **23**(24), 13183–13204 (2019)
41. Roul, R.K., Rai, P.: A new feature selection technique combined with elm feature space for text classification. In: Proceedings of the 13th International Conference on Natural Language Processing, pp. 285–292 (2016)
42. Roul, R.K., Sahoo, J.K.: Query-optimized pagerank: a novel approach. In: Advances in Intelligent Systems and Computing 711, Springer, pp. 673–683 (2017)
43. Roul, R.K., Sahoo, J.K.: Sentiment analysis and extractive summarization based recommendation system. In: Computational Intelligence in Data Mining, Springer, pp. 473–487 (2020)
44. Roul, R.K., Gugnani, S., Kalpeshbhai, S.M.: Clustering based feature selection using extreme learning machines for text classification. In: 2015 Annual IEEE India Conference (INDICON), IEEE, pp. 1–6 (2015)
45. Roul, R.K., Asthana, S.R., Kumar, G.: Spam web page detection using combined content and link features. Int. J. Data Min. Modell. Manag. **8**(3), 209–222 (2016a)
46. Roul, R.K., Bhalla, A., Srivastava, A.: Commonality-rarity score computation: a novel feature selection technique using extended feature space of elm for text classification. In: Proceedings of the 8th Annual Meeting of the Forum on Information Retrieval Evaluation, pp. 37–41 (2016b)
47. Roul, R.K., Asthana, S.R., Kumar, G.: Study on suitability and importance of multilayer extreme learning machine for classification of text data. Soft Comput. **21**, 4239 (2017a)
48. Roul, R.K., Sahoo, J.K., Goel, R.: Deep learning in the domain of multi-document text summarization. PReMI, LNCS **10597**, 575–581 (2017b)
49. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
50. Santos, I., Laorden, C., Sanz, B., Bringas, P.G.: Enhanced topic-based vector space model for semantics-aware spam filtering. Expert Syst. Appl. **39**(1), 437–444 (2012)
51. Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z.: A novel feature selection algorithm for text categorization. Expert Syst. Appl. **33**(1), 1–5 (2007)
52. Song, Y., Pan, S., Liu, S., Zhou, M.X., Qian, W.: Topic and keyword re-ranking for LDA-based topic modeling. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, ACM, pp. 1757–1760 (2009)
53. Sparck Jones, K.: A statistical interpretation of term specificity and its application in retrieval. J. Documentation **28**(1), 11–21 (1972)
54. Spink, A., Wolfram, D., Jansen, M.B., Saracevic, T.: Searching the web: the public and their queries. J. Am. Soc. Inform. Sci. Technol. **52**(3), 226–234 (2001)
55. Tao, T., Zhai, C.: Regularized estimation of mixture models for robust pseudo-relevance feedback. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 162–169 (2006)
56. Vuurens, J.B., de Vries, A.P.: Distance matters! cumulative proximity expansions for ranking documents. Inf. Retr. **17**(4), 380–406 (2014)
57. Wang, Y., Lu, J., Chen, J., Li, Y.: Crawling ranked deep web data sources. World Wide Web **20**(1), 89–110 (2017)
58. Xu, J., Cao, Y., Li, H., Zhao, M.: Ranking definitions with supervised learning methods. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, ACM, pp. 811–819 (2005)
59. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. ICML **97**, 412–420 (1997)
60. Yulianti, E., Chen, R.C., Scholer, F., Croft, W.B., Sanderson, M.: Ranking documents by answer-passage quality. In: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 335–344 (2018)
61. Zhai, C., Cohen, W.W., Lafferty, J.: Beyond independent relevance: methods and evaluation metrics for subtopic retrieval. In: ACM SIGIR Forum, ACM vol. 49, pp. 2–9 (2015)
62. Zhao, J., Yun, Y.: A proximity language model for information retrieval. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 291–298 (2009)