Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 6406901, 10 pages
http://dx.doi.org/10.1155/2016/6406901

Research Article

An Improved Focused Crawler: Using Web Page Classification and Link Priority Evaluation

Houqing Lu,[1] Donghui Zhan,[1] Lei Zhou,[2] and Dengchao He[3]

[1] *College of Field Engineering, The PLA University of Science and Technology, Nanjing 210007, China*
[2] *Baicheng Ordnance Test Center of China, Baicheng 137000, China*
[3] *College of Command Information System, The PLA University of Science and Technology, Nanjing 210007, China*

Correspondence should be addressed to Donghui Zhan; 416275237@qq.com

Received 13 January 2016; Accepted 27 April 2016

Academic Editor: Jean J. Loiseau

A focused crawler is topic-specific and aims selectively to collect web pages that are relevant to a given topic from the Internet. However, the performance of the current focused crawling can easily suffer the impact of the environments of web pages and multiple topic web pages. In the crawling process, a highly relevant region may be ignored owing to the low overall relevance of that page, and anchor text or link-context may misguide crawlers. In order to solve these problems, this paper proposes a new focused crawler. First, we build a web page classifier based on improved term weighting approach (ITFIDF), in order to gain highly relevant web pages. In addition, this paper introduces an evaluation approach of the link, link priority evaluation (LPE), which combines web page content block partition algorithm and the strategy of joint feature evaluation (JFE), to better judge the relevance between URLs on the web page and the given topic. The experimental results demonstrate that the classifier using ITFIDF outperforms TFIDF, and our focused crawler is superior to other focused crawlers based on breadth-first, best-first, anchor text only, link-context only, and content block partition in terms of harvest rate and target recall. In conclusion, our methods are significant and effective for focused crawler.

1. Introduction

With the rapid growth of network information, the Internet has become the greatest information base. How to get the knowledge of interest from massive information has become a hot topic in current research. But the first important task of those researches is to collect relevant information from the Internet, namely, crawling web pages. Therefore, in order to crawl web pages effectively, researchers proposed web crawlers. Web crawlers are programs that collect information from the Internet. It can be divided into general-purpose web crawlers and special-purpose web crawlers [1, 2]. General-purpose web crawlers retrieve enormous numbers of web pages in all fields from the huge Internet. To find and store these web pages, general-purpose web crawlers must have long running times and immense hard-disk space. However, special-purpose web crawlers, known as focused crawlers, yield good recall as well as good precision by restricting themselves to a limited domain [3–5]. Compared with general-purpose web crawlers, focused crawlers obviously need a smaller amount of runtime and hardware resources. Therefore, focused crawlers have become increasingly important in gathering information from web pages for finite resources and have been used in a variety of applications such as search engines, information extraction, digital libraries, and text classification.

Classifying the web pages and selecting the URLs are two most important steps of the focused crawler. Hence, the primary task of the effective focused crawler is to build a good web page classifier to filter irrelevant web pages of a given topic and guide the search. It is generally known that Term Frequency Inverse Document Frequency (TFIDF) [6, 7] is the most common approach of term weighting in text classification problem. However, TFIDF does not take into account the difference of expression ability in the different page position and the proportion of feature distribution when computing weights. Therefore, our paper presents a TFIDF-improved approach, ITFIDF, to make up for the defect of TFIDF in web page classification. According to ITFIDF, the page content is classified into four sections: headline, keywords, anchor text, and body. Then we set different weights to different sections based on their expression ability for page content. That means, the stronger expression ability of page content is, the higher weight would be obtained. In addition, ITFIDF develops a new weighting equation to improve the convergence of the algorithm by introducing the information gain of the term.

The approach of selecting the URLs has also another direct impact on the performance of focused crawling. The approach ensures that the crawler acquires more web pages that are relevant to a given topic. The URLs are selected from the unvisited list, where the URLs are ranked in descending order based on weights that are relevant to the given topic. At present, most of the weighting methods are based on link features [8, 9] that include current page, anchor text, link-context, and URL string. In particular, current page is the most frequently used link feature. For example, Chakrabarti et al. [10] suggested a new approach to topic-specific Web resource discovery and Michelangelo et al. [11] suggested focused crawling using context graphs. Motivated by this, we propose link priority evaluation (LPE) algorithm. In LPE, web pages are partitioned into some smaller content blocks by content block partition (CBP) algorithm. After partitioning the web page, we take a content block as a unit to evaluate each content block, respectively. If relevant, all unvisited URLs are extracted and added into frontier, and the relevance is treated as priority weight. Otherwise, discard all links in the content block.

The rest of this paper is organized as follows: Section 2 briefly introduces the related work. In Section 3, the approach of web page classification based on ITFIDF is proposed. Section 4 illustrates how to use LPE algorithm to extract the URLs and calculate the relevance. The whole crawling architecture is proposed in Section 5. Several relevant experiments are performed to evaluate the effectiveness of our method in Section 6. Finally, Section 7 draws a conclusion of the whole paper.

2. Related Work

Since the birth of the WWW, researchers have explored different methods of Internet information collection. Focused crawlers are commonly used instruments for information collector. The focused crawlers are affected by the method of selecting the URLs. In what follows, we briefly review some work on selecting the URLs.

Focused crawlers must calculate the priorities for unvisited links to guide themselves to retrieve web pages that are related to a given topic from the internet. The priorities for the links are affected by topical similarities of the full texts and the features (anchor texts, link-context) of those hyperlinks [12]. The formula is defined as

$$\text{Priority}(l) = \frac{1}{2} \cdot \frac{1}{n} \sum_{p}^{n} \text{Sim}\left(u_p, t\right) + \frac{1}{2} \cdot \text{Sim}\left(f_l, t\right), \quad (1)$$

where Priority(l) is the priority of the link l ($1 \leq l \leq L$) and L is the number of links. n is the number of retrieved web pages including the link l. Sim(u_p, t) is the similarity between the topic t and the full text u_p, which corresponds to web page p including the link l. Sim(f_l, t) is the similarity between the topic t and the anchor text f_l corresponding to anchor texts including the link l.

In the above formula, many variants have been proposed to improve the efficiency of predicting the priorities for links. Earlier, researchers took the topical similarities of the full texts of those links as the strategy for prioritizing links, such as Fish Search [13], Shark Search algorithm [14], and other focused crawlers including [8, 10, 15, 16]. Due to the features provided by link, the anchor texts and link-context in web pages are utilized by many researchers to search the web [17]. Eiron and McCurley [18] put forward a statistical study of the nature of anchor text and real user queries on a large corpus of corporate intranet documents. Li et al. [19] presented a focused crawler guided by anchor texts using a decision tree. Chen and Zhang [20] proposed HAWK, which is simply a combination of some well-known content-based and link-based crawling approaches. Peng and Liu [3] suggested an improved focused crawler combining full texts content and features of unvisited hyperlink. Du et al. [2] proposed an improved focused crawler based on semantic similarity vector space model. This model combines cosine similarity and semantic similarity and uses the full text and anchor text of a link as its documents.

3. Web Page Classification

The purpose of focused crawling is to achieve relevant web pages of a given topical and discard irrelevant web pages. It can be regarded as the problem of binary classification. Therefore, we will build a web page classifier by Naive Bayes, the most common algorithm used for text classification [21]. Constructing our classifier adopts three steps: first, pruning the feature space, then term weighting, and finally building the web page classifier.

3.1. Pruning the Feature Space. Web page classifier embeds the documents into some feature space, which may be extremely large, especially for very large vocabularies. And, the size of feature space affects the efficiency and effectiveness of page classifier. Therefore, pruning the feature space is necessary and significant. In this paper, we adopt the method of mutual information (MI) [22] to prune the feature space. MI is an approach of measuring information in information theory. It has been used to represent correlation of two events. That is, the greater the MI is, the more the correlation between two events is. In this paper, MI has been used to measure the relationship between feature t_j and class C_i.

Calculating MI has two steps: first, calculating MI between feature t_j in current page and each class and selecting the biggest value as the MI of feature t_j. Then, the features are ranked in descending order based on MI and maintain features which have higher value better than threshold. The formula is represented as follows:

$$\text{MI}\left(t_j, C_i\right) = \log \frac{p\left(t_j, C_i\right)}{p\left(t_j\right) p\left(C_i\right)}, \quad (2)$$

where $\mathrm{MI}(t_j, C_i)$ denote the MI between the feature t_j and the class C_i; $p(t_j)$ denote the probability that a document arbitrarily selected from the corpus contains the feature t_j; $p(C_i)$ denote the probability that a document arbitrarily selected from the corpus belongs to the class C_i; $p(t_j, C_i)$ denote the joint probability that this arbitrarily selected document belongs to the class C_i as well as containing the feature t_j at the same time.

3.2. Term Weighting. After pruning the feature space, the document d_i is represented as $d_i = (t_{i1}, \ldots, t_{ij}, \ldots, t_{im})$. Then, we need to calculate weight of terms by weighting method. In this paper, we adopt ITFIDF to calculate weight of terms. Compared with TFIDF, the improvements of the ITFIDF are as follows.

In ITFIDF, the web page is classified into four sections: headline, keywords, anchor text, and body, and we set the different weights to different sections based on their express ability for page content. The frequency of term t_j in document d_i is computed as follows:

$$tf_{ij} = \alpha \times tf_{ij1} + \beta \times tf_{ij2} + \lambda \times tf_{ij3} + \varepsilon \times tf_{ij4}, \tag{3}$$

where tf_{ij1}, tf_{ij2}, tf_{ij3}, and tf_{ij4} represent occurrence frequency of term t_j in the headline, keywords, anchor text, and content of the document d_i, respectively; α, β, λ, and ε are weight coefficients, and $\alpha > \beta > \lambda > \varepsilon \geq 1$.

Further analysis found that TFIDF method is not considering the proportion of feature distribution. We also develop a new term weighting equation by introducing the information gain of the term. The new weights calculate formula as follows:

$$w_{ij} = \frac{tf_{ij} \times idf_j \times \mathrm{IG}_j}{\sqrt{\sum_{m=1}^{M} (tf_{im} \times idf_m)^2}}, \tag{4}$$

where w_{ij} is the weight of term t_j in document d_i; tf_{ij} and idf_j are, respectively, the term frequency and inverse document frequency of term t_j in document d_i; M is the total number of documents in sets; IG_j is the information gain of term t_j and might be obtained by

$$\mathrm{IG}_j = H(D) - H(D \mid t_j). \tag{5}$$

$H(D)$ is the information entropy of document set D and could be obtained by

$$H(D) = -\sum_{d_i \in D} (p(d_i) \times \log_2 p(d_i)). \tag{6}$$

$H(D \mid t_j)$ is the conditional entropy of term t_j and could be obtained by

$$H(D \mid t_j) = -\sum_{d_i \in D} (p(d_i \mid t_j) \times \log_2 p(d_i \mid t_j)). \tag{7}$$

$p(d_i)$ is the probability of document d_i. In this paper, we compute $p(d_i)$ based on [23], and the formula is defined as

$$p(d_i) = \frac{|\mathrm{wordset}(d_i)|}{\sum_{d_k \in D} |\mathrm{wordset}(d_k)|}, \tag{8}$$

where $|\mathrm{wordset}(d_i)|$ refers to the sum of feature frequencies of all the terms in the document d_i.

3.3. Building Web Page Classifier. After pruning feature space and term weighting, we build the web page classifier by the Naïve Bayesian algorithm. In order to reduce the complexity of the calculation, we fail to consider the relevance and order between terms in web page. Assume that N is the number of web pages in set D; N_i is the number of web pages in the class C_i. According to Bayes theorem, the probability of web page d_j that belongs to class C_i is represented as follows:

$$P(C_i \mid d_j) \propto \frac{1}{P(d_j)} P(C_i) \prod_{t_k \in d_j} P(t_k \mid C_i), \tag{9}$$

where $P(C_i) = N_i/N$ and the value is constant; $P(d_j)$ is constant too; t_k is the term of web page for the document d_j; and d_j can be represented as eigenvector of t_k, that is, $(t_1, t_2, \ldots, t_k, \ldots, t_n)$. Therefore, $P(C_i \mid d_j)$ is mostly impacted by $P(t_k \mid C_i)$. According to independence assumption above, $P(t_k \mid C_i)$ are computed as follows:

$$P(t_k \mid C_i) = \frac{1 + \sum_{d_s \in C_i} N(t_k, d_s)}{|V| + \sum_{t_p \in V} \sum_{d_s \in C_i} N(t_p, d_s)}, \tag{10}$$

where $N(t_k, d_s)$ is the number of terms t_k in the document d_s; V is vocabulary of class C_i.

4. Link Priority Evaluation

In many irrelevant web pages, there may be some regions that are relevant to a given topic. Therefore, in order to more fully select the URLs that are relevant to the given topic, we propose the algorithm of link priority evaluation (LPE). In LPE algorithm, web pages are partitioned into some smaller content blocks by content block partition (CBP) [3, 24, 25]. After partitioning the web page, we take a content block as a unit of relevance calculating to evaluate each content block, respectively. A highly relevant region in a low overall relevance web page will not be obscured, but the method omits the links in the irrelevant content blocks, in which there may be some anchors linking the relevant web pages. Hence, in order to solve this problem, we develop the strategy of JFE, which is the relevance evaluate method between link and the content block. If a content block is relevant, all unvisited URLs are extracted and added into frontier, and the content block relevance is treated as priority weight. Otherwise, LPE will adopt JFE to evaluate the links in the block.

4.1. JFE Strategy. Researchers often adopt anchor text or link-context feature to calculate relevance between the link and topic, in order to achieve the goal of extracting relevant links from irrelevant content block. However, some web page designers do not summarize the destination web pages in the anchor text. Instead, they use words such as "Click here," "here," "Read more," "more," and "next" to describe the texts around them in anchor text. If we calculate relevance between anchor text and topic, we may omit some destination link. Similarly, if we calculate relevance between link-context and topic, we may also omit some links or extract some irrelevant links. In view of this, we propose JFE strategy to reduce

```
Input: current web page, eigenvector $\mathbf{v}$ of a given topic, threshold
Output: url_queue
procedure LPE
block_list ← CBP(web page)
for each block in block_list
  extract features from block and compute weights, and generate eigenvector $\mathbf{u}$ of block
  $\mathrm{Sim}_1 \leftarrow \mathbf{Sim}_{\mathbf{CBP}}(u, v)$
  if $\mathrm{Sim}_1$ > threshold then
    link_list ← extract each link of block
      for each link in link_list
        Priority(link) ← $\mathrm{Sim}_1$
        enqueue its unvisited urls into url_queue based on priorities
      end for
  else
      temp_queue ← extract all anchor texts and link_contexts
      for each link in temp_queue
        extract features from anchor text and compute weights, and generate eigenvector $\mathbf{u_1}$ of anchor text
        extract features from link_contexts and compute weights, and generate eigenvector $\mathbf{u_2}$ of link_contexts text
        $\mathrm{Sim}_2 \leftarrow \mathbf{Sim}_{\mathbf{JFE}}(u, v)$
        if $\mathrm{Sim}_2$ > threshold then
          Priority(link) ← $\mathrm{Sim}_2$
          enqueue its unvisited urls into url_queue based on priorities
        end if
        dequeue url in temp_queue
      end for
  end if
end for
end procedure
```

Algorithm 1: Link priority evaluation.

abovementioned omission and improve the performance of the focused crawlers. JFE combine the features of anchor text and link-context. The formula is shown as follows:

$$\mathrm{Sim}_{\mathrm{JFE}}(u, v) = \lambda \times \mathrm{Sim}_{\mathrm{anchor}}(u, v) + (1 - \lambda) \times \mathrm{Sim}_{\mathrm{context}}(u, v), \tag{11}$$

where $\mathrm{Sim}_{\mathrm{JFE}}(u, v)$ is the similarity between the link u and topic v; $\mathrm{Sim}_{\mathrm{anchor}}(u, v)$ is the similarity between the link u and topic v when only adopting anchor text feature to calculate relevance; $\mathrm{Sim}_{\mathrm{context}}(u, v)$ is the similarity between the link u and topic v when only adopting link-context feature to calculate relevance; λ $(0 < \lambda < 1)$ is an impact factor, which is used to adjust weighting between $\mathrm{Sim}_{\mathrm{anchor}}(u, v)$ and $\mathrm{Sim}_{\mathrm{context}}(u, v)$. If $\lambda > 0.5$, then the anchor text is more important than link-context feature in the JFE strategy; if $\lambda < 0.5$, then the link-context feature is more important than anchor text in the JFE strategy; if $\lambda = 0.5$, then the anchor text and link-context feature are equally important in the JFE strategy. In this paper, λ is assigned a constant 0.5.

4.2. LPE Algorithm. LPE is uses to calculate similarity between links of current web page and a given topic. It can be described specifically as follows. First, the current web page is partitioned into many content blocks based on CBP. Then, we compute the relevance of content blocks with the topic using the method of similarity measure. If a content block is relevant, all unvisited URLs are extracted and added into frontier, and the content block similarity is treated as priority, if the content block is not relevant, in which JFE is used to calculate the similarity, and the similarity is treated as priority weight. Algorithm 1 describes the process of LPE.

LPE compute the weight of each term based on TFC weighting scheme [26] after preprocessing. The TFC weighting equation is as follows:

$$w_{t,u} = \frac{f_{t,u} \times \log(N/n_t)}{\sqrt{\sum_{r=1}^{M}\left[f_{r,u} \times \log(N/n_t)\right]^2}}, \tag{12}$$

where $f_{t,u}$ is the frequency of term t in the unit u (content block, anchor text, or link-context); N is the number of feature units in the collection; M is the number of all the terms; n_t is the number of units where word t occurs.

Then, we are use the method of cosine measure to compute the similarity between link feature and topic. The formula is shown as follows:

$$\mathrm{Sim}(u, v) = \frac{\mathbf{u} \cdot \mathbf{v}}{|\mathbf{u}| \times |\mathbf{v}|} = \frac{\sum_{t=1}^{n} w_{t,u} \times w_{t,v}}{\sqrt{\sum_{t=1}^{n} w_{t,u}^2} \times \sqrt{\sum_{t=1}^{n} w_{t,v}^2}}, \tag{13}$$

where $\mathbf{u}$ is eigenvector of a unit, that is, $\mathbf{u} = \{w_{1,u}, w_{2,u}, \ldots, w_{n,u}\}$; $\mathbf{v}$ is eigenvector of a given topic, that is, $\mathbf{v} = \{w_{1,v}, w_{2,v}, \ldots, w_{n,v}\}$; $w_{t,u}$ and $w_{t,v}$ are the weight of $\mathbf{u}$ and $\mathbf{v}$, respectively. Hence, when $\mathbf{u}$ is eigenvector of the content block, we can use the above formula to compute $\mathrm{Sim}_{\mathrm{CBP}}(u, v)$. In the same way,

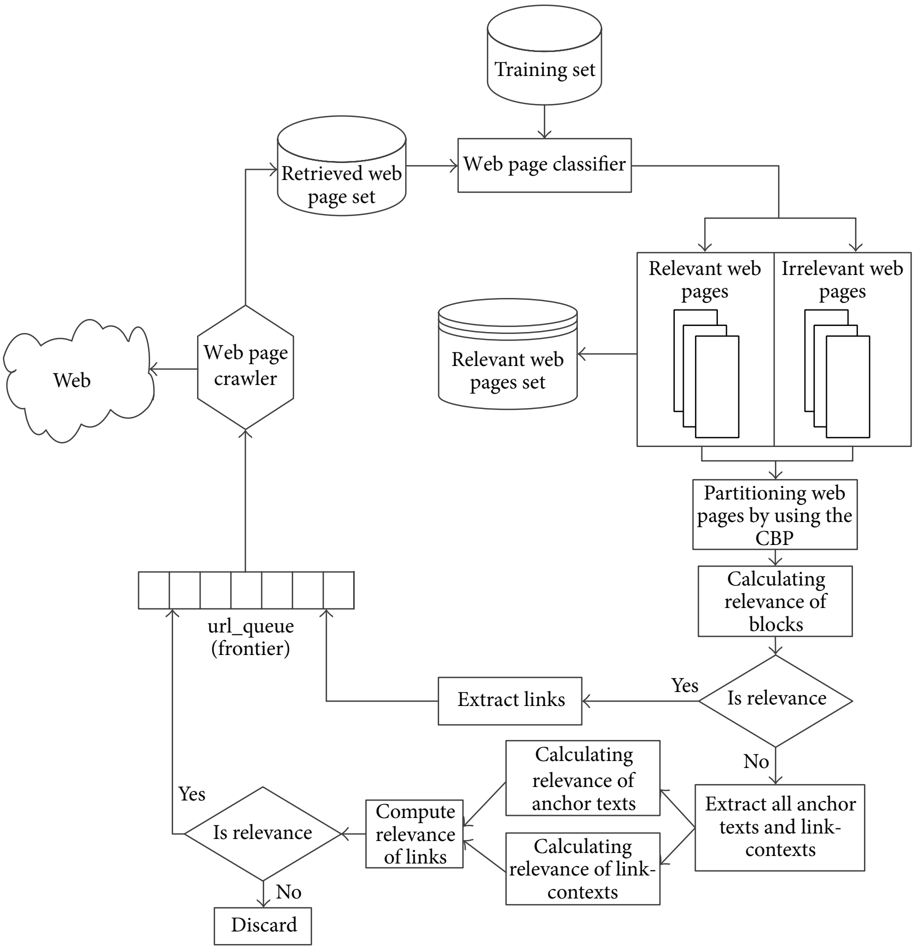

Training set
Retrieved web page set
Web page classifier
Relevant web pages
Irrelevant web pages
Relevant web pages set
Web
Web page crawler
Partitioning web pages by using the CBP
Calculating relevance of blocks
url_queue (frontier)
Yes
Is relevance
Extract links
No
Calculating relevance of anchor texts
Extract all anchor texts and link-contexts
Yes
Is relevance
Compute relevance of links
Calculating relevance of link-contexts
No
Discard

Figure 1: The architecture of the improved focused crawler.

we can use the above formula to compute $\mathrm{Sim}_{\mathrm{anchor}}(u, v)$ and $\mathrm{Sim}_{\mathrm{context}}(u, v)$ too.

5. Improved Focused Crawler

In this section, we provide the architecture of focused crawler enhanced by web page classification and link priority evaluation. Figure 1 shows the architecture of our focused crawler. The architecture for our focused crawler is divided into several steps as follows:

(1) The crawler component dequeues a URL from the url_queue (frontier), which is a priority queue. Initially, the seed URLs are inserted into the url_queue with the highest priority score. Afterwards, the items are dequeued on a highest-priority-first basis.

(2) The crawler locates the web pages pointed and attempts to download the actual HTML data of the web page by the current fetched URL.

(3) For each downloaded web page, the crawler adopts web page classifier to classify. The relevant web pages are added into relevant web page set.

(4) Then, web pages are parsed into the page's DOM tree and partitioned into many content blocks according to HTML content block tags based on CBP algorithm. And calculating the relevance between each content block and topic is by using the method of similarity measure. If a content block is relevant, all unvisited URLs are extracted and added into frontier, and the content block relevance is treated as priority weight.

(5) If the content block is not relevant, we need to extract all anchors and link-contexts and adopt the JFE strategy to get each link's relevance. If relevant, the link is also added into frontier, and the relevance is treated as priority weight; otherwise give up web page.

(6) The focused crawler continuously downloads web pages for given topic until the frontier becomes empty or the number of the relevant web pages reaches a default.

6. Experimental Results and Discussion

In order to verify the effectiveness of the proposed focused crawler, several tests have been achieved in this paper. The tests are Java applications running on a Quad Core Processor

2.4 GHz Core i7 PC with 8 G of RAM and SATA disk. The experiments include two parts: evaluate the performance of web page classifier and evaluate the performance of focused crawler.

6.1. Evaluate the Performance of Web Page Classifier

6.1.1. Experimental Datasets. In this experiment, we used the Reuters-21,578 (evaluate the performance), Reuters Corpus Volume 1 (RCV1) (http://trec.nist.gov/data/reuters/reuters.html), 20 Newsgroups (http://qwone.com/~jason/20Newsgroups/), and Open Directory Project (http://www.droz.org/) as our training and test dataset. Of the 135 topics in Reuters-21,578, 5480 documents from 10 topics are used in this paper. RCV1 has about 810,000 Reuters, English language news stories collected from the Reuters newswire. We use "topic codes" set, which include four hierarchical groups: CCAT, ECAT, GCAT, and MCAT. Among 789,670 documents, 5,000 documents are used in this paper. The 20 Newsgroups dataset has about 20,000 newsgroup documents collected by Ken Lang. Of the 20 different newsgroups in dataset, 8540 documents from 10 newsgroups are used in this paper. ODP is the largest, most comprehensive human-edit directory of the Web. The data structure of ODP is organized as a tree, where nodes contain URLs that link to the specific topical web pages; thus we use the first three layers and consider both hyperlink text and the corresponding description. We choose ten topics as samples to test the performance of our method, and 500 samples are chosen from each topic.

6.1.2. Performance Metrics. The performance of web page classifier can reflect the availability of the focused crawler directly. Hence, it is essential to evaluate the performance of web page classifier. Most classification tasks are evaluated using Precision, Recall, and F-Measure. Precision for text classifying is the fraction of documents assigned that are relevant to the class, which measures how well it is doing at rejecting irrelevant documents. Recall is the proportion of relevant documents assigned by classifier, which measures how well it is doing at finding all the relevant documents. We assume that T is the set of relevant web pages in test dataset; U is the set of relevant web pages assigned by classifier. Therefore, we define Precision [3, 27] and Recall [3, 27] as follows:

$$\begin{aligned}\text{Precision} &= \frac{|U \cap T|}{|U|} \times 100\%,\\ \text{Recall} &= \frac{|U \cap T|}{|T|} \times 100\%.\end{aligned} \tag{14}$$

The Recall and Precision play very important role in the performance evaluation of classifier. However, they have certain defects; for example, when improving one performance value, the other performance value will decline [27]. For mediating the relationship between Recall and Precision, Lewis [28, 29] proposes F-Measure that is used to evaluate the performance of classifier. Here, F-Measure is also used to measure the performance of our web page classifier in this paper. F-Measure is defined as follows:

$$F\text{-Measure} = \frac{\left(\beta^2 + 1\right) \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \tag{15}$$

where β is a weight for reflecting the relative importance of Precision and Recall. Obviously, if $\beta > 1$, then Recall is more important than Precision; if $0 < \beta < 1$, then Precision is more important than Recall; if $\beta = 1$, then Recall and Precision are equally important. In this paper, β is assigned a constant 1.

6.1.3. Evaluate the Performance of Web Page Classifier. In order to test the performance of ITFIDF, we run the classifier using different term weighting methods. For a fair comparison, we use the same method of pruning the feature space and classification model in the experiment. Figure 2 compares the performance of F-Measure achieved by our classifying method using ITFIDF and TFIDF weighting for each topic on the four datasets.

As can be seen from Figure 2, we observe that the performance of classification method using ITFIDF weighting is better than TFIDF on each dataset. In Figure 2, the average of the ITFIDF's F-Measure has exceeded the TFIDF's 5.3, 2.0, 5.6, and 1.1 percent, respectively. Experimental results show that our classification method is effective in solving classifying problems, and proposed ITFIDF term weighting is significant and effective for web page classification.

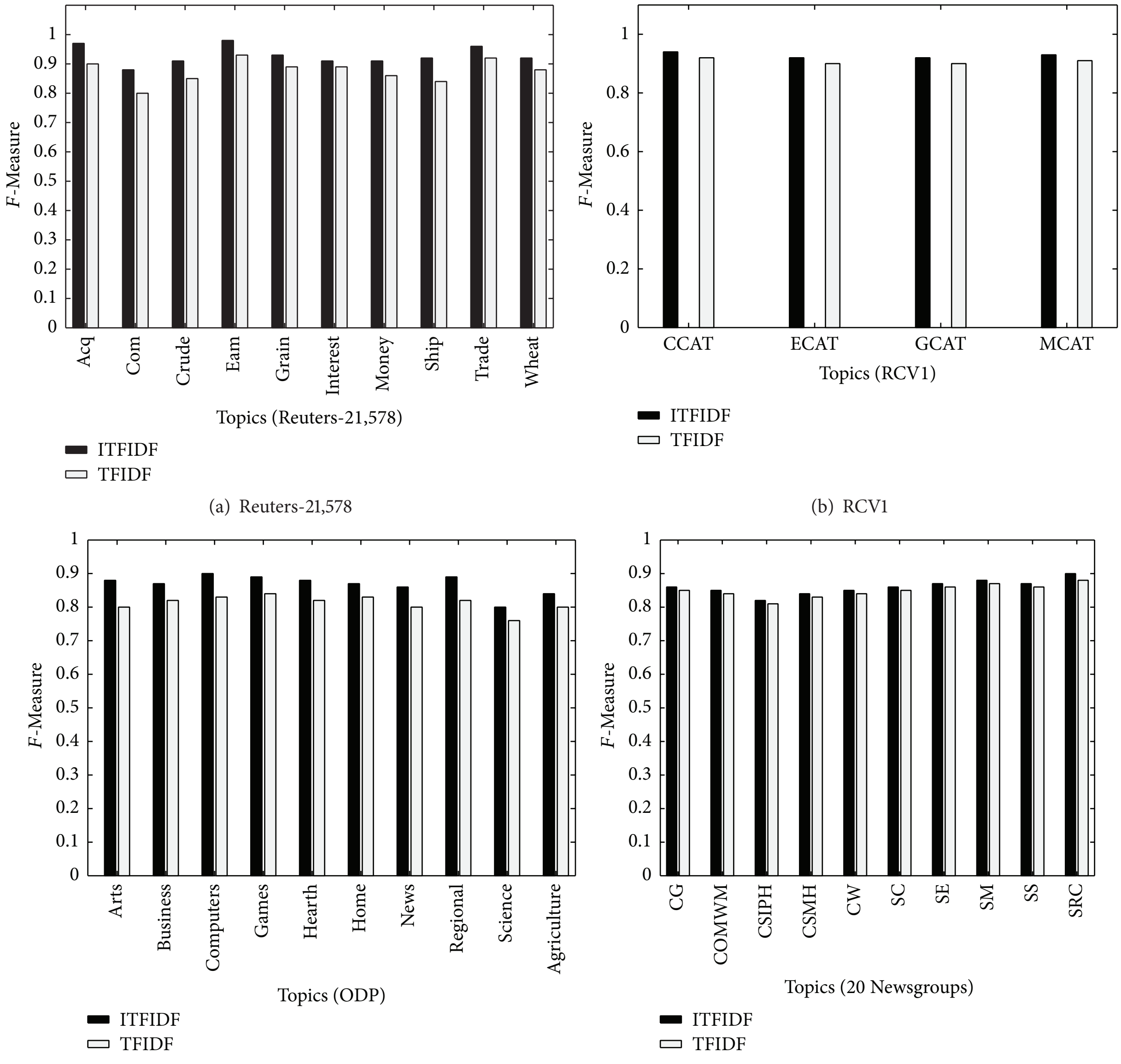
6.2. Evaluate the Performance of Focused Crawler

6.2.1. Experimental Data. In this experiment, we selected the relevant web pages and the seed URLs for the above 10 topics as input data of our crawler. These main topics are basketball, military, football, big data, glasses, web games, cloud computing, digital camera, mobile phone, and robot. The relevant web pages for each topic accurately describe the corresponding topic. In this experiment, the relevant web pages for all of the topics were selected by us, and the number of those web pages for each topic was set to 30. At the same time, we used the artificial way to select the seed URLs for each topic. And, the seed URLs were shown in Table 1 for each topic.

6.2.2. Performance Metrics. The performance of focused crawler can also reflect the availability of the crawling directly. Perhaps the most crucial evaluation of focused crawler is to measure the rate at which relevant web pages are acquired and how effectively irrelevant web pages are filtered out from the crawler. With this knowledge, we could estimate the precision and recall of focused crawler after crawling n web pages. The precision would be the fraction of pages crawled that are relevant to the topic and recall would be the fraction of relevant pages crawled. However, the relevant set for any given topic is unknown in the web, so the true recall is hard to measure. Therefore, we adopt harvest rate and target recall to evaluate the performance of our focused crawler. And, harvest rate and target recall were defined as follows:

(1) The harvest rate [30, 31] is the fraction of web pages crawled that are relevant to the given topic, which measures how well it is doing at rejecting irrelevant web pages. The expression is given by

$$\text{Harvest rate} = \frac{\sum_{i \in V} r_i}{|V|}, \tag{16}$$



(a) Reuters-21,578

(b) RCV1

(c) ODP

(d) 20 Newsgroups

Figure 2: The comparison of F-Measures achieved by our classification method using ITFIDF and TFIDF term weighting for each dataset.

where V is the number of web pages crawled by focused crawler in current; r_i is the relevance between web page i and the given topic, and the value of r_i can only be 0 or 1. If relevant, then $r_i = 1$; otherwise $r_i = 0$.

(2) The target recall [30, 31] is the fraction of relevant pages crawled, which measures how well it is doing at finding all the relevant web pages. However, the relevant set for any given topic is unknown in the Web, so the true target recall is hard to measure. In view of this situation, we delineate a specific network, which is regarded as a virtual WWW in the experiment. Given a set of seed URLs and a certain depth, the range reached by a crawler using breadth-first crawling strategy is the virtual Web. We assume that the target set T is the relevant set in the virtual Web; C_t is the set of first t pages crawled. The expression is given by

$$\text{Target recall} = \frac{|T \cap C_t|}{|T|}. \qquad (17)$$

6.2.3. Evaluation the Performance of Focused Crawler. An experiment was designed to indicate that the proposed method of web page classification and the algorithm of LPE can improve the performance of focused crawlers. In this experiment, we built crawlers that used different techniques (breadth-first, best-first, anchor text only, link-context only, and CBP), which are described in the following, to crawl the web pages. Different web page content block partition methods have different impacts on focused web page crawling

Table 1: The seed URLs for 10 different topics.

Topic	Seed URLs
(1) Basketball	https://en.wikipedia.org/wiki/Basketball http://espn.go.com/mens-college-basketball/ http://english.sina.com/news/sports/basketball.html
(2) Military	https://en.wikipedia.org/wiki/Military http://eng.chinamil.com.cn/ http://www.foxnews.com/category/us/military.html
(3) Football	https://en.wikipedia.org/wiki/Football http://espn.go.com/college-football/ http://collegefootball.ap.org/
(4) Big data	https://en.wikipedia.org/wiki/Big_data http://bigdatauniversity.com/ http://www.ibm.com/big-data/us/en/
(5) Glasses	https://en.wikipedia.org/wiki/Glasses http://www.glasses.com/ http://www.visionexpress.com/glasses/
(6) Web games	http://www.games.com/ http://games.msn.com/ http://www.manywebgames.com/
(7) Cloud computing	https://de.wikipedia.org/wiki/Cloud_Computing https://www.oracle.com/cloud/index.html http://www.informationweek.com/
(8) Digital camera	https://en.wikipedia.org/wiki/Digital_camera http://digitalcameraworld.net/ http://www.gizmag.com/digital-cameras/
(9) Mobile phone	https://en.wikipedia.org/wiki/Mobile_phone http://www.carphonewarehouse.com/mobiles http://www.mobilephones.com/
(10) Robot	https://en.wikipedia.org/wiki/Robot http://www.botmag.com/ http://www.merriam-webster.com/dictionary/robot

performance. According to the experimental result in [25], alpha in CBP algorithm is assigned a constant 0.5 in this paper. Threshold in LPE algorithm is a very important parameter. Experiment shows that if the value of threshold is too big, focused crawler finds it hard to collect web page. Conversely, if the value of threshold is too small, the average harvest rate for focused crawler is low. Therefore, according to the actual situations, threshold is assigned a constant 0.5 in the rest of the experiments. In order to reflect the comprehensiveness of our method, Figures 3 and 4 show the average harvest rate and average target recall on ten topics for each crawling strategy, respectively.

Figure 3 shows a performance comparison of the average harvest rates for six crawling methods for ten different topics. In Figure 3, x-axis represents the number of crawled web pages; y-axis represents the average harvest rates when the number of crawled pages is N. As can be seen from Figure 3, as the number of crawled web pages increases, the average harvest rates of six crawling methods are falling. This occurs because the number of crawled web pages and the number of relevant web pages have different increasing extent, and the increment of the former was bigger than that of the latter. From Figure 3, we can also see that the numbers of crawled web pages of the LPE crawler is higher than those of the other five crawlers. In addition, the harvest rates of breadth-first crawler, best-first crawler, anchor text only crawler, link-context only crawler, CBP crawler, and LPE crawler are, respectively, 0.16, 0.28, 0.39, 0.48, 0.61, and 0.80, at the point that corresponds to 10000 crawled web pages in Figure 3. These values indicate that the harvest rate of the LPE crawler is 5.0, 2.9, 2.0, 1.7, and 1.3 times as large as those of breadth-first crawler, best-first crawler, anchor text only crawler, link-context only crawler, and CBP crawler, respectively. Therefore, the figure indicates that the LPE crawler has the ability to collect more topical web pages than the other five crawlers.

Figure 4 shows a performance comparison of the average-target recallfor six crawling methods for ten different topics. In Figure 4, x-axis represents the number of crawled web pages; y-axis represents the average target recall when the number of crawled pages is N. As can be seen from Figure 4, as the number of crawled web pages increases, the average target recall of six crawling methods is rising. This occurs

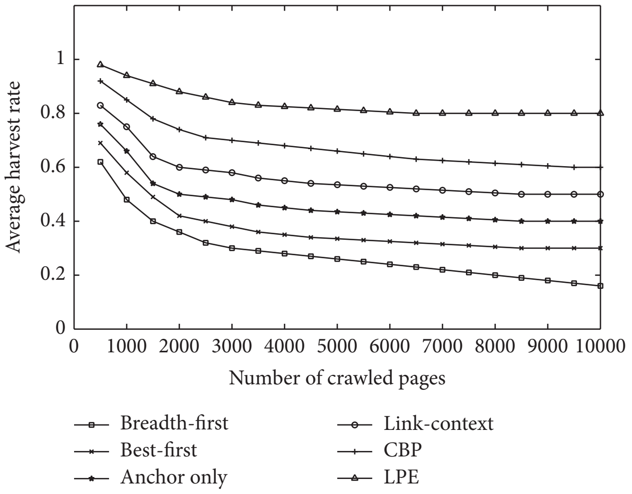


Figure 3: A performance comparison of the average harvest rates for six crawling methods for ten different topics.

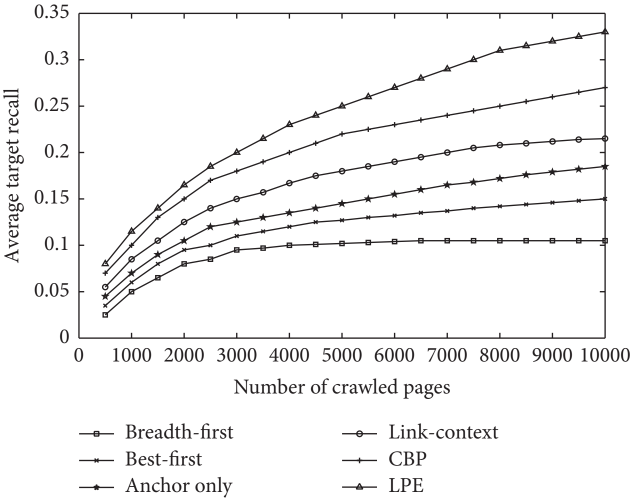


Figure 4: A performance comparison of the average target recalls for six crawling methods for ten different topics.

because the number of crawled web pages is increasing, but the target set is unchanged. The average target recall of the LPE crawler is higher than the other five crawlers for the numbers of crawled web pages. In addition, the harvest rates of breadth-first crawler, best-first crawler, anchor text only crawler, link-context only crawler, CBP crawler, and LPE crawler are, respectively, 0.10, 0.15, 0.19, 0.21, 0.27, and 0.33, at the point that corresponds to 10000 crawled web pages in Figure 4. These values indicate that the harvest rate of the LPE Crawler is 3.3, 2.2, 1.7, 0.16, and 1.2 times as large as those of breadth-first crawler, best-first crawler, anchor text only crawler, link-context only crawler, and CBP crawler, respectively. Therefore, the figure indicates that the LPE crawler has the ability to collect greater qualities of topical web pages than the other five crawlers.

It can be concluded that the LPE crawler has a higher performance than the other five focused crawlers. For the 10 topics, the LPE crawler has the ability to crawl greater quantities of topical web pages than the other five crawlers. In addition, the LPE crawler has the ability to predict more accurate topical priorities of links than other crawlers. In short, the LPE, by CBP algorithm and JFE strategy, improves the performance of the focused crawlers.

7. Conclusions

In this paper, we presented a novel focused crawler which increases the collection performance by using the web page classifier and the link priority evaluation algorithm. The approaches proposed and the experimental results draw the following conclusions.

TFIDF does not take into account the difference of expression ability in the different page position and the proportion of feature distribution when building web pages classifier. Therefore, ITFIDF can be considered to make up for the defect of TFIDF in web page classification. The performance of classifier using ITFIDF is compared with classifier using TFIDF in four datasets. Results show that the ITFIDF classifier outperforms TFIDF for each dataset. In addition, in order to gain better selection of the relevant URLs, we propose link priority evaluation algorithm. The algorithm was classified into two stages. First, the web pages were partitioned into smaller blocks by the CBP algorithm. Second, we calculated the relevance between links of blocks and the given topic using LPE algorithm. The comparison between LPE crawler and other crawlers uses 10 topics, whereas it is superior to other techniques in terms of average harvest rate and target recall. In conclusion, web page classifier and LPE algorithm are significant and effective for focused crawlers.

Competing Interests

The authors declare that they have no competing interests.

References

[1] P. Bedi, A. Thukral, and H. Banati, “Focused crawling of tagged web resources using ontology,” *Computers and Electrical Engineering*, vol. 39, no. 2, pp. 613–628, 2013.

[2] Y. Du, W. Liu, X. Lv, and G. Peng, “An improved focused crawler based on semantic similarity vector space model,” *Applied Soft Computing Journal*, vol. 36, pp. 392–407, 2015.

[3] T. Peng and L. Liu, “Focused crawling enhanced by CBP-SLC,” *Knowledge-Based Systems*, vol. 51, pp. 15–26, 2013.

[4] Y. J. Du and Z. B. Dong, “Focused web crawling strategy based on concept context graph,” *Journal of Computer Information Systems*, vol. 5, no. 3, pp. 1097–1105, 2009.

[5] F. Ahmadi-Abkenari and A. Selamat, “An architecture for a focused trend parallel Web crawler with the application of clickstream analysis,” *Information Sciences*, vol. 184, no. 1, pp. 266–281, 2012.

[6] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval,” *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[7] K. Sparck Jones, “IDF term weighting and IR research lessons,” *Journal of Documentation*, vol. 60, no. 5, pp. 521–523, 2004.

[8] S. Chakrabarti, K. Punera, and M. Subramanyam, "Accelerated focused crawling through online relevance feedback," in *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pp. 148–159, Honolulu, Hawaii, USA, May 2002.

[9] D. Davison Brian, "Topical locality in the web," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval(SIGIR '00)*, pp. 272–279, Athens, Greece, 2000.

[10] S. Chakrabarti, M. van den Berg, and B. Dom, "Focused crawling: a new approach to topic-specific Web resource discovery," *Computer Networks*, vol. 31, no. 11–16, pp. 1623–1640, 1999.

[11] D. Michelangelo, C. Frans, L. Steve, and G. C. Lee, "Focused crawling using context graphs," in *Proceedings of the 26th International Conference on Very Large Data Bases (VLDB '00)*, pp. 527–534, Cairo, Egypt, September 2000.

[12] A. Patel and N. Schmidt, "Application of structured document parsing to focused web crawling," *Computer Standards and Interfaces*, vol. 33, no. 3, pp. 325–331, 2011.

[13] P. Bra and R. Post, "Searching for arbitrary information in the WWW: the fish-search for Mosac," in *Proceedings of the 2nd International Conference on Word Wide Web (WWW '02)*, Chicago, Ill, USA, 1994.

[14] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, "The shark-search algorithm. An application: tailored web site mapping," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 317–326, 1998.

[15] F. Menczer and R. K. Belew, "Adaptive retrieval agents: internalizing local context and scaling up to the Web," *Machine Learning*, vol. 39, no. 2, pp. 203–242, 2000.

[16] G. Pant and F. Menczer, "Topical crawling for business intelligence," in *Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL '03)*, Trondheim, Norway, August 2003.

[17] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. W. Flake, "Using web structure for classifying and describing web pages," in *Proceedings of the 11th International Conference on World Wide Web (WWW '02)*, pp. 562–569, May 2002.

[18] N. Eiron and K. S. McCurley, "Analysis of anchor text for web search," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGI '03)*, pp. 459–460, Toronto, Canada, August 2003.

[19] J. Li, K. Furuse, and K. Yamaguchi, "Focused crawling by exploiting anchor text using decision tree," in *Proceedings of the 14th International World Wide Web Conference (WWW '05)*, pp. 1190–1191, Chiba, Japan, May 2005.

[20] X. Chen and X. Zhang, "HAWK: a focused crawler with content and link analysis," in *Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE '08)*, pp. 677–680, October 2008.

[21] J. Pi, X. Shao, and Y. Xiao, "Research on focused crawler based on Naïve bayes algorithm," *Computer and Digital Engineering*, vol. 40, no. 6, pp. 76–79, 2012.

[22] R. M. Fano, *Transmission of Information: A Statistical Theory of Communications*, MIT Press, Cambridge, Mass, USA, 1961.

[23] J. B. Zhu and T. S. Yao, "FIFA: a simple and effective approach to text topic automatic identification," in *Proceedings of the International Conference on Multilingual Information Processing*, pp. 207–215, Shenyang, China, 2002.

[24] N. Luo, W. Zuo, F. Yuan, and C. Zhang, "A New method for focused crawler cross tunnel," in *Rough Sets and Knowledge Technology: First International Conference, RSKT 2006, Chongquing, China, July 24–26, 2006. Proceedings*, vol. 4062 of *Lecture Notes in Computer Science*, pp. 632–637, Springer, Berlin, Germany, 2006.

[25] T. Peng, C. Zhang, and W. Zuo, "Tunneling enhanced by web page content block partition for focused crawling," *Concurrency Computation Practice and Experience*, vol. 20, no. 1, pp. 61–74, 2008.

[26] G. Salton and C. Buckley, "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

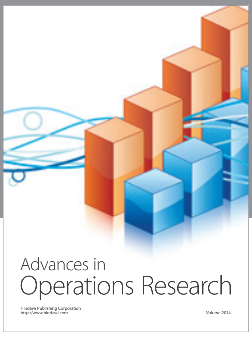
[27] L. He, *Research on some problems in text classification [Ph.D. thesis]*, Jilin University, Changchun, China, 2009.

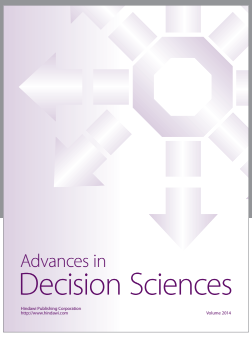
[28] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12, Springer, Dublin, Ireland, 1994.

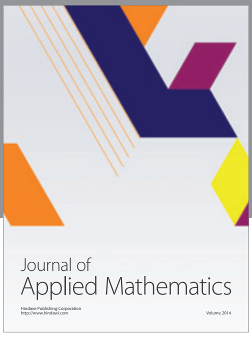
[29] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '95)*, pp. 246–254, ACM, Seattle, Wash, USA, July 1995.

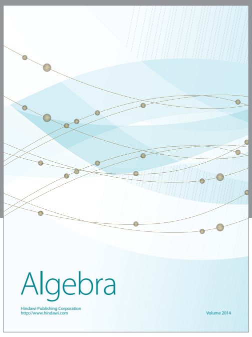
[30] G. Pant and F. Menczer, "Topical crawling for business intelligence," in *Research and Advanced Technology for Digital Libraries: 7th European Conference, ECDL 2003 Trondheim, Norway, August 17–22, 2003 Proceedings*, vol. 2769 of *Lecture Notes in Computer Science*, pp. 233–244, Springer, Berlin, Germany, 2003.

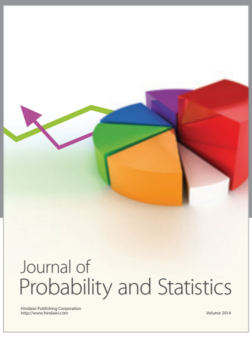
[31] G. Pant and P. Srinivasan, "Link contexts in classifier-guided topical crawlers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 107–122, 2006.


Advances in
Operations Research

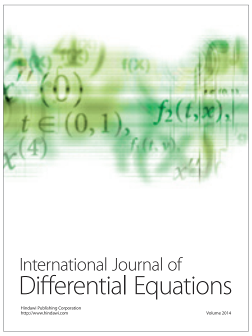

Advances in
Decision Sciences


Journal of
Applied Mathematics

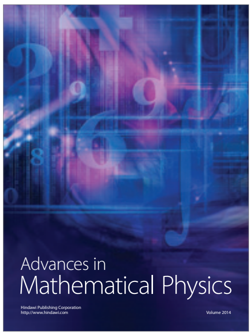

Algebra

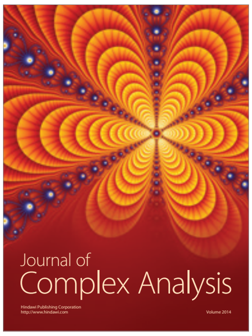

Journal of
Probability and Statistics

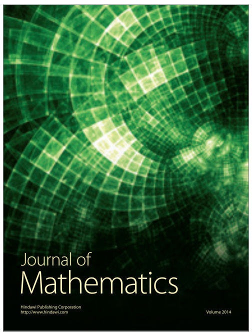
The Scientific
World Journal

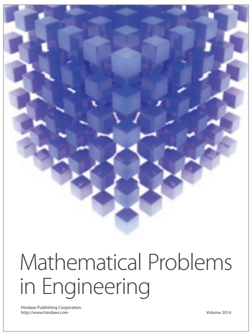

International Journal of
Differential Equations

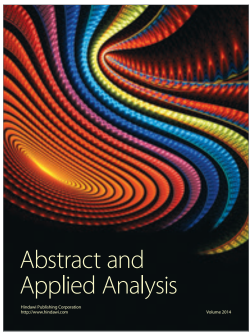
International Journal of
Combinatorics

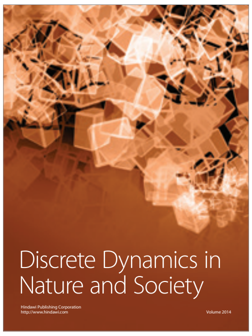

Advances in
Mathematical Physics

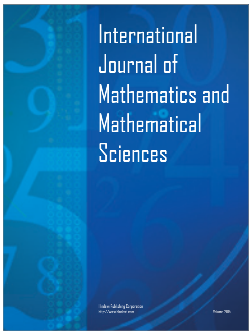

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

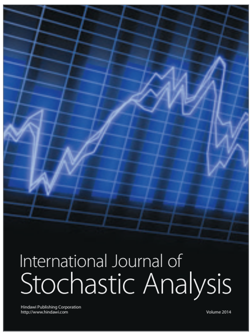

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization