Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A focused crawling strategy based on comprehensive priority evaluation of hyperlinks and improved Bayesian classifier*

Jingfa LIU[1], Yongchuang WU[‡1], Zhaoxia LIU[2]

[1]School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China

[2]Network and Information Center, Guangdong University of Foreign Studies, Guangzhou 510006, China

E-mail: jfliu@gdufs.edu.cn; wu112002@outlook.com; 554822022@qq.com

Received Oct. 22, 2024; Revision accepted Sept. 28, 2025; Crosschecked Nov. 13, 2025; Published online Dec. 12, 2025

Abstract: Avoidance of topic drift and enabling crossing tunnels are two main difficulties in focused crawling. To overcome the problem of topic drift, we design a comprehensive priority evaluation (CPE) method based on the web text, anchor text, and context of hyperlinks, which improves the topic-relevance evaluation of unvisited hyperlinks. Subsequently, we propose an improved Bayesian classifier with weights (BCW), which adds label weights to the feature words of the Bayesian classifier to enhance the accuracy of webpage classification. To cross tunnels through which some topic-relevant webpages can be reached from low-relevance webpages, we construct a content block segmentation (CBS) technology for webpages based on the backtracking method, which segments a webpage into multiple blocks and then judges the relevance of every content block, extracting hyperlinks with high comprehensive relevance. Finally, a BCW-based focused crawling strategy combining the CPE and CBS strategies (BCW_CC) is proposed and experimentally evaluated for focused crawling in two domains: rainstorm disasters and sports. The results demonstrate the effectiveness of the developed BCW_CC method.

Key words: Focused crawler (FC); Bayesian classifier; Information retrieval; Priority evaluation
https://doi.org/10.1631/FITEE.2400939 **CLC number:** TP391

1 Introduction

Owing to its diversity, timeliness, and sharing ability, the Internet has become an important information source. The 53rd survey report of the China Internet Network Information Center (2024) recorded 3.88 million websites and 382 billion webpages in China as of December 2023. The number of webpages has increased by 6.5% since December 2022. The number of webpages on the global Internet is even vaster and very difficult to estimate. Faced with such huge resources, traditional search engines such as Google and Baidu cannot always match users' personalized needs to topic-relevant webpages. Furthermore, some existing open-source crawler tools, such as WebCollector, Crawler4j, Scrapy, and Nutch, are generally limited by low recall and accuracy (Yu J and Liu, 2015; Hosseinkhani et al., 2021). Unlike general web crawlers, the focused crawler (FC) (Deng, 2020; Xiong and Yang, 2025) can filter webpages related to specific topics.

The FC comprehensively judges the topic relevance of webpages by setting a threshold or classifier based on various evaluation indicators of the webpages, according to the needs of users. The FC, which usually crawls the webpages of specific topics more accurately than traditional crawler tools, is widely applied in information filtering, precise information retrieval, data mining and analysis, and large models (for example, data crawling of pre-trained corpora). The FC can also

‡ Corresponding author
* Project supported by the Guangdong Basic and Applied Basic Research Foundation of China (No. 2023A1515011344) and the Guangdong Philosophy and Social Science Foundation Regular Project of China (No. GD24CGL54)
ORCID: Jingfa LIU, https://orcid.org/0000-0002-0407-1522
© Zhejiang University Press 2025

collect public opinions and observe the emotional tendencies on specific topics on social media and news websites, helping enterprises and governments understand the public's views on a certain topic. This paper investigates the FC on two topics: rainstorm disasters and sports events. Early warnings, preventive measures, and emergency response information are essential for reducing and avoiding losses caused by a rainstorm disaster and ensuring the safety of human life and property. In sports, the FC can not only recommend important sports events related to the user's interests but also provide users with relevant comments at a deeper level. However, the information in specific fields is generally sparse and scattered throughout the vast Internet, possessing big data characteristics that challenge the accuracy rate (AC) of information retrieval by FC.

Most of the current FC methods are based on heuristic strategies, semantic analysis, or machine learning.

1. Heuristic-based FCs are classifiable into webpage content-based and hyperlink structure-based FC methods. The main webpage content-based FC methods are the best-first search (Rawat and Patil, 2013), fish-search (Kumar and Gupta, 2021), and shark-search algorithms (Ding et al., 2022). However, these methods are insufficiently comprehensive and cannot correlate webpages with topics. To resolve this problem, Cheng et al. (2018) adopted a word-weighted vector clustering method that arranges the contents of hyperlinks to be visited and improves the mechanism of scoring adjacent hyperlinks. However, analysis strategies based on webpage content cannot capture the impact of hyperlink structure on relevance. The main hyperlink structure-based FC methods include the PageRank (Yuan et al., 2017; Yu LX et al., 2021) and hyperlink-induced topic search (Yang B et al., 2014; Khan et al., 2024) algorithms. Hyperlink structure-based FC methods focus on the structure but not the relevance of the topic, increasing the risk of "topic drift" in crawling. To solve these problems, Seyfi et al. (2016) proposed a crawling method based on hyperlink and webpage content, in which a hierarchical structure called T-Graph assigns an appropriate priority score to each unvisited hyperlink for prioritization.

Although FCs based on heuristic strategies can interpret webpage content and hyperlink structures, they cannot easily handle dynamic and multi-topic webpages, which may be topic-irrelevant overall while containing topic-relevant content blocks.

2. FCs based on semantic analysis usually adopt context graphs (CGs) or ontology. CG-based strategies have been extensively researched, culminating in relevancy CG (Hsu and Wu, 2006), concept similarity CG (Yang YK et al., 2008), concept CG (Guan and Luo, 2016), path-trust knowledge graphs (Du et al., 2017), and knowledge graphs (Jia et al., 2021). However, the performance of CG methods largely depends on users' query histories, and broad thematic relationships are not captured. In contrast, domain ontology clarifies the conceptual semantic hierarchies and relationships between concepts. The FC method of Liu JF et al. (2022a) combines latent Dirichlet allocation for semi-automatic domain ontology construction with the Apriori algorithm for ontology learning. Liu JF et al. (2022b) proposed a multi-objective optimization model based on web text and link structures, designing a web space evolution crawler framework. During sorting challenges, their approach selects Pareto optimal hyperlinks and creates a topic model leveraging domain ontology based on formal concept analysis. The FC approach of Liu JF et al. (2023) uses an improved tabu search algorithm with domain ontology and host information, which relies on a semantic disambiguation vector space model (SDVSM) established by the term frequency–inverse document frequency (TF–IDF) method. The SDVSM approach integrates the semantic disambiguation graph with the semantic VSM.

Although semantic analysis improves the accuracy and deepens the semantic relevance of focused crawling, its accuracy may be constrained by the limitations of semantic models, especially when dealing with polysemous, ambiguous, or domain-specific terminologies.

3. Machine learning approaches can enhance the effectiveness of crawling. Saleh et al. (2017) introduced a domain distiller that filters hyperlinks before they enter the queue. The domain distiller combines Naïve Bayes with support vector machines (SVMs), forming an optimized Naïve Bayes classifier. Zhang et al. (2021) proposed a public-opinion analysis method based on a combined crawler and SVM. Given that website administrators commonly group webpages with similar themes and languages within the same directory (segment), Suebchua et al. (2016) improved the accuracy of website-segment prediction with two

predictors: one learned from the features extracted from relevant source website segments, and the other learned from features extracted from irrelevant source website segments. Gao et al. (2023) proposed a reinforcement learning-based method that detects diverse and dynamic webpages using a feature selector and a session classifier. Dhanith et al. (2024) proposed weakly supervised learning for FC based on the gated recurrent unit mechanism, which inputs topic vectors and crawls webpages to generate meaningful semantic vectors. Ai and Yin (2024) proposed a topic crawler that integrates the Biterm topic model (BTM) and TextCNN model. They regarded the content topic discrimination module as a text classification problem. The text semantic information was enhanced by fusing the text topic vectors obtained from BTM with Word2Vec word vectors. The convolutional neural network was used to improve the accuracy of the discrimination module.

Most machine learning-based FCs share common drawbacks. First, they treat the entire webpage as a unified text in classification tasks, excluding the label attributes carried by the webpage structure itself. Second, during the crawling process, the complexity of web content and the isolation of relevant information usually prevent tunnel crossing, through which some topic-relevant webpages can be reached from low-relevance webpages. This limitation reduces the coverage of FC.

Here, we propose an improved Bayesian classifier with weights (BCW), which integrates a comprehensive priority evaluation (CPE) method for unvisited hyperlinks with a content block segmentation (CBS)-based crossing tunnel technique. This crawler, called BCW_CC, prevents topic drift caused by inaccurate webpage classification (a common problem with simple classifiers). The CBS-based technique allows the crawler to capture more topic-relevant webpages. Our main contributions are:

1. We propose a modified BCW that adds label weights to the feature words of the Bayesian classifier, enhancing the accuracy of webpage classification.

2. We propose a CPE method that considers the relevance of anchor text, hyperlink context, and the webpage pointed by unvisited hyperlinks, thus improving the topic-relevance evaluation of unvisited hyperlinks.

3. We design a CBS technique based on the backtracking approach, enabling tunnel crossing through which some topic-relevant webpages can be reached from low-relevance webpages.

2 Priority hyperlink evaluation method

This section first constructs the topic weight vector using the TF–IDF method (Wu et al., 2017). Based on the VSM (Farag et al., 2018), the topic-relevance computation methods of webpage, anchor text, and hyperlink context are then developed. Finally, the CPE method that calculates the topical relevance of hyperlinks is developed.

2.1 Construction of topic weight vector

Before constructing the topic weight vector, the training dataset was preprocessed through segmentation and data cleaning. Next, the weight of each feature word in the dataset was computed using the TF–IDF method. Denoting the topic feature vector of the dataset as $\boldsymbol{T}=[t_1, t_2, ..., t_i, ..., t_n]$ and the topic weight vector as $\boldsymbol{V}_{\mathrm{T}}=[\omega_{t_1}, \omega_{t_2}, ..., \omega_{t_i}, \cdots, \omega_{t_n}]$, where n is the number of topic feature words, the weight ω_{t_i} of topic feature word t_i in the training set is computed as

$$\omega_{t_i} = \mathrm{tf}_{t_i}\mathrm{idf}_{t_i} = \frac{f_{t_i}}{\sum_{m=1}^{n} f_{t_m}} \log_a \left(\frac{N}{N_{t_i}} + 0.01 \right), \quad (1)$$

where f_{t_i} represents the TF of the feature word t_i and tf_{t_i} is the normalized TF of t_i. The term idf_{t_i} defines the IDF of t_i. N and N_{t_i} represent the number of texts in the entire training set and the number of texts containing topic feature word t_i in the training set, respectively, and $a>1$.

2.2 Topic relevance of webpage text

In the HyperText Markup Language 4.0 standard developed by W3C, the content of a webpage is typically composed of multiple elements and labels. Topic feature words within different labels often exert varying impacts on the topic. Based on the literature (Liu JF et al., 2022b, 2023) and multiple experimental trials, different weights are assigned to the labels in webpage texts, as listed in Table 1.

A webpage text is mapped into a feature vector $\mathbf{DK}=[\mathrm{dk}_1, \mathrm{dk}_2, \cdots, \mathrm{dk}_i, \cdots, \mathrm{dk}_n]$. When constructing the topic feature weight vector of a webpage, we assume

Table 1 Division of labels and their weights

Group	Label	Meaning	Weight
Group 1	<title>, <description>, <keyword>, <h1>	Title, description, keyword, first-level headline	2.0
Group 2	<h2>, <h3>	Secondary-level headline, third-level headline	1.5
Group 3	<h4>, <h5>,	Fourth-level headline, fifth-level headline, bold text	1.2
Group 4	<p>, <td>,	Body information	1.0
Group 5	Other labels	Non-body information	0.2

that the weights of the same topic word can depend on the label to which the topic word belongs. The feature weight vector of the webpage is denoted as $\boldsymbol{V}_{\mathrm{DK}}=[w_{\mathrm{dk}_1}, w_{\mathrm{dk}_2}, \cdots, w_{\mathrm{dk}_i}, \cdots, w_{\mathrm{dk}_n}]$, where w_{dk_i} is the weight of the ith topic word on the webpage and is computed as

$$w_{\mathrm{dk}_i}=\sum_{j=1}^{J}\mathrm{tf}_{i,j}L_j=\sum_{j=1}^{J}\left(\frac{f_{i,j}}{\max f_{i,j}}L_j\right), \tag{2}$$

where $\mathrm{tf}_{i,j}$ and $f_{i,j}$ represent the normalized word frequency and word frequency of the ith topic word in the jth group, respectively, $\max f_{i,j}$ is the maximum word frequency of the ith topic word among all groups, and L_j is the weight associated with the jth label group. J represents the number of groups (J=5 in Table 1).

The relevance between a webpage and the given topic is computed in terms of the VSM. To this end, we compute the cosine similarity between $\boldsymbol{V}_{\mathrm{DK}}$ and $\boldsymbol{V}_{\mathrm{T}}$, thereby determining a topic relevance $R(G)$ between the feature vector **DK** of a webpage G and the topic feature vector $\boldsymbol{T}$. The cosine similarity calculation is given by

$$\begin{aligned}R(G)&=\mathrm{sim}(\boldsymbol{T},\mathbf{DK})=\frac{\boldsymbol{V}_{\mathrm{T}}\cdot\boldsymbol{V}_{\mathrm{DK}}}{\|\boldsymbol{V}_{\mathrm{T}}\|\|\boldsymbol{V}_{\mathrm{DK}}\|}\\&=\frac{\sum_{i=1}^{n}\omega_{t_i}w_{\mathrm{dk}_i}}{\sqrt{\sum_{i=1}^{n}\omega_{t_i}^2}\sqrt{\sum_{i=1}^{n}w_{\mathrm{dk}_i}^2}},\end{aligned} \tag{3}$$

where $R(G)$ takes values in the range [0, 1]. When the angle between vectors $\boldsymbol{V}_{\mathrm{DK}}$ and $\boldsymbol{V}_{\mathrm{T}}$ is 0°, the two vectors are maximally correlated. When the angle is 90°, webpage G is considered irrelevant. In summary, the webpage becomes increasingly more (less) relevant as $R(G)$ approaches 1 (0).

2.3 Topic relevance of anchor text and context

The anchor text of a hyperlink is a key indicator of the topic relevance of the hyperlink. However, the context of the hyperlink is more important than anchor texts such as "next" or "click here," which do not effectively represent the topic of the hyperlink. In this paper, we obtain the feature weight vectors of the anchor text using the TF–IDF method (similar to Eq. (1)).

After obtaining the feature weight vector of the anchor text $\boldsymbol{V}_{\mathrm{AL}}$ and the feature weight vector of context of the hyperlink $\boldsymbol{V}_{\mathrm{CL}}$ via the TF–IDF method, the topic relevance of the anchor text $R(\mathrm{AL})$ and the topic relevance of the context $R(\mathrm{CL})$ are calculated using the VSM, similar to Eq. (3).

2.4 Comprehensive priority evaluation of hyperlink

When evaluating an unvisited hyperlink l during focused crawling, the priority of that hyperlink cannot be determined from the hyperlink structure or content of the webpage alone. Therefore, we design the CPE method for hyperlink l. The relevance of webpage G $R(G)$ pointed by hyperlink l, the relevance of the anchor text $R(\mathrm{AL})$ of hyperlink l, and the relevance of the context $R(\mathrm{CL})$ related to hyperlink l are combined into a measure called the comprehensive priority $E(l)$ of the hyperlink l:

$$E(l)=\alpha R(G)+\beta R(\mathrm{AL})+\gamma R(\mathrm{CL}), \tag{4}$$

where α, β, and γ are weight coefficients satisfying $\alpha+\beta+\gamma=1$. Based on the grid-search method, we set $\alpha=0.5$, $\beta=0.3$, and $\gamma=0.2$.

3 Improved Bayesian classifier with weights

The FC aims to obtain relevant webpages for a given topic from the Internet and discard irrelevant webpages. Therefore, the FC can be regarded as a binary or multivariate classification problem to be solved by a classifier. In this paper, we apply our newly constructed improved BCW to webpage classification.

Based on the Bayesian formula with a prior probability and conditional probability for a given class, the Naïve Bayesian classification algorithm (Hu et al., 2023; He et al., 2025) calculates the posterior probability and hence predicts the most likely class of an item. When applying the Bayesian classification algorithm to webpage classification, we must select the characteristics of the webpages.

Suppose that webpage X is expressed as $\{U_1, U_2, \cdots, U_n\}$, where U_i is the ith feature word in the webpage. Assuming independence and given a test webpage X, the probability of X belonging to a certain class C_i through feature words is calculated as

$$\begin{aligned} P(C_i|X) &= P(C_i|U_1,U_2,\cdots,U_n) \\ &= \frac{P(C_i)P(U_1,U_2,\cdots,U_n|C_i)}{\sum_{j=1}^{m}P(C_j)P(U_1,U_2,\cdots,U_n|C_j)} \\ &= \frac{P(C_i)\prod_{k=1}^{n}P(U_k|C_i)}{\sum_{j=1}^{m}P(C_j)\prod_{k=1}^{n}P(U_k|C_j)}, \end{aligned} \tag{5}$$

where m is the number of classes and C_i is the ith class. Note that the denominator of Eq. (5) plays no role in determining the class of the classified webpage and can thus be disregarded. Taking only the numerator of Eq. (5), the class X of the webpage is computed as

$$C(X)=\left\{C_i \middle| \arg\max_i P(C_i)\prod_{k=1}^{n}P(U_k|C_i) \middle| i=1,2,\cdots,m\right\}. \tag{6}$$

From the webpage classification formula, one can conclude that the Bayesian training model mainly requires the prior probability $P(C_i)$ and the conditional probability $P(U_k|C_i)$ of class C_i, which are respectively calculated as

$$P(C_i) = \frac{K_i}{\sum_{j=1}^{m}K_j}, \tag{7}$$

$$P(U_k|C_i) = \begin{cases} \mathrm{tf}_{U_k}^{i}\mathrm{idf}_{U_k}, & U_k \in Q_i, \\ \dfrac{1}{M_i + M_{\mathrm{a}}}, & U_k \notin Q_i, \end{cases} \tag{8}$$

where K_i (i=1, 2, $\cdots$, m) denotes the number of documents belonging to class C_i in the training set, $\mathrm{tf}_{U_k}^{i}$ represents the normalized TF of feature word U_k in the ith class, idf_{U_k} is the IDF of feature word U_k in the training set, and Q_i is the set of feature words in the ith class C_i. M_i represents the total number of feature words in all texts belonging to class C_i in the training set, and M_{a} indicates the number of non-repeating feature words in all classes.

Traditional Bayesian classifiers underperform on webpage classification tasks, causing webpage misclassification and the "topic-drift" phenomenon. The accuracy is poor because generic webpage classifiers typically classify the entire webpage as a whole. Zhao et al. (2025) pointed out that different classifiers should be customized and trained to meet the specific needs of individual clients. FC classifiers can be customized based on the unique features of webpages. Within a webpage, the same feature word can exist in different label groups and carry different degrees of importance to the topic. To distinguish the importance of different feature words in different label groups, we introduce label weights L_j (j=1, 2, $\cdots$, J) to the Bayesian classifier as shown in Table 1. This strategy can amplify the importance of feature words representing the webpage's topic while reducing the weights of feature words irrelevant to that topic.

Combining Eqs. (6)–(8), we propose an improved BCW. The classification result of webpage X is determined by

$$\begin{aligned} C(X) = \Bigg\{ C_i \Bigg| & \arg\max_i P(C_i)\prod_{k=1}^{n}P(U_k|C_i)L_j \Bigg| \\ & i = 1, 2, \cdots, m;\ U_k \text{ is located in Group}_j, \\ & j \in \{1, 2, \cdots, J\} \Bigg\}. \end{aligned} \tag{9}$$

As the value of $\prod_{k=1}^{n}P(U_k|C_i)L_j$ becomes excessively small in actual computations, it is converted to logarithmic form $\log_b \prod_{k=1}^{n}P(U_k|C_i)L_j = \sum_{k=1}^{n}\log_b(P(U_k|C_i)L_j)$ for update, where b is an integer greater than 1. In this paper, b is set to 10.

4 Crawling strategies based on BCW and CBS

This section first introduces the tunneling technology based on CBS and then combines the BCW, CBS, and CPE into the BCW_CC. Finally, the time complexity of BCW_CC is analyzed.

4.1 Content block segmentation

The crossing tunnel technique based on the CBS traverses topic-irrelevant webpages to access topic-relevant webpages. A document object model tree of webpages generally contains numerous labels. A webpage analyzed as a whole may be topic-irrelevant but include topic-relevant content between the and labels. The CBS approach aims to segment a webpage into multiple sections using labels, enabling finer-grained analysis of a webpage. The weight vector of the content block is calculated by the TF–IDF method, similar to Eq. (1). The topic relevance of the content block is calculated using the VSM (similar to Eq. (3)). As nested labels are possible, identifying the innermost labels containing no other labels is essential. Here, we segment webpages using a specific CBS technique based on an outside-to-inside backtracking method.

4.2 FC based on BCW_CC

This subsection introduces our FC strategy BCW_CC, which combines the improved BCW, the CPE for hyperlinks, and the crossing tunnel technology based on CBS. The improved BCW classifier facilitates precise webpage filtering, the CPE reduces the likelihood of topic drift, and the tunneling technology enables the crawler to access otherwise inaccessible topic-relevant webpages.

First, we define the topic and construct a topic weight vector. We then add seed hyperlinks to the priority-queue and select a hyperlink source-link from the priority-queue. The webpage source-page pointed by the source-link is downloaded, and its topic is classified by the BCW. If the topic matches the target topic of the crawler, the topic relevance of the source-page R(source-page) is calculated by Eq. (3); otherwise, the current hyperlink is discarded, and a new hyperlink is fetched from the priority-queue. If R(source-page)> τ (τ is the topic relevance threshold of webpages), the webpage is assessed as topic-relevant; otherwise, it is partitioned into multiple (for example, r) content blocks using the CBS technology. The topic relevance of each content block B_i (i=1, 2, $\cdots$, r) $R(B_i)$ is then determined. If $R(B_i)>\lambda$ (the priority threshold of hyperlinks), all sub-links (denoted as block-links) in block B_i are obtained, and all webpages (denoted as block-pages) pointed by block-links are downloaded. Subsequently, all sub-links in block-pages are extracted. The comprehensive priority E(sub-link$_i$) of the ith sub-link sub-link$_i$ in sub-links is computed by Eq. (4). If E(sub-link$_i$)>λ, sub-link$_i$ is added to the priority-queue; otherwise, it is discarded. The above process iterates until the ending conditions are met. The detailed steps are given in Algorithm 1. By removing the BCW classifier from the BCW_CC algorithm, we obtain an FC called FCCBS, which combines CBS and CPE. In addition, by removing the CBS strategy from BCW_CC, we obtain an FC called FCBCW, which combines BCW and CPE.

4.3 Complexity analysis of BCW_CC

Let N represent the number of documents in the training set, L be the number of words in the document with the most words among all training documents, s be the number of seed hyperlinks, and n be the number of topic feature words. DP is the number of downloaded webpages and x is the number of sub-links in the webpage with the most sub-links among all downloaded webpages.

During the initial stage of the algorithm, the time complexity of adding the s seed hyperlinks to the priority-queue is $O(s)$. Next, a topic weight vector is constructed by the TF–IDF method. This step, with a time consumption of $O(nLN)$, involves calculating the TF and IDF of N training documents, each containing n topic feature words. The time consumption of selecting a head hyperlink from the priority-queue is $O(s)$. The time complexities of segmenting and obtaining the feature vector of the downloaded webpage source-page are $O(L)$ and $O(nL)$, respectively. Because the time consumption of computing the prior probability $P(C_i)$ and conditional probability $P(U_k|C_i)$ of class C_i in the trained BCW classifier are $O(N)$ and $O(LN)$, respectively, determining the topic of the downloaded webpage source-page requires $O(c_1NnLN)$ time, where c_1 is the number of theme categories. The time consumption of calculating the topic relevance and extracting all sub-links of the source-page are $O(n)$ and $O(x)$, respectively. In the tunnel-crossing based on CBS, the topic relevance of all r innermost content blocks of a webpage is computed in $O(c_2rn)$ time, where c_2 is the maximum layer number nesting . Thus, the time consumption of tunneling (i.e., obtaining all x sub-links in the topic-relevant content blocks in a topic-irrelevant

Algorithm 1 BCW_CC

Input: seed hyperlinks
Output: downloaded topic-relevant webpages

1 Determine the topic S, add the seed hyperlinks to the priority-queue, and initialize parameters τ=0.70, λ=0.30, temp=0, α=0.5, β=0.3, γ=0.2, DP=0, and RP=0. // DP is the number // of downloaded webpages, and RP is the number of topic- // relevant webpages. In the paper, the topic S is the "rain- // storm disaster" or "sports."
2 Construct the topic weight vector by the TF–IDF method.
3 **If** priority-queue is not empty and DP<15 000 **then**
4 Select the head hyperlink source-link from the priority-queue;
5 **Else**
6 Output the downloaded webpages and end the algorithm.
7 **End if**
8 Download the webpage source-page to which the source-link points, and let DP=DP+1.
9 Segment the source-page to obtain its feature vector.
10 Determine the topic of the source-page using the trained BCW classifier and Eq. (9).
11 **If** C(source-page) is equal to S **then**
12 Calculate the topic relevance of the source-page using Eq. (3) and go to lines 16–31;
13 **Else**
14 Discard the source-link and go to lines 3–7.
15 **End if**
16 **If** R(source-page)>τ **then**
17 Let RP=RP+1;
18 Extract all sub-links in the webpage source-page;
19 **Else** // Crossing tunnels based on the CBS strategy.
20 Using the backtracking method based on labels, divide the source-page into r content blocks B_i (i=1, 2, ⋯, r);
21 **For** i=1 to r **do**
22 Calculate the topic relevance $R(B_i)$ of B_i;
23 **If** $R(B_i)>\lambda$ **then**
24 Obtain all block-links in B_i, and download all webpage block-pages to which block-links point;
25 Extract all sub-links in the block-pages, and set temp=1;
26 **End if**
27 **End for**
28 **If** temp is equal to 1 **then** go to lines 32–39;
29 **Else** go to lines 3–7.
30 **End if**
31 **End if**
32 **For** i=1 to x **do** // x is the number of sub-links.
33 Compute the comprehensive priority E(sub-link$_i$) of each sub-link$_i$ in sub-links using Eq. (4).
34 **If** E(sub-link$_i$)>λ **then**
35 Add sub-link$_i$ to the priority-queue;
36 **Else**
37 Discard sub-link$_i$.
38 **End if**
39 **End for**
40 Go to lines 3–7.

webpage) is $O(c_2rnx)$. In addition, calculating the topic relevance of the webpage $R(G)$, the topic relevance of anchor text R(AL), and the topic relevance of content block R(CL) of a sub-link all consume $O(n)$ time, so the time consumption of calculating the comprehensive priority of all x sub-links is $O(xn)$. Therefore, for DP downloaded webpages, the time complexity of BCW_CC is $O[s+nLN+\mathrm{DP}(s+L+nL+c_1NnLN+n+x+c_2rnx+rxn)]$. Because $L\sim O(n)$, $r\sim O(n)$, and $x\sim O(s)$, the time complexity of BCW_CC can be further simplified to $O[\mathrm{DP}(N^2+s)n^2]$.

5 Numerical results and analysis

To evaluate the effectiveness of the proposed FCs and compare their performances with several state-of-the-art crawling algorithms, we selected two topics: rainstorm disasters and sports. All the experiments were implemented using Python and conducted on a personal computer equipped with a 1.8 GHz CPU and 8.0 GB of memory.

5.1 Dataset description

The dataset was THUCNews (http://thuctc.thunlp.org/), obtained by filtering historical data from the Sina News really simple syndication (RSS) feed channel between 2005 and 2011. Based on the original dataset, we reorganized the dataset into four candidate classes: real estate (RE), stocks (ST), technology (TE), and sports (SP). To obtain a rainstorm disaster-related dataset, we performed web scraping using a generic web crawler. After removing the irrelevant webpages, we created a dataset rainstorm disasters (RD). The five datasets are described in Table 2.

5.2 Metric indices

The common measures of FC performance are the AC and recall rate (RC). The AC represents the ratio of topic-relevant webpages crawled by the crawler to the total number of webpages crawled, whereas the RC represents the ratio of topic-relevant webpages crawled by the crawler to the total number of topic-relevant webpages RP on the entire Internet. These indices are given by

$$\mathrm{AC}(\mathrm{DP}) = \frac{\sum_{i=1}^{\mathrm{DP}} r_i}{\mathrm{DP}}, \tag{10}$$

Table 2 Datasets of the five classes

Class	Size of the training set	Size of the test set	Topic description
RE	1000	200	News and articles related to the real estate market
ST	1000	200	Information related to stock markets and finance
TE	1000	200	News, innovations, and technological developments in the technology industry
RD	1000	200	News and information on rainstorm disasters
SP	1000	200	Sports events, athletes, and sports news

where $r_i = 1$ if page G_i is relevant to topic and $r_i = 0$ otherwise. DP is the number of downloaded webpages, and

$$\mathrm{RC} = \frac{|P_1 \cap P_2|}{|P_1|}, \tag{11}$$

where P_1 and P_2 refer to the sets of uniform resource locators (URLs) of the target pages and the result pages grabbed by the crawlers, respectively.

Referring to Fan et al. (2022), we selected two additional measures for evaluating the crawling algorithm: the average topic relevance (AR) and standard deviation (SD) of all downloaded webpages, which are calculated as

$$\mathrm{AR} = \frac{1}{\mathrm{DP}} \sum_{i=1}^{\mathrm{DP}} R(G_i), \tag{12}$$

$$\mathrm{SD} = \sqrt{\frac{1}{\mathrm{DP}} \sum_{i=1}^{\mathrm{DP}} (R(G_i) - \mathrm{AR})^2}, \tag{13}$$

where $R(G_i)$ is the topic relevance of webpage G_i. Note that the SD measures the spread of topic relevance among all downloaded webpages. The value of SD lies in [0, 1].

5.3 Experimental results and analysis

Experiments were performed in the rainstorm disaster and sports domains. Referring to Liu JF et al. (2023), we set the total number of downloaded webpages to 15 000. With 15 000 downloaded webpages, we can understand the trend of each crawler algorithm and evaluate the performance indices of different crawlers.

5.3.1 Experimental results in the rainstorm disaster domain

In the rainstorm disaster experiments, the initial seed hyperlinks were obtained through Baidu, which ranks among the most authoritative and widely used search engines in China. Webpages were collected by searching for the keyword "rainstorm disaster." The top 24 URLs of the obtained webpages were employed as the initial seed hyperlinks.

Within the same experimental environment in the rainstorm disaster domain, we evaluated the breadth-first search (BFS) (Wang, 2011), optimal priority search (OPS) (Rawat and Patil, 2013), the FC based on the simulated annealing algorithm (FCSA) (Liu JF et al., 2019), the FC based on an improved tabu search strategy combining ontology and host information (FCITS_OH) (Liu JF et al., 2023), the FC based on particle swarm optimization (FCPSO) (Liu JF et al., 2024), the FCBCW algorithm, the FCCBS algorithm, and the BCW_CC algorithm. The BCW classifier in the FCBCW and BCW_CC algorithms was initially trained on the RE, ST, TE, SP, and RD datasets prior to crawling in the rainstorm disaster domain. The four performance metrics of the eight crawling algorithms (in units of 1000 downloads) are compared in Figs. 1–4.

Fig. 1 displays the numbers of topic-relevant webpages RP obtained by the eight crawling algorithms in the rainstorm disaster domain. After 15 000 downloaded webpages, the BCW_CC crawls 13 522 topic-relevant webpages, while the BFS, OPS, FCSA, FCITS_OH,

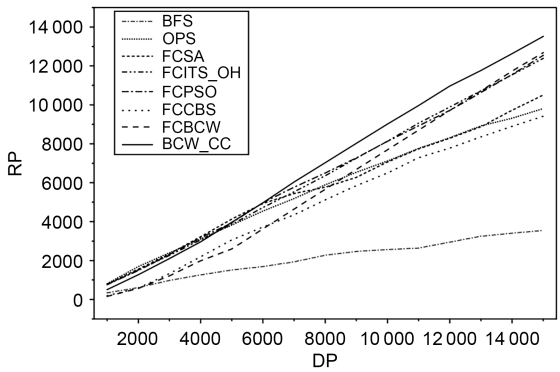

Fig. 1 RP obtained by the eight crawling algorithms in the rainstorm disaster domain

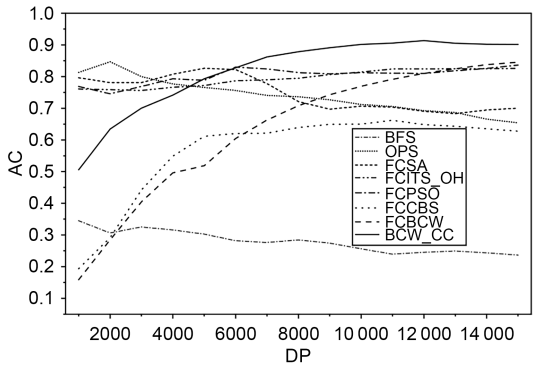

Fig. 2 ACs of the eight crawling algorithms in the rainstorm disaster domain

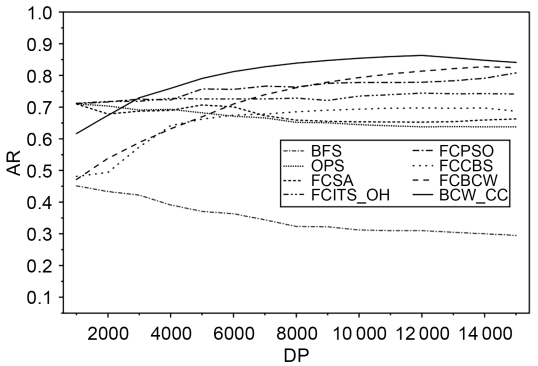

Fig. 3 AR values of the webpages obtained by the eight crawling algorithms in the rainstorm disaster domain

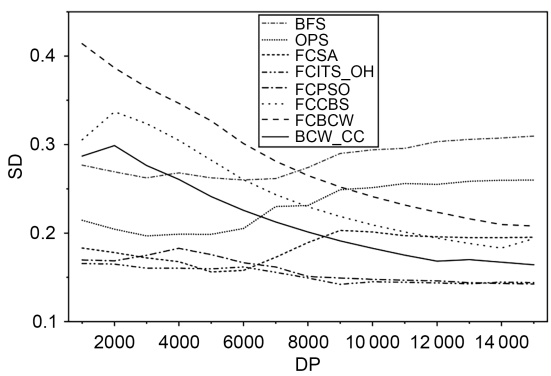

Fig. 4 SDs of the webpages downloaded by the eight crawling algorithms in the rainstorm disaster domain

FCPSO, FCCBS, and FCBCW crawl 3549, 9813, 10 506, 12 384, 12 546, 9413, and 12 680 RPs, respectively. Obviously, BCW_CC outperforms the other algorithms in terms of the RP metric.

Fig. 2 shows the ACs of the eight algorithms in the rainstorm disaster domain. When DP exceeds 7000, BCW_CC achieves a significantly higher AC than the other algorithms. At DP=10 000, the AC of BCW_CC gradually stabilizes. The final ACs of the BFS, OPS, FCSA, FCITS_OH, FCPSO, FCCBS, FCBCW, and BCW_CC crawling algorithms are 0.2366, 0.6542, 0.7004, 0.8256, 0.8364, 0.6275, 0.8453, and 0.9015, respectively. BCW_CC exhibits the highest accuracy among the eight algorithms.

Fig. 3 displays the AR values of the webpages downloaded by the eight crawling algorithms. At DP= 15 000, the ARs of the BFS, OPS, FCSA, FCITS_OH, FCPSO, FCCBS, FCBCW, and BCW_CC crawling algorithms are 0.2947, 0.6376, 0.6627, 0.7412, 0.8079, 0.6874, 0.8246, and 0.8412, respectively. After 3000 downloaded webpages, the BCW_CC algorithm outperforms the other algorithms in AR.

As demonstrated in Figs. 1–3, the OPS algorithm performs well during the early stages but its greedy strategy degrades the later-stage performance. Although FCSA also adopts the greedy strategy, it better optimizes the search direction than the OPS and BFS, owing to its metropolis sampling mechanism. The parameter-dependent FCSA slightly outperforms the OPS and BFS but is outperformed by the other algorithms overall. The FCITS_OH and FCPSO, which use adaptive hyperlink selection, outperform BFS, OPS, and FCSA but obtain lower RP, AC, and AR values than FCBCW and BCW_CC because they lack tunnel traversal and machine learning classification. The BCW_CC algorithm outperforms FCCBS and FCBCW. FCCBS uncovers hidden pages via tunneling but lacks accurate classification, whereas FCBCW prioritizes precision during the early stages but lacks tunneling for deep relevant-page retrieval. As the BCW classifier facilitates the accumulation of high-accuracy topic-relevant webpages, the accuracy of the FCBCW algorithm gradually enhances. Obviously, the BCW and CBS are more effective when combined than when individually employed in FC, validating the synergistic enhancement hypothesis proposed in Section 4.2, that is, collaboration between the BCW classifier and CBS tunneling technology.

Fig. 4 shows the SDs of the webpages downloaded by the eight crawling algorithms in the rainstorm disaster domain. After crawling 2000 webpages, BCW_CC exhibits a downward trend in SD. At DP= 15 000, the SDs of the BFS, OPS, FCSA, FCITS_OH, FCPSO, FCCBS, FCBCW, and BCW_CC crawling

algorithms are 0.3096, 0.2599, 0.1953, 0.1441, 0.1424, 0.1937, 0.2080, and 0.1643, respectively. As shown in Fig. 4, the SD of BCW_CC exceeds those of FCITS_OH and FCPSO but was lower than those of the other five algorithms. Table 3 compares the four metrics of the eight FCs at DP=15 000.

Table 3 Comparison of metric indices of the eight crawling algorithms at DP=15 000 in the rainstorm disaster domain

Algorithm	RP	AC	AR	SD
BFS	3549	0.2366	0.2947	0.3096
OPS	9813	0.6542	0.6376	0.2599
FCSA	10 506	0.7004	0.6627	0.1953
FCITS_OH	12 384	0.8256	0.7412	0.1441
FCPSO	12 546	0.8364	0.8079	0.1424
FCCBS	9413	0.6275	0.6874	0.1937
FCBCW	12 680	0.8453	0.8246	0.2080
BCW_CC	13 522	0.9015	0.8412	0.1643

To determine the main differences among the behaviors of the eight crawling algorithms, we conducted a Friedman test (Yu ZW et al., 2017) of the algorithms, ranking their scores of each metric from best to worst. In this paper, the RP, AC, AR, and SD of each algorithm are ranked from 1 to 8 and averaged to give a final score for each algorithm. A smaller average value indicates a higher-performance algorithm. The results are listed in Table 4. BCW_CC and BFS deliver the best and worst overall performances, respectively.

Table 4 Comparison of Friedman values of eight crawling algorithms in the rainstorm disaster domain

Algorithm	Friedman value				Average
	RP	AC	AR	SD	
BFS	8	8	8	8	8
OPS	6	6	7	7	6.5
FCSA	5	5	6	5	5.25
FCITS_OH	4	4	4	2	3.5
FCPSO	3	3	3	1	2.5
FCCBS	7	7	5	4	5.75
FCBCW	2	2	2	6	3
BCW_CC	1	1	1	3	1.5

5.3.2 Experimental results in the sports domain

For the SP experiments, we searched the keyword “sports” in Baidu and selected the top 15 URLs of the obtained sports-related webpages as the initial seed hyperlinks. We further obtained the URLs of 10 000 webpages related to “sports” in Baidu and Google and manually filtered them to obtain a target page collection.

Five crawling algorithms—BFS, OPS, the FCCBS algorithm, the FCBCW algorithm, and the BCW_CC algorithm—are implemented in the sports domain in the same experimental environment. Figs. 5–9 compare the values of the five metric indices (RP, AC, AR, SD, and RC) obtained by the five crawling algorithms (in units of 1000 downloads).

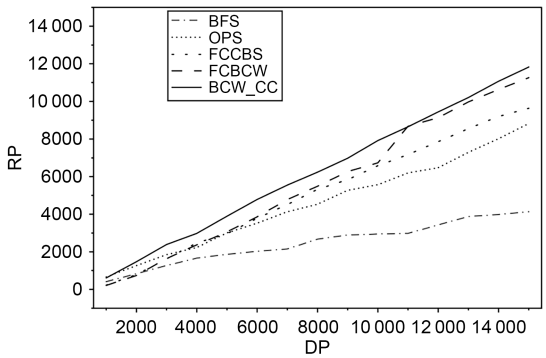

Fig. 5 RP obtained by the five crawling algorithms in the sports domain

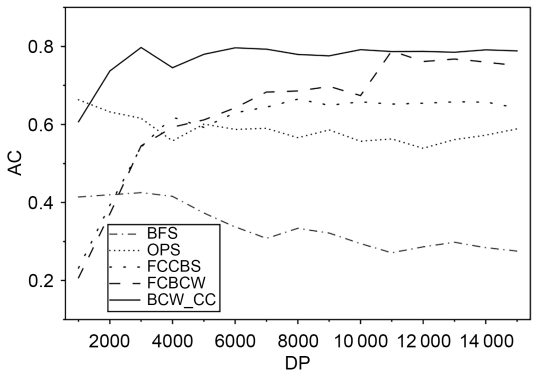

Fig. 6 ACs of the five crawling algorithms in the sports domain

Fig. 5 shows the RP of the five crawling algorithms in the sports domain. At DP=15 000, BCW_CC crawls 11 828 topic-relevant webpages, whereas the BFS, OPS, FCCBS, and FCBCW algorithms crawl 4131, 8828, 9635, and 11 260 webpages, respectively.

Fig. 6 presents the ACs of the five algorithms in the sports domain. BCW_CC significantly outperforms the other algorithms at DP=2000, but its AC performance gradually stabilizes toward DP=6000. The final

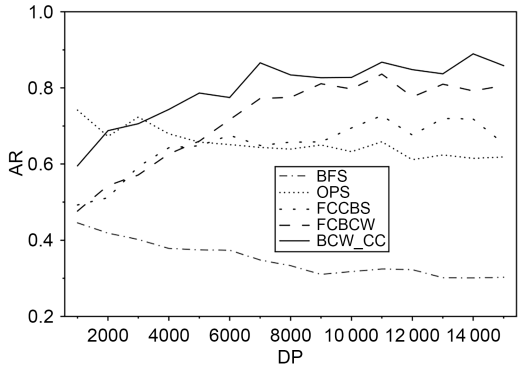

Fig. 7 ARs of the webpages obtained by the five crawling algorithms in the sports domain

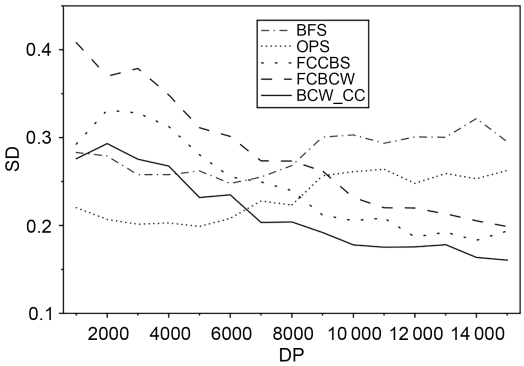

Fig. 8 SDs of the webpages obtained by the five crawling algorithms in the sports domain

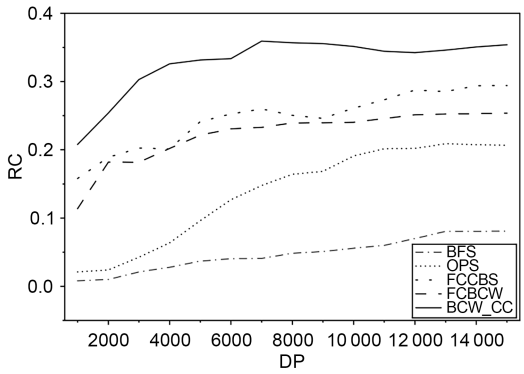

Fig. 9 RCs of the five crawling algorithms in the sports domain

ACs of the BFS, OPS, FCCBS, FCBCW, and BCW_CC algorithms are 0.2754, 0.5885, 0.6423, 0.7507, and 0.7885, respectively. Clearly, BCW_CC achieves the highest accuracy performance, with a 33.98% improvement over the traditional OPS algorithm.

Fig. 7 displays the ARs of the webpages downloaded by the five crawling algorithms. At DP=15 000, the ARs of the BFS, OPS, FCCBS, FCBCW, and BCW_CC crawling algorithms are 0.3027, 0.6184, 0.6534, 0.8061, and 0.8586, respectively. After crawling more than 3000 webpages, the BCW_CC algorithm downloads more relevant webpages than the other algorithms.

Fig. 8 displays the SDs of the webpages downloaded by the five crawling algorithms. At DP=15 000, the SDs of the BFS, OPS, FCCBS, FCBCW, and BCW_CC crawling algorithms are 0.2952, 0.2627, 0.1937, 0.1987, and 0.1606, respectively. The SD of BCW_CC trends downward overall, indicating that as more websites are crawled, BCW_CC stabilizes more strongly than the other crawling algorithms.

Fig. 9 shows the RC results of the five algorithms in the sports domain. The RCs of all five crawling strategies trend upward during the early stages and gradually stabilize with increasing DP, eventually becoming parallel to the *X*-axis. The final RCs of the BFS, OPS, FCCBS, FCBCW, and BCW_CC algorithms are 0.0808, 0.2067, 0.2941, 0.2537, and 0.3539, respectively. At DP=15 000, BCW_CC achieves the best recall performance, with a 71.21% improvement over the traditional OPS algorithm. Table 5 compares the five metrics of the five FCs at DP=15 000.

Table 5 Comparison of five metric indices obtained by five crawling algorithms in the sports domain at DP=15 000

Algorithm	RP	AC	AR	SD	RC
BFS	4131	0.2754	0.3027	0.2952	0.0808
OPS	8828	0.5885	0.6184	0.2627	0.2067
FCCBS	9635	0.6423	0.6534	0.1937	0.2941
FCBCW	11 260	0.7507	0.8061	0.1987	0.2537
BCW_CC	11 828	0.7885	0.8586	0.1606	0.3539

As evidenced in Figs. 5–9 and Table 5, FCs adopting the CBS or BCW strategy outperform the traditional BFS and OPS topic crawler algorithms, but BCW_CC combining both strategies most significantly improves the performance over the traditional algorithms, primarily because it integrates tunnel crossing with a page-filtering strategy. By traversing some irrelevant webpages, the CBS-based tunneling technique enables BCW_CC to capture otherwise inaccessible topic-relevant webpages, while the BCW reduces interference from irrelevant pages, enhancing both the crawling efficiency and precision.

5.4 Comparative experiments of key parameters

Some key parameters significantly impact the performance of crawling algorithms. Through comparative experiments, we analyzed two critical BCW_CC parameters: τ and λ. A strict τ can improve the precision of the crawler but reduces the coverage of borderline pages; conversely, a lenient τ increases crawling of unrelated pages. Similarly, the threshold λ significantly determines whether hyperlinks are added into the priority-queue. Setting an appropriate λ expands the crawling coverage of the crawler, uncovering more topic-relevant webpages.

To systematically evaluate the parameter sensitivity, we adopted a grid-search methodology across empirically validated ranges. Drawing on domain expertise and established practices (Liu WJ and Du, 2014), we confined the τ and λ parameters to [0.60, 0.80] and [0.20, 0.40] in the rainstorm disaster domain, respectively.

Table 6 presents the ACs of BCW_CC at DP= 15 000 in the rainstorm disaster domain as τ and λ vary over the above ranges. As λ increases from 0.20 to 0.30, the average AC improves by 19.67% (from 0.7416 to 0.8875) across the range of τ. However, when λ exceeds 0.35, the average AC gradually reduces, suggesting that λ=0.30 is an appropriate threshold. From Table 6, we similarly conclude that 0.70 is the optimal choice of τ. Accordingly, we set λ=0.30 and τ=0.70 in this paper.

Table 6 ACs of BCW_CC in the rainstorm disaster domain with different thresholds of τ and λ

τ	AC				
	λ=0.20	0.25	0.30	0.35	0.40
0.60	0.7508	0.8122	0.8819	0.8416	0.8103
0.65	0.7615	0.8344	0.8784	0.8644	0.8008
0.70	0.7594	0.8516	0.9015	0.8745	0.8110
0.75	0.7155	0.8466	0.8910	0.8630	0.8064
0.80	0.7206	0.8484	0.8846	0.8516	0.7949

6 Conclusions

Unlike traditional web crawlers that often prioritize large-scale crawling, FCs tend to crawl over themed webpages. However, as web content is complex and relevant information is isolated, FC methods are frequently degraded by topic drift and inability to cross tunnels. Moreover, the accuracy of classifiers in webpage classification is relatively low. To address these limitations, we combine an improved BCW with a CPE method for unvisited hyperlinks, which avoids topic drift caused by inaccurate webpage classification. Meanwhile, the CBS-based crossing tunnel technique allows the crawler to capture otherwise inaccessible topic-relevant webpages. We thus propose a focused crawling algorithm called BCW_CC and compare its performance with those of state-of-the-art methods such as FCITS_OH and FCPSO. The experimental results in the rainstorm disaster domain demonstrate higher accuracy and stronger overall performance of BCW_CC than the other algorithms (Friedman metric). Furthermore, we compare the experimental results of BCW_CC with those of FCBCW using the BCW strategy alone, FCCBS using the CBS strategy alone, and conventional topic crawlers. The results in both the rainstorm disaster and sports domains reaffirm the effectiveness of the improved strategies.

However, the drawbacks of the BCW_CC algorithm should not be ignored. BCW_CC lacks semantic analysis of the words in webpage texts and does not consider the hyperlink structure. In following work, we plan to optimize the similarity assessment and weight coefficient calculation from a semantic perspective. Moreover, FCs could benefit from recent advancements in deep learning. Crawlers incorporating word embedding into deep learning could learn from related corpora and are expected to expand the search scope and further improve the accuracy of crawling. In addition, the presently proposed FCs are designed to crawl 15 000 topical webpages. This relatively small number of crawled pages can meet current needs on a PC, but when upscaled to large-scale data crawling, the crawling time may become excessive. Large-scale web crawling requires a cluster of multiple servers or virtual hosts in a distributed computing environment, which can balance the load of task allocations during big data crawling. Therefore, we hope to create a more powerful distributed web-crawling system based on a Spark framework for large-scale web crawling.

Contributors

Jingfa LIU designed the research. Yongchuang WU drafted the paper, implemented the software, and performed the experiments. Jingfa LIU and Zhaoxia LIU revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Ai FJ, Yin XY, 2024. Research on text semantic enhancement topic crawler integrating BTM and TextCNN. *Softw Guide*, 23(3):21-26 (in Chinese). https://doi.org/10.11907/rjdk.231116

Cheng YK, Liao WJ, Cheng G, 2018. Strategy of focused crawler with word embedding clustering weighted in shark-search algorithm. *Comput Digit Eng*, 46(1):144-148 (in Chinese). https://doi.org/10.3969/j.issn.1672-9722.2018.01.031

China Internet Network Information Center, 2024. The 53rd Statistical Report on China's Internet Development (in Chinese). https://www.cnnic.net.cn/NMediaFile/2024/0325/MAIN1711355296414FIQ9XKZV63.pdf [Accessed on Mar. 25, 2024].

Deng SQ, 2020. Research on the focused crawler of mineral intelligence service based on semantic similarity. *J Phys Conf Ser*, 1575(1):012142. https://doi.org/10.1088/1742-6596/1575/1/012142

Dhanith PRJ, Saeed K, Rohith G, et al., 2024. Weakly supervised learning for an effective focused web crawler. *Eng Appl Artif Intell*, 132:107944. https://doi.org/10.1016/j.engappai.2024.107944

Ding SC, Liu K, Fang Z, 2022. Crawler with dynamic thesaurus and improved shark-search algorithm: case study of military equipment. *Data Anal Knowl Discov*, 6(8):52-60 (in Chinese). https://doi.org/10.11925/infotech.2096-3467.2021.1125

Du YJ, Li CX, Hu Q, et al., 2017. Ranking webpages using a path trust knowledge graph. *Neurocomputing*, 269:58-72. https://doi.org/10.1016/j.neucom.2016.08.142

Fan GF, Zhang LZ, Yu M, et al., 2022. Applications of random forest in multivariable response surface for short-term load forecasting. *Int J Electr Power Energy Syst*, 139:108073. https://doi.org/10.1016/j.ijepes.2022.108073

Farag MMG, Lee S, Fox EA, 2018. Focused crawler for events. *Int J Digit Libr*, 19(1):3-19. https://doi.org/10.1007/s00799-016-0207-1

Gao Y, Feng ZL, Wang XY, et al., 2023. Reinforcement learning based web crawler detection for diversity and dynamics. *Neurocomputing*, 520:115-128. https://doi.org/10.1016/j.neucom.2022.11.059

Guan WG, Luo YC, 2016. Design and implementation of focused crawler based on concept context graph. *Comput Eng Des*, 37(10):2679-2684 (in Chinese). https://doi.org/10.16208/j.issn1000-7024.2016.10.019

He YL, Ou GL, Fournier-Viger P, et al., 2025. Attribute grouping-based naive Bayesian classifier. *Sci China Inform Sci*, 68(3): 132106. https://doi.org/10.1007/s11432-022-3728-2

Hosseinkhani J, Taherdoost H, Keikhaee S, 2021. ANTON framework based on semantic focused crawler to support web crime mining using SVM. *Ann Data Sci*, 8(2):227-240. https://doi.org/10.1007/s40745-019-00208-5

Hsu CC, Wu F, 2006. Topic-specific crawling on the Web with the measurements of the relevancy context graph. *Inform Syst*, 31(4-5):232-246. https://doi.org/10.1016/j.is.2005.02.007

Hu ZW, Cui JJ, Lin A, 2023. Identifying potentially excellent publications using a citation-based machine learning approach. *Inform Process Manag*, 60(3):103323. https://doi.org/10.1016/j.ipm.2023.103323

Jia Z, Pramanik S, Roy RS, et al., 2021. Complex temporal question answering on knowledge graphs. Proc 30th ACM Int Conf on Information and Knowledge Management, p.792-802. https://doi.org/10.1145/3459637.3482416

Khan M, Mello GBM, Habib L, et al., 2024. HITS-based propagation paradigm for graph neural networks. *ACM Trans Knowl Discov Data*, 18(4):100. https://doi.org/10.1145/3638779

Kumar S, Gupta M, 2021. A review of focused crawling schemes for search engine. In: Zhang YD, Senjyu T, So–In C, et al. (Eds.), Smart Trends in Computing and Communications: Proceedings of SmartCom 2020. Springer, Singapore, p.311-317. https://doi.org/10.1007/978-981-15-5224-3_30

Liu JF, Li F, Jiang SY, 2019. Focused annealing crawler algorithm for rainstorm disasters based on comprehensive priority and host information. *Comput Sci*, 46(2): 215-222 (in Chinese). https://doi.org/10.11896/j.issn.1002-137X.2019.02.033

Liu JF, Dong Y, Liu ZX, et al., 2022a. Applying ontology learning and multi-objective ant colony optimization method for focused crawling to meteorological disasters domain knowledge. *Expert Syst Appl*, 198:116741. https://doi.org/10.1016/j.eswa.2022.116741

Liu JF, Li X, Zhang QS, et al., 2022b. A novel focused crawler combining Web space evolution and domain ontology. *Knowl Based Syst*, 243:108495. https://doi.org/10.1016/j.knosys.2022.108495

Liu JF, Wang Z, Zhong G, et al., 2023. A new focused crawler using an improved tabu search algorithm incorporating ontology and host information. *Front Inform Technol Electron Eng*, 24(6):859-875. https://doi.org/10.1631/FITEE.2200315

Liu JF, Yang ZH, Yan XM, et al., 2024. Applying particle swarm optimization-based dynamic adaptive hyperlink evaluation to focused crawler for meteorological disasters. *Complex Intell Syst*, 10(1):233-255. https://doi.org/10.1007/s40747-023-01121-4

Liu WJ, Du YJ, 2014. A novel focused crawler based on cell-like membrane computing optimization algorithm. *Neurocomputing*, 123:266-280. https://doi.org/10.1016/j.neucom.2013.06.039

Rawat S, Patil DR, 2013. Efficient focused crawling based on best first search. Proc 3rd IEEE Int Advance Computing Conf, p.908-911. https://doi.org/10.1109/IAdCC.2013.6514347

Saleh AI, Abulwafa AE, Al Rahmawy MF, 2017. A web page distillation strategy for efficient focused crawling based on optimized Naïve Bayes (ONB) classifier. *Appl Soft Comput*,

53:181-204.
https://doi.org/10.1016/j.asoc.2016.12.028

Seyfi A, Patel A, Júnior JC, 2016. Empirical evaluation of the link and content-based focused treasure-crawler. *Comput Stand Interfaces*, 44:54-62.
https://doi.org/10.1016/j.csi.2015.09.007

Suebchua T, Rungsawang A, Yamana H, 2016. Adaptive focused website segment crawler. Proc 19th Int Conf on Network-Based Information Systems, p.181-187.
https://doi.org/10.1109/NBiS.2016.5

Wang H, 2011. Design and implementation of theme crawling based on breadth first. MS Thesis, Fudan University, Shanghai (in Chinese).
https://cdmd.cnki.com.cn/Article/CDMD-10246-1012330588.htm

Wu YL, Zhao SL, Li CJ, et al., 2017. Text classification method based on TF-IDF and cosine similarity. *J Chin Inform Process*, 31(5):138-145 (in Chinese).
https://doi.org/10.3969/j.issn.1003-0077.2017.05.020

Xiong GY, Yang BL, 2025. A self-decision topic crawler algorithm with online training. *J Beijing Univ Aeronaut Astronaut*, 51(2):602-615 (in Chinese).
https://doi.org/10.13700/j.bh.1001-5965.2023.0002

Yang B, Chen HC, Zhu GY, et al., 2014. A new web page ranking algorithm based on hyperlink diversity analysis. *Chin J Comput*, 37(4):833-847 (in Chinese).
https://doi.org/10.3724/SP.J.1016.2014.00833

Yang YK, Du YJ, Sun JY, et al., 2008. A topic-specific web crawler with concept similarity context graph based on FCA. Proc 4th Int Conf on Intelligent Computing, p.840-847.
https://doi.org/10.1007/978-3-540-85984-0_101

Yu J, Liu Q, 2015. Survey on topic-focused crawlers. *Comput Eng Sci*, 37(2):231-237 (in Chinese).
https://doi.org/10.3969/j.issn.1007-130X.2015.02.007

Yu LX, Li YL, Zeng QT, 2021. Design of topic Web crawler based on improved PageRank algorithm. *J Phys Conf Ser*, 1754(1):012210.
https://doi.org/10.1088/1742-6596/1754/1/012210

Yu ZW, Wang ZQ, You J, et al., 2017. A new kind of nonparametric test for statistical comparison of multiple classifiers over multiple datasets. *IEEE Trans Cybern*, 47(12):4418-4431.
https://doi.org/10.1109/TCYB.2016.2611020

Yuan ZQ, Zhang WH, Fu HJ, et al., 2017. A PageRank-improved ranking algorithm based on cheating similarity and cheating relevance. Proc IEEE/ACIS 16th Int Conf on Computer and Information Science, p.257-263.
https://doi.org/10.1109/ICIS.2017.7960003

Zhang HH, Li SH, Feng JY, et al., 2021. Public opinion analysis of Weibo comments based on crawler and SVM. Proc IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conf, p.589-593. https://doi.org/10.1109/IMCEC51613.2021.9482219

Zhao YF, He XT, Yu GX, et al., 2025. Personalized federated few-shot node classification. *Sci China Inform Sci*, 68(1):112105. https://doi.org/10.1007/s11432-024-4254-5