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

Focused crawling strategies based on ontologies and simulated annealing methods for rainstorm disaster domain knowledge[*#]

Jingfa LIU[†1,2], Fan LI[†‡3], Ruoyao DING[1,2], Zi'ang LIU[4]

[1]*Guangzhou Key Laboratory of Multilingual Intelligent Processing, Guangdong University of Foreign Studies, Guangzhou 510006, China*
[2]*School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006, China*
[3]*School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[4]*Faculty of Science, University of Alberta, Edmonton T6G2H6, Canada*
[†]E-mail: jfliu@nuist.edu.cn; bj2014_lifan@163.com

Received July 23, 2021; Revision accepted Mar. 23, 2022; Crosschecked June 17, 2022

Abstract: At present, focused crawler is a crucial method for obtaining effective domain knowledge from massive heterogeneous networks. For most current focused crawling technologies, there are some difficulties in obtaining high-quality crawling results. The main difficulties are the establishment of topic benchmark models, the assessment of topic relevance of hyperlinks, and the design of crawling strategies. In this paper, we use domain ontology to build a topic benchmark model for a specific topic, and propose a novel multiple-filtering strategy based on local ontology and global ontology (MFSLG). A comprehensive priority evaluation method (CPEM) based on the web text and link structure is introduced to improve the computation precision of topic relevance for unvisited hyperlinks, and a simulated annealing (SA) method is used to avoid the focused crawler falling into local optima of the search. By incorporating SA into the focused crawler with MFSLG and CPEM for the first time, two novel focused crawler strategies based on ontology and SA (FCOSA), including FCOSA with only global ontology (FCOSA_G) and FCOSA with both local ontology and global ontology (FCOSA_LG), are proposed to obtain topic-relevant webpages about rainstorm disasters from the network. Experimental results show that the proposed crawlers outperform the other focused crawling strategies on different performance metric indices.

Key words: Focused crawler; Ontology; Priority evaluation; Simulated annealing; Rainstorm disaster
https://doi.org/10.1631/FITEE.2100360 **CLC number:** TP39

1 Introduction

Crawler, which is an important part of search engines for information retrieval (IR), is a technology for automatically obtaining webpages. To acquire domain-specific knowledge, traditional crawlers have difficulties in implementing semantic analysis. Therefore, focused crawler technologies with topic preference characteristics have received great attention in recent years (Bajpai and Arora, 2018). A focused crawler (Tsikrika et al., 2016) aims to retrieve large-quantity and high-quality topic-relevant webpages in a short time. Focused crawler has many applications in the fields of business, transmission, biomedicine, and meteorology (Boukadi et al., 2018; Liu B et al., 2020). This paper addresses focused crawling on the topic of rainstorm disaster, which is one of the most frequent meteorological disasters. It is extremely important to obtain

[‡] Corresponding author

[*] Project supported by the Special Foundation of Guangzhou Key Laboratory of Multilingual Intelligent Processing, China (No. 201905010008), the Program of Science and Technology of Guangzhou, China (No. 202002030238), and the Guangdong Basic and Applied Basic Research Foundation, China (No. 2021A1515011974)

[#] Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2100360) contains supplementary materials, which are available to authorized users

ORCID: Jingfa LIU, https://orcid.org/0000-0002-0407-1522; Fan LI, https://orcid.org/0000-0001-7836-0522; Ruoyao DING, https://orcid.org/0000-0001-9282-7846; Zi'ang LIU, https://orcid.org/0000-0003-3475-9915

early warning to ensure preventive measures and emergency response avoidance or reduce the loss caused by rainstorm disasters, and important to ensure the safety of people and property.

However, the scale of webpages on the Internet is massive and continuously growing. The content of webpages is highly dynamic and complex. The information about webpages related to rainstorm disasters is sparse, showing the characteristics of big data. In the field of IR, traditional focused crawlers face great challenges in improving their accuracy. The main difficulties are the establishment of topic benchmark models, the assessment of topic relevance (including hyperlinks and texts), and the design of crawler strategies.

Semantic description methods on a given topic are the most popular topic modeling strategies, and include mainly the conceptual graph (CG) (Du et al., 2013, 2017; Guan and Luo, 2016) and domain ontology (Du et al., 2014; Zhu et al., 2017; Capuano et al., 2020; Khadir et al., 2021; Lakzaei and Shmasfard, 2021). Currently, the methods of determining the virtual concept are not uniform in CGs, and some methods require user interaction, which may cause topic deviations because of insufficient user knowledge or inaccurate understanding. Therefore, most crawling methods use domain ontology to specify the domain knowledge hierarchy, and compute conceptual semantic weights of the topic words occurring in the domain ontology. These are ultimately used to compute the topic relevance score to judge whether one text or hyperlink is relevant to a specific topic. Thereafter, during the crawling process, most scholars perform hyperlink filtering based on the priority (topic relevance) and a preset threshold. However, existing filtering methods on hyperlinks are all single-filtering processes, and a hyperlink filtering method with multiple criteria has not been considered.

In the focused crawler, the primary methods to predict the priority of an unvisited hyperlink (webpage) include two categories: hyperlink structure based method (Du et al., 2017) and web text analysis based method (Liu WJ and Du, 2014; Prakash and Kumar, 2015; Cheng et al., 2018). However, most research ignores the impact of the combination of these two methods, and the indicators considered are not comprehensive.

In the design of crawler strategies, breadth first search (BFS) (Vidal et al., 2006) and optimal priority search (OPS) (Rawat and Patil, 2013) are frequently applied. BFS ignores unvisited hyperlinks' priorities, so the performance of BFS is generally inferior to that of OPS. Most scholars now use the OPS crawler strategy, but the OPS strategy is a greedy algorithm, where it is easy for the search to be trapped in local optima. To avoid inherent flaws of the greedy algorithm, researchers have recently proposed some heuristic crawler methods based on meta-heuristic strategies, such as particle swarm optimization (PSO) (Tong, 2008), the genetic algorithm (GA) (Jing et al., 2016), the ant colony optimization (ACO) algorithm (Chen et al., 2011), and the tabu search (TS) algorithm (Liu JF et al., 2020). However, some improvements and developments are still required to enhance their effectiveness.

The simulated annealing (SA) algorithm (He et al., 2009) has a strong global search capability and can accept the sub-optimal links based on Metropolis sampling and avoid the focused crawling falling into local search. Thus, in this paper we apply the SA algorithm and ontology technology (which combines a multiple-filtering strategy and a comprehensive priority evaluation method (CPEM)) to execute focused crawling. Experimental results of the focused crawlers on the rainstorm disaster show the effectiveness of the proposed method. The main contributions of this paper are as follows:

1. A novel multiple-filtering strategy based on local ontology and global ontology (MFSLG) is proposed to find more topic-relevant hyperlinks.

2. A CPEM considering four indicators (topic relevance of the webpage containing the unvisited hyperlink, topic relevance of anchor text, the PageRank (PR) value, and topic relevance of the webpage to which the unvisited hyperlink points) is used to evaluate the unvisited hyperlinks.

3. An annealing strategy based on Metropolis sampling is applied to avoid the focused crawler falling into a local optimal search.

4. A new focused crawler combining domain ontology and the SA algorithm has been used to obtain the effective domain knowledge of the rainstorm disaster for the first time.

2 Construction of an ontology about rainstorm disaster

Formal concept analysis (FCA) is a semi-automated method of constructing ontology. The “concept lattice” is the core data structure of knowledge representation and is the core mathematical theory of FCA (Yang et al., 2008). Each node of the concept lattice is a concept that consists of an extension and an intension of the concept. We analyze the conceptual hierarchy relation and conceptual intrinsic link to construct the concept lattice. The construction of the concept lattice includes three main steps:

1. Data extraction

First, we obtain topic-relevant academic papers from the database CNKI (i.e., China National Knowledge Infrastructure) as data sources, and extract titles, abstracts, and keywords from the academic papers as a candidate set of domain terminologies. Then, we use the word segmentation technology to find the core vocabulary of the field and count the number of occurrences of each word.

2. Formal context creation

A formal context is also called a “Document–Terms” matrix, and can be defined as a triple: F= (Documents, Terms, Relation). Documents, Terms, and Relation represent the document collection, terminology, and relationship between documents and terms, respectively.

3. Concept lattice construction

The concept lattice is a Hasse graph (Zhu et al., 2017). Each node in the concept lattice is a concept C(Denotation, Connotation), where Denotation represents the extension of concept C and Denotation⊆ Documents, and Connotation represents the intension of concept C and Connotation⊆Terms. Generally, one can use the tool ConExp (https://sourceforge.net/projects/conexp/) to construct the concept lattice semi-automatically, that is, to generate the Hasse graph.

Example 1 (Construction of the concept lattice) First, we obtain five academic papers related to the topic of “rainstorm disaster” from CNKI. The selected field terms are Terms 1–8 (Term 1, rainstorm disaster; Term 2, disaster management; Term 3, emergency alert level; Term 4, weather monitoring; Term 5, hydraulic engineering; Term 6, city waterlogging; Term 7, floodwater; Term 8, landslide). According to the above method, we build binary relations of the formal context (Table 1) and the Hasse graph of the concept lattice (Fig. 1).

In Fig. 1, Terms 1–8 represent the attribute sets and Docs 1–5 represent the object sets. The upper half circle represents the attribute, and the lower half represents the object. If the attribute part of a node is blue, it means that there is a new attribute linked to the node. If the object part of a node is black, it means that a new object is linked to the node. The attribute set of each concept node is the sum of all the attributes at the upper level of the node (inheriting the parent concept attributes), and the object set is the sum of all the objects at the lower level of the node (covering the sub-concept objects). For example, in the “Doc 1, Term 2” node in Fig. 1, its attribute set is {Term 1, Term 2}, and the object set is {Doc 1, Doc 2, Doc 3}. As reported by Rios-Alvarado et al. (2013), a hyponym is defined as a word of more specific meaning than a general or superordinate term, and a hypernym is a word with a broad meaning constituting a category under which more specific words fall. Thus, Term 1 is the hypernym of Term 2, and Term 2 is the hyponymy of Term 1. The hyponymy of other terms in the figure can be obtained similarly.

After the concept lattice is built, we use the ontology web language (OWL) (http://www.w3.org/TR/owl-features/) to formalize the concept hierarchy. Each term is defined as a class, and the relationship between terms is defined as the hyponymy relation of the class. Generally, the development tool Protégé (https://protege.stanford.edu/) can be used to write and implement visualizations of the OWL.

Table 1 Binary relations of the formal context

Term No.	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5
1	√	√	√	√	√
2	√	√	√		
3		√	√		√
4		√			√
5			√		√
6				√	√
7				√	
8					√

“√” means that the document contains the corresponding term

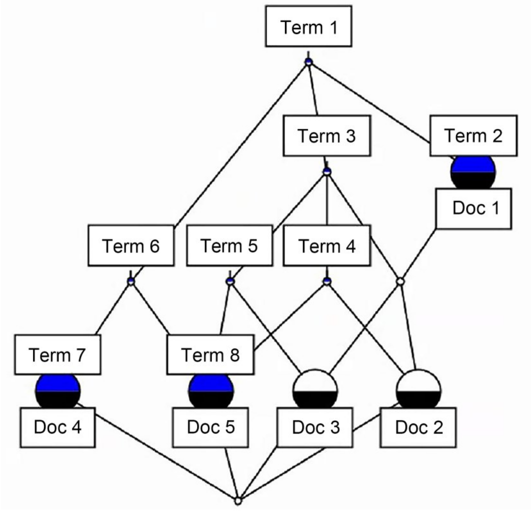

Fig. 1 Hasse graph of the concept lattice that corresponds to the relations in Table 1 (References to color refer to the online version of this figure)

Ontology is a formal and explicit specification of a shared conceptualization (Gruber, 1995). This means that ontology explicitly defines the rich relations between concepts. The key component in the ontology is the hierarchy of concepts. Domain ontology is a formal description of the background knowledge in a specific field. In a semantic sense, the identification of hypernymy or hyponymy relations between words is mandatory for building a hierarchy of concepts. In this study we build a domain ontology based on the topic of rainstorm disaster, where the hierarchy of concepts is built based on hyponymy in the concept lattice and the expert experience. Also, for the ontology construction, readers can refer to WordNet and Chinese Classified thesaurus (http://cct.nlc.cn/login.aspx). In the following, we give two definitions based on the domain ontology:

Definition 1 (Global ontology) Global ontology provides a relatively complete semantic model, which includes all related entities (concepts) in a specified domain and basic knowledge hierarchy among entities for sharing and reusing characteristics (Fig. 2).

Definition 2 (Local ontology) Local ontology is a special domain ontology whose topic is a reuse of a concept separated from the global ontology, and consists of all sub-concepts and the hierarchy relations under this concept of the global ontology (Fig. 2). A local ontology can be reused by adding, deleting, modifying, and other operations.

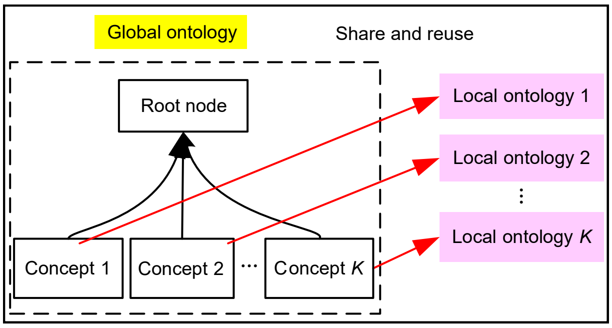

Fig. 2 Schematic of the global ontology and local ontology

Example 2 We construct a global ontology (shown in Fig. S1 in the supplementary materials) on the topic of rainstorm disaster according to the above mentioned method. The constructed global ontology includes 50 concepts and a six-level hierarchical structure. We use the concepts of disaster management, secondary disaster, and rainstorm level, separated from the rainstorm disaster global ontology, to build three local ontologies.

The above approach for building an ontology of the topic rainstorm disaster unifies and structures the description of the domain background knowledge, including prediction, early warning, disaster level, and related emergency management knowledge. In this paper we propose the idea of constructing a global ontology and generating some local ontologies separated from the global ontology for the first time, which makes full use of the ontologies to describe the topics. In this study, the main aim of constructing ontology is to calculate the concept semantic similarity in crawling, while the comprehensive applications of the global ontology and local ontology in the crawling process are to make the webpages be fully analyzed to reduce the omission of topic-relevant webpages and effectively prevent the topic drifting problem.

3 Concept semantic similarity calculation based on ontology

In the ontology structure, the five attribute relationships between two concepts C_1 and C_2 can effectively quantify the similarity of concepts, including semantic distance ($\mathrm{IF}_{\mathrm{Dis}}$), concept density ($\mathrm{IF}_{\mathrm{Den}}$), concept depth ($\mathrm{IF}_{\mathrm{Dep}}$), concept coincidence degree ($\mathrm{IF}_{\mathrm{Coi}}$), and concept semantic relationship ($\mathrm{IF}_{\mathrm{Rel}}$).

The definitions of these five attribute relationships can be found in Dong et al. (2020).

Based on the above five factors, the semantic similarity Sem(C_1, C_2) between concepts C_1 and C_2 is calculated as follows:

$$\begin{aligned}\mathrm{Sem}(C_1, C_2) = a \cdot \mathrm{IF}_{\mathrm{Dis}} + b \cdot \mathrm{IF}_{\mathrm{Den}} + c \cdot \mathrm{IF}_{\mathrm{Dep}} \\ + d \cdot \mathrm{IF}_{\mathrm{Coi}} + e \cdot \mathrm{IF}_{\mathrm{Rel}},\end{aligned} \tag{1}$$

where the adjustment factors satisfy $a+b+c+d+e=1$. In this study, they are set as $a=0.7$, $b=0.04$, $c=0.11$, $d=0.03$, and $e=0.12$, according to the results of many experiments. Suppose that **GTK**=(gtk$_1$, gtk$_2$, ..., gtk$_r$) indicates the vector of the topic word sets, where r represents the number of topic words in the global ontology. $\boldsymbol{W}_{\mathrm{GTK}}$ is the semantic weight vector of the topic words corresponding to the global ontology, and w_{gtk_i} indicates the weight of the i^{th} topic word gtk$_i$. Thus, if the topic of a global ontology is GC, the method of calculating $\boldsymbol{W}_{\mathrm{GTK}}$ is as follows:

$$\begin{aligned}\boldsymbol{W}_{\mathrm{GTK}} &= \left(w_{\mathrm{gtk}_1}, w_{\mathrm{gtk}_2}, \ldots, w_{\mathrm{gtk}_r}\right) \\ &= \left(\mathrm{Sem}\left(\mathrm{GC}, \mathrm{gtk}_1\right), \mathrm{Sem}\left(\mathrm{GC}, \mathrm{gtk}_2\right), \ldots, \right. \\ &\quad \left. \mathrm{Sem}\left(\mathrm{GC}, \mathrm{gtk}_r\right)\right).\end{aligned} \tag{2}$$

Suppose that the topics of k local ontologies are LC$_1$, LC$_2$, ..., LC$_k$. $\boldsymbol{W}_{\mathrm{LTK}}$ indicates the semantic weight vector of the topic word sets corresponding to k local ontologies. $\boldsymbol{W}_{\mathrm{LTK}_i}$ indicates the semantic weight vector of the topic words corresponding to the i^{th} local ontology. $w_{\mathrm{ltk}_j^i}$ indicates the weight of the j^{th} topic word in the i^{th} local ontology. **LTK**=(LTK$_1$, LTK$_2$, ..., LTK$_k$) indicates a vector of topic word sets corresponding to k local ontologies. LTK$_i$={ltk$_1^i$, ltk$_2^i$, ..., ltk$_{N_i}^i$} indicates the topic word set of the i^{th} local ontology, and N_i indicates the number of topic words in the i^{th} local ontology. Thus, the method of calculating $\boldsymbol{W}_{\mathrm{LTK}}$ is as follows:

$$\begin{aligned}\boldsymbol{W}_{\mathrm{LTK}} &= \left(\boldsymbol{W}_{\mathrm{LTK}_1}, \boldsymbol{W}_{\mathrm{LTK}_2}, \ldots, \boldsymbol{W}_{\mathrm{LTK}_k}\right) \\ &= \left(\left(w_{\mathrm{ltk}_1^1}, w_{\mathrm{ltk}_2^1}, \ldots, w_{\mathrm{ltk}_{N_1}^1}\right), \left(w_{\mathrm{ltk}_1^2}, w_{\mathrm{ltk}_2^2}, \ldots, w_{\mathrm{ltk}_{N_2}^2}\right), \ldots, \right. \\ &\quad \left. \left(w_{\mathrm{ltk}_1^k}, w_{\mathrm{ltk}_2^k}, \ldots, w_{\mathrm{ltk}_{N_k}^k}\right)\right) \\ &= \left(\left(\mathrm{Sem}\left(\mathrm{LC}_1, \mathrm{ltk}_1^1\right), \mathrm{Sem}\left(\mathrm{LC}_1, \mathrm{ltk}_2^1\right), \ldots, \right.\right. \\ &\quad \left. \mathrm{Sem}\left(\mathrm{LC}_1, \mathrm{ltk}_{N_1}^1\right)\right), \\ &\quad \left(\mathrm{Sem}\left(\mathrm{LC}_2, \mathrm{ltk}_1^2\right), \mathrm{Sem}\left(\mathrm{LC}_2, \mathrm{ltk}_2^2\right), \ldots, \right. \\ &\quad \left. \mathrm{Sem}\left(\mathrm{LC}_2, \mathrm{ltk}_{N_2}^2\right)\right), \ldots, \\ &\quad \left(\mathrm{Sem}\left(\mathrm{LC}_k, \mathrm{ltk}_1^k\right), \mathrm{Sem}\left(\mathrm{LC}_k, \mathrm{ltk}_2^k\right), \ldots, \right. \\ &\quad \left.\left. \mathrm{Sem}\left(\mathrm{LC}_k, \mathrm{ltk}_{N_k}^k\right)\right)\right).\end{aligned} \tag{3}$$

4 Comprehensive topic relevance calculation for the focused crawler

This section introduces the topic-relevance calculation methods. We use a vector space model (VSM) to calculate the topic relevance of webpage text (Farag et al., 2018; Jia et al., 2021), and propose a CPEM for predicting the priority (topic relevance) of the unvisited hyperlink.

4.1 Topic relevance calculation of webpages

In a search engine, the regular expression (RE) method (Colazzo et al., 2013) uses some metacharacters to compose information that meets certain rules. The process of using RE to match words directly is very slow. Therefore, in this study we use a word segmentation tool with open source, IK Analzyer (IK) (Wang and Meng, 2014), to process word segmentation by loading custom extending dictionaries and deactivating dictionaries.

A webpage is an HTML file composed of many elements and tags (Patel and Schmidt, 2011). The topic word appears in different tags with different influences. We assign different weights w_j to different labels (Liu JF et al., 2019a). We choose main tags from HTML and divide them into five groups, as shown in Table 2.

We now describe the vectorization of webpage text. First, remove all noise in the webpage text. Then, after matching or segmenting the extracted content, count the term frequency (TF) of each topic word. The webpage text may be represented as a TF vector $\boldsymbol{D}_{\mathrm{TF}}$=(TF$_1$, TF$_2$, ..., TF$_n$), where n is the number of topic words. Considering the weight of different tags extracted from HTML, the webpage text can be represented as a TF vector $\boldsymbol{D}_{\mathrm{TF}}$=((TF$_{1,1}$, TF$_{1,2}$, ..., TF$_{1,J}$), (TF$_{2,1}$, TF$_{2,2}$, ..., TF$_{2,J}$), ..., (TF$_{n,1}$, TF$_{n,2}$, ..., TF$_{n,J}$)), where TF$_{i,j}$ represents the TF of the i^{th} topic word in

Table 2 Division of labels and their weights

Group No.	Label	Meaning	Weight
1	<title>, <keyword>, <description>, <h1>	Title, keyword, description, first-level headline, respectively	2.0
2	<h2>, <h3>	Second- and third-level headline, respectively	1.5
3	<h4>, <h5>, <h6>, <strong>	Fourth-, fifth-, and sixth-level headline, bold text, respectively	1.2
4	<p>, <td>, <li>	Body information	1.0
5	Other labels	Non-body information	0.2

the j^{th} position (group) of the webpage text, and J represents the size of tag groups (here, J=5). Generally, if a feature word appears frequently in a text (i.e., its TF is large), this feature word should be important and has good classification ability. To take full account of the importance of tag information, we use the following model to calculate the weight w_{dk_j} of the i^{th} topic word in the webpage feature set DK={dk$_1$, dk$_2$, ..., dk$_n$}:

$$w_{\text{dk}_i} = \sum_{j=1}^{J}\left(\text{tf}_{i,j} w_j\right) = \sum_{j=1}^{J}\left(\frac{\text{TF}_{i,j}}{\max \text{TF}_{i,j}} w_j\right), \tag{4}$$

where tf$_{i,j}$ represents the normalized TF of the i^{th} topic word at the j^{th} position (group) of the webpage text, max TF$_{i,j}$ represents the maximum TF of the i^{th} topic word occurring at all positions, and w_j represents the weight of the j^{th} tag group.

Some previous focused crawler algorithms ignore the impact of semantics on crawlers, and consider only TF to calculate the topic relevance $R(P)$ of webpage P:

$$R(P) = \sum_{j=1}^{J}\sum_{i=1}^{n} \text{TF}_{i,j} w_j. \tag{5}$$

Here, we use the VSM method to calculate the topic relevance $R(P)$ of webpage P:

$$\begin{aligned} R(P) &= \text{Sim}(\text{TK}, \text{DK}) \\ &= \frac{\boldsymbol{W}_{\text{TK}}\boldsymbol{W}_{\text{DK}}}{\left\|\boldsymbol{W}_{\text{TK}}\right\|\left\|\boldsymbol{W}_{\text{DK}}\right\|} = \frac{\sum_{i=1}^{n} w_{\text{tk}_i} w_{\text{dk}_i}}{\sqrt{\sum_{i=1}^{n} w_{\text{tk}_i}^2}\sqrt{\sum_{i=1}^{n} w_{\text{dk}_i}^2}}, \end{aligned} \tag{6}$$

where $\boldsymbol{W}_{\text{TK}} = \left(w_{\text{tk}_1}, w_{\text{tk}_2}, ..., w_{\text{tk}_n}\right)$ represents the semantic weight vector of topic words, with w_{tk_i} indicating the weight of the i^{th} topic word in the topic word set TK={tk$_1$, tk$_2$, ..., tk$_n$}, and $\boldsymbol{W}_{\text{DK}} = \left(w_{\text{dk}_1}, w_{\text{dk}_2}, ..., w_{\text{dk}_n}\right)$ represents the feature weight vector of a webpage, with w_{dk_i} indicating the weight of the i^{th} topic word in the webpage feature set DK={dk$_1$, dk$_2$, ..., dk$_n$}.

VSM is a well-known measure of cosine and transforms a language problem into a mathematical problem. The cosine similarity between two vectors is considered as the similarity of the text related to the given topic. When the angle between two vectors is equal to 0°, the relevance between them is maximum and equals 1, indicating that they are the most relevant. When the angle is equal to 90°, the relevance is minimum and equals 0, indicating that they are irrelevant. We set threshold σ to determine whether the webpage is related to the topic. If $R(P)$ is greater than σ, webpage P is considered to be related to the topic; otherwise, webpage P is regarded as irrelevant to the topic.

4.2 Improved webpage PR calculation

For webpage P, the traditional method of calculating the PR value is as follows:

$$\text{PR}(P) = (1-d) + d\sum_{i=1}^{m}\frac{\text{PR}(P_i)}{C(P_i)}, \tag{7}$$

where d=0.85, m represents the number of in-links of P in the crawled webpage set, PR(P_i) represents the PR value of the i^{th} in-link of webpage P, and $C(P_i)$ represents the number of out-links of the i^{th} in-link of webpage P.

In Eq. (7), the larger the number of in-links of P and the higher the average importance of the out-link webpage, the higher the importance of P (i.e., the higher the PR value). This importance has nothing to do with the topic and easily leads to topic drifting. To overcome this topic drifting problem, the topic relevance of anchor text is introduced in the calculation of the PR value. In Eq. (8), except for the number of in-links and the average importance of the out-link webpages, the higher the topic relevance of the in-link anchor text, the higher the importance of P (i.e., the larger the PR value).

$$\mathrm{PR}(P)=(1-d)+d\sum_{i=1}^{m}\left[\frac{\mathrm{PR}(P_i)}{C(P_i)}\left(1+\omega R(A_i)\right)\right],\tag{8}$$

where ω is the adjustment factor and is set to 0.6, and $R(A_i)$ represents the topic relevance of anchor text A_i of the i^{th} in-link of P (Section 4.3). Assume all crawled webpages as the entire Internet. Therefore, the PR value of each webpage is constantly updated.

4.3 Comprehensive priority evaluation of hyperlinks

The anchor text usually has only a few words or phrases, but can clearly describe the main idea of the webpage to which the hyperlink points. If a topic word frequently appears in a certain anchor text and rarely appears in the other anchor texts, this topic word should be important and has a strong distinguishing ability. According to TF×IDF (IDF is short for inverse document frequency), the calculation formula of the weight w_{ak_i} of the i^{th} topic word in an anchor text is as follows:

$$w_{\mathrm{ak}_i}=\mathrm{TF}_i\times\mathrm{IDF}_i=\frac{f_i}{\sum_{m=1}^{n}f_m}\log_S\left(\frac{N}{N_i}+0.01\right),\tag{9}$$

where f_i represents the TF of the i^{th} topic word in the anchor text, n is the number of topic words, N represents the number of crawled webpages, N_i represents the number of crawled webpages containing the i^{th} topic word of this anchor text, and $s>1$. By considering the cosine between $\boldsymbol{W}_{\mathrm{TK}}$ and $\boldsymbol{W}_{\mathrm{AK}}$, the topic relevance $R(A_l)$ of anchor text A_l is computed by

$$\begin{aligned}R(A_l)=\mathrm{Sim}(\mathrm{TK},\mathrm{AK})&=\frac{\boldsymbol{W}_{\mathrm{TK}}\boldsymbol{W}_{\mathrm{AK}}}{\|\boldsymbol{W}_{\mathrm{TK}}\|\|\boldsymbol{W}_{\mathrm{AK}}\|}\\&=\frac{\sum_{i=1}^{n}w_{\mathrm{tk}_i}w_{\mathrm{ak}_i}}{\sqrt{\sum_{i=1}^{n}w_{\mathrm{tk}_i}^2}\sqrt{\sum_{i=1}^{n}w_{\mathrm{ak}_i}^2}},\end{aligned}\tag{10}$$

where $\boldsymbol{W}_{\mathrm{AK}}=\left(w_{\mathrm{ak}_1},w_{\mathrm{ak}_2},\ldots,w_{\mathrm{ak}_n}\right)$ represents the feature weight vector of the anchor text, and w_{ak_i} represents the weight of the i^{th} topic word in the anchor text feature set AK=$\{\mathrm{ak}_1, \mathrm{ak}_2, \ldots, \mathrm{ak}_n\}$.

In addition, the relevance of the next webpage P_l to which hyperlink l points affects the priority of hyperlink l. Let $\boldsymbol{W}_{\mathrm{UK}}=\left(w_{\mathrm{uk}_1},w_{\mathrm{uk}_2},\ldots,w_{\mathrm{uk}_n}\right)$ denote the feature weight vector of the next webpage P_l to which hyperlink l points, where w_{uk_i} represents the weight of the i^{th} topic word in webpage P_l, and UK= $\{\mathrm{uk}_1, \mathrm{uk}_2, \ldots, \mathrm{uk}_n\}$ represents the webpage feature set of P_l. According to Eq. (6), topic relevance of P_l is as follows:

$$R(P_l)=\mathrm{Sim}(\mathrm{TK},\mathrm{UK}).\tag{11}$$

To evaluate the topic relevance (or priority) of the unvisited hyperlink l, we propose a CPEM which involves the topic relevance $R(A_l)$ of the anchor text A_l of link l, the sum of topic relevance $R(P_i)$ of the webpages at which link l is located, the PR(P_l) value of webpage P_l to which hyperlink l points, and the topic relevance $R(P_l)$ of webpage P_l. The comprehensive priority Priority(l) of the unvisited hyperlink l is

$$\begin{aligned}\mathrm{Priority}(l)=\alpha R(A_l)+\beta\sum_{i=1}^{m}R(P_i)\\+\gamma\mathrm{PR}(P_l)+\theta R(P_l),\end{aligned}\tag{12}$$

where $\alpha+\beta+\gamma+\theta=1$.

We predict the priority of each hyperlink by Eq. (12) and set the threshold to η for filtering unvisited hyperlinks. If Priority(l)>η, we add link l into an ordered link-waiting queue Q_{w} according to the priority; otherwise, we discard it. Generally, an OPS strategy is applied to select next hyperlink to crawl from Q_{w}. However, it is easy for this strategy to cause the search to fall into local optima. To avoid this, we introduce an SA algorithm to select the next hyperlink by optimizing maximum Priority(l) to enhance the global search performance of focused crawlers.

5 Simulated annealing based focused crawler strategy

In this section, we first introduce an SA algorithm and give its improved version for selecting the next link in the focused crawler. Then, the framework of the focused crawler strategy based on the SA (FCSA) algorithm is proposed. Finally, by incorporating MFSLG and CPEM into FCSA, the focused crawler strategy based on the ontology and SA (FCOSA) is presented.

5.1 Simulated annealing algorithm based link selection

The SA algorithm has been widely used in combinatorial optimization problems (Liu JF et al., 2010). The idea of SA originates from the annealing process of solid matter in physics. It is a random optimization algorithm and generally starts from a high initial temperature. It iterates from an initial solution and updates the solution by an effective neighborhood strategy. For the newly generated solution, it uses the Metropolis sampling criterion to determine whether it is accepted. With the stability of sampling, the system gradually begins to cool down. The above process is repeated until the algorithm obtains the optimal solution to the problem or reaches the preset minimum temperature.

We introduce the SA procedure for selecting the next link in the focused crawler. This gives the suboptimal links the opportunity to be selected and prevents the crawler from falling into local traps. At the same time, it helps the crawler extend the search range and find better retrieval paths by traversing the tunnel. The detailed steps of the SA process for selecting a link are outlined in Algorithm 1. The algorithm begins from the header-link from the link-waiting queue Q_w, and selects the next link from Q_w by the roulette method. For the generation of seed URLs of Q_w, there are three ways (Liu JF et al., 2019a): (1) manual method—collecting the seed URLs from experts in the field; (2) auto-generated method—entering the specified topic words, for example, rainstorm disaster and disaster management, into regular search engines (e.g., Baidu and Google) in sequence, and selecting the URLs listed in the preceding pages as seed URLs; (3) mixed mode—combining the manual method and auto-generated method. We use the mixed mode to generate seed URLs of Q_w. In the executing process of the SA algorithm, there are some important parameters which need to be set, including the initial annealing temperature T, the controlled annealing speed parameter C, and the number of inner cycles M during the annealing process. The parameters T, C, and M are all empirical values. The parameter C controls the rate of temperature dropping. When the temperature T tends to 0, the possibility of accepting suboptimal links also tends to 0.

Algorithm 1 SA(Q_w)

```
Input: $Q_w$
Output: a link
1: Set $T$=1, $M$=10, and $C$=0.9
2: Choose the header-link from $Q_w$ and mark it as current
3: Set $q$=1
4: Select randomly a link from $Q_w$ by the roulette method,
   and mark it as next
5: If random[0, 1]<exp[(Priority(next)−Priority(current))/$T$]
   then
     Accept next for the next link, and let current=next
   Else
     Do not accept next for the next link, and keep current
     unchanged
   End if
6: If $q$>$M$ then go to step 7
    Else let $q$=$q$+1, and go to step 4
    End if
7: Let $T$=$C$×$T$
8: If $T$⩽0.01 then
     Output current
   Else go to step 3
   End if
```

To shorten the runtime of the algorithm, we introduce the target completion rate (com-rate) to adjust the temperature. Suppose that the target of the focused crawler is to download 15 000 webpages from the Internet, and that the number of current downloaded webpages is DP. Thus, the target completion rate is defined as com-rate=DP/15 000. With the increase of the downloaded DP, the initial temperature of the SA algorithm is reduced to accelerate the convergence. The improved SA (ISA) for selecting a link is obtained by modifying step 1 in the above SA algorithm by the following steps: Compute the target completion rate com-rate. If com-rate=0, set T=1; otherwise, set T=T× com-rate. Set M=10 and C=0.9.

5.2 Simulated annealing strategy based focused crawler

This subsection introduces the specific process of FCSA. FCSA starts with the first link ordered in

Q_{w}. For each obtained webpage, we pre-treat it and extract the webpage text features and subsequently calculate the relevance of the webpage by Eq. (6) to judge whether the obtained webpage is related to the topic. Then, all child-links included in the webpage are extracted and their comprehensive priorities are calculated by Eq. (12). Once the comprehensive priority of a child-link exceeds the threshold value η (η> 0), the child-link is inserted into Q_{w}. To overcome the defect of the greedy algorithms such as the OPS strategy, we use the SA algorithm (Section 5.1) to implement the link selection operation. The iterative steps of FCSA are outlined in Algorithm 2.

5.3 Focused crawler strategy based on the ontology and SA algorithm

In FCSA, the fact that the computation of topic relevance relies simply on TK and $\boldsymbol{W}_{\mathrm{TK}}$ has limitations in controlling the crawler search (TK and $\boldsymbol{W}_{\mathrm{TK}}$ consist of several simple topic words with semantic weights). To prevent the focused crawler from accessing irrelevant links and to lead it to access as many topic-relevant links as possible, we make full use of ontology features by constructing a global ontology and three local ontologies. This method will make topic search more extensive and can retrieve more topic-relevant hyperlinks.

FCOSA is divided into two groups. One is a focused crawler strategy based on only global ontology (FCOSA_G), and the other is a focused crawler strategy based on both the global ontology and local ontology (FCOSA_LG) (Algorithm 3). In FCOSA_LG, we use the local ontology for the first filtering, where LTK is used to compute the topic relevance of the page (to which each child-link points), and the global ontology for the second filtering, where GTK is used to compute the comprehensive priority of each saved child-link. The FCOSA_G algorithm is obtained by deleting step 9 in the FCOSA_LG algorithm. In both FCOSA_G and FCOSA_LG algorithms, we use the ISA strategy to select a link.

Algorithm 2 Focused crawler strategy based on the simulated annealing (FCSA)

```
Input: seed URLs
Output: downloaded webpages
1:  Add the seed URLs to $Q_w$. Set $\sigma$ and $\eta$. Let DP=0 and LP=0
2:  Select the first link ordered in $Q_w$, and mark it as Header-
    link. The webpage to which Header-link points is marked as
    Current-page
3:  Remove Header-link from $Q_w$ and download the Current-
    page
4:  Let DP=DP+1
5:  Remove the noise and extract tag information (Table 2)
    from the Current-page, and gain the feature vector DK
    of the Current-page
6:  Calculate the topic relevance $R$(Current-page) of the
    Current-page text according to Eq. (6)
7:  If $R$(Current-page)>$\sigma$ then
      Download the Current-page, and let LP=LP+1
    End if
8:  Extract all the child-links and the corresponding anchor
    texts from Current-page, and remove repeated links
    // For the irrelevant webpage, the purpose of extracting
    // all the child-links in the page is to provide a chance for
    // the crawler to pass through the tunnel
9:  For $i$=1 to $k$ do
    // $k$ is the size of the child-links
      Calculate the comprehensive priority of child-link$_i$
      according to Eq. (12)
        If Priority(child-link$_i$)>$\eta$ then
          Insert child-link$_i$ into $Q_w$
        Else give up child-link$_i$
        End if
    End For
10: Recalculate PR values of all downloaded webpages and
    update the comprehensive priority values of all links in $Q_w$
11: If $Q_w$ is not empty then
        Let $l$=SA($Q_w$)
        Insert link $l$ into the head of $Q_w$
    Else the algorithm ends
    End if
12: If DP<15 000 then
        Go to step 2
    Else the algorithm ends
    End if
```

6 Experimental results and discussion

To evaluate the performance of the proposed focused crawler algorithms FCSA, FCOSA_G, and FCOSA_LG, we implement BFS (Vidal et al., 2006), OPS (Rawat and Patil, 2013), the focused crawler based on the web space evolutionary (WSE) algorithm (Liu JF et al., 2019b), the focused crawler based on the improved tabu search (ITS) algorithm

Algorithm 3 FCOSA_LG

```
Input: seed URLs
Output: downloaded webpages
1:  Add seed URLs to $Q_w$. Set $\sigma$, $\varphi$, and $\eta$. Let DP=0 and LP=0
2:  Select the first link ordered in $Q_w$, and mark it as Header-link. The webpage to which Header-link points is marked as the Current-page
3:  Remove Header-link from $Q_w$ and download the Current-page
4:  Let DP=DP+1
5:  Remove the noise and extract tag information (Table 2) from the Current-page. Use IK for word segmentation and gain the feature vector DK of the Current-page
6:  Calculate the topic relevance $R$(Current-page) of the Current-page text according to Eq. (6)
7:  If $R$(Current-page)>$\sigma$ then
      Download the Current-page and let LP=LP+1
    End if
8:  Extract all the child-links and the corresponding anchor texts from the Current-page, and remove repeated links
    // Local ontology is used to implement the first filtering
    // of child-links
9:  For $i$=1 to $k_1$ do
    // $k_1$ is the size of the child-links
      For $j$=1 to $k_2$ do
      // $k_2$ is the number of local ontologies, and $k_2$=3 in this
      // study
        Calculate the topic relevance $R_{i,j}$ of child-link$_i$ based on the $j^{th}$ local ontology according to Eq. (11):
        $R_{i,j}$= Sim(LTK$_j$, UK)
        If $R_{i,j}\geqslant\varphi$ then
        // $\varphi$ is a positive parameter
            Save child-link$_i$
        break
        Else if $R_{ij}<\varphi$ and $j=k_2$
            Discard child-link$_i$
        End if
      End for
    End for
    // Global ontology is used to implement the second fil-
    // tering of the saved child-links
10: For $j$=1 to $k_3$ do
    // $k_3$ is the number of the saved child-links
      Calculate the comprehensive priority of the child-link$_j$ according to Eq. (12), where TK is replaced by GTK=(gtk$_1$, gtk$_2$, ..., gtk$_r$)
      If Priority(child-link$_j$)>$\eta$ then
      // $\eta$ is a positive parameter
        Insert child-link$_j$ into $Q_w$
      Else give up child-link$_j$
      End if
    End for
11: Recalculate PR values of all the downloaded webpages and update the comprehensive priority values of all links in $Q_w$
12: If $Q_w$ is not empty then
        Let $l$=ISA ($Q_w$)
        // Return link $l$
        Insert link $l$ into the head of $Q_w$
    Else the algorithm ends
    End if
13: If DP<15 000 then
        Go to step 2
    Else the algorithm ends
    End if
```

(Liu JF et al., 2020), the focused crawler based on the ontology and ITS algorithm (On-ITS) (Liu JF et al., 2020), and three algorithms proposed in this study. All algorithms are compiled in the Java language and run on a personal computer with Intel Pentium G3260, 3.30 GHz processor, and 4.0 GB RAM. We execute a series of experimental tests and analyze the computational results.

6.1 Performance metric indices

The performance metric indices of crawler algorithms generally include recall rate (Recall) and accuracy (Accuracy):

$$\text{Recall} = \frac{\text{LP}}{\text{TP}}, \tag{13}$$

$$\text{Accuracy} = \frac{\text{LP}}{\text{DP}}, \tag{14}$$

where TP represents the total number of topic-relevant webpages in the whole Internet, LP represents the number of downloaded topic-relevant webpages, and DP represents the total number of downloaded webpages. Since it is difficult to count the total number of topic-relevant webpages in the whole Internet, we choose Accuracy as the standard for comparison.

In addition, we use the average relevance (AR) and standard deviation (SD) of the downloaded webpages and the downloaded topic-relevant webpages to analyze the results of the algorithms. AR and SD of the downloaded topic-relevant webpages are calculated using Eqs. (15) and (16), respectively. AR and SD of the downloaded webpages are calculated using Eqs. (17) and (18), respectively.

$$\mathrm{AR_{LP}} = \frac{1}{\mathrm{LP}}\sum_{i=1}^{\mathrm{LP}} R(P_i),\ R(P_i) > \sigma, \tag{15}$$

$$\mathrm{SD_{LP}} = \sqrt{\frac{1}{\mathrm{LP}}\sum_{i=1}^{\mathrm{LP}}\left(R(P_i) - \mathrm{AR_{LP}}\right)^2},\ R(P_i) > \sigma, \tag{16}$$

$$\mathrm{AR_{DP}} = \frac{1}{\mathrm{DP}}\sum_{i=1}^{\mathrm{DP}} R(P_i), \tag{17}$$

$$\mathrm{SD_{DP}} = \sqrt{\frac{1}{\mathrm{DP}}\sum_{i=1}^{\mathrm{DP}}\left(R(P_i) - \mathrm{AR_{DP}}\right)^2}, \tag{18}$$

where $\mathrm{AR_{LP}}$ denotes the average relevance of all downloaded topic-relevant webpages LP, and $\mathrm{SD_{LP}}$ is the standard deviation of all downloaded topic-relevant webpages compared to $\mathrm{AR_{LP}}$ and is used to measure the spread of the topic relevance of all downloaded topic-relevant webpages LP. $\mathrm{AR_{DP}}$ and $\mathrm{SD_{DP}}$ have similar meanings.

6.2 Experimental results of different algorithms

As mentioned, we use a mixed mode to generate seed URLs (Section 5.1). The total number of seed URLs (shown in Table S1 in the supplementary materials) is 30. The settings of experimental parameters or strategies for different algorithms are listed in Table 3. In particular, we use the same TK to evaluate the relevance of webpages, and the pre-defined threshold σ is set to 0.7 (Liu WJ and Du, 2014) in all experiments. The threshold σ is used to measure whether the webpage is a topic-relevant page. In addition, the experimental environments and initialization conditions of different algorithms are the same. The number of retrieved webpages starts with 100, and then 500, and progressively is increased by 500. All algorithms end when DP reaches 15 000 or Q_w is empty.

Table 4 displays the results of Accuracy, LP, AR, and SD when DP reaches 1000, 5000, 10 000, and 15 000. Table 4 shows that when DP reaches 1000, OPS finds the optimal results of Accuracy, LP, $\mathrm{AR_{LP}}$, and $\mathrm{SD_{LP}}$, ITS achieves the optimal value of $\mathrm{AR_{DP}}$, and FCOSA_LG finds the lowest $\mathrm{SD_{DP}}$. When DP reaches 5000, OPS finds the optimal values of Accuracy, LP, and $\mathrm{SD_{LP}}$, while FCOSA_LG obtains the optimal results of $\mathrm{AR_{LP}}$, $\mathrm{AR_{DP}}$, and $\mathrm{SD_{DP}}$. When DP reaches 10 000, FCOSA_LG obtains the highest Accuracy, LP, $\mathrm{AR_{LP}}$, $\mathrm{AR_{DP}}$, and the least $\mathrm{SD_{DP}}$. For the FCOSA_LG algorithm, when DP reaches 15 000, its $\mathrm{AR_{LP}}$ is greater than 0.8 and $\mathrm{AR_{DP}}$ exceeds 0.75. In addition, FCOSA_LG always has the lowest $\mathrm{SD_{DP}}$, while the OPS algorithm has the smallest $\mathrm{SD_{LP}}$. This can be explained by the fact that FCOSA_LG always extracts child-links whose comprehensive priority exceeds the preset threshold η, and that OPS always downloads the most relevant webpages in the process of crawling.

The results of Accuracy and LP reflect the ability of the crawlers to retrieve absolute quantities of topic-relevant pages. When DP reaches 15 000, Accuracy of the FCOSA_LG algorithm is about 0.03 higher than those of WSE and On-ITS, about 0.13 higher than that of ITS, about 0.18 higher than that of FCSA, about 0.1 higher than that of FCOSA_G, and far

Table 3 Experimental strategies and parameters of different algorithms

Strategy	BFS	OPS	WSE	ITS	On-ITS	FCSA	FCOSA_G	FCOSA_LG
SA	×	×	×	×	×	√	×	×
ISA	×	×	×	×	×	×	√	√
MFSLG	×	×	×	×	×	×	×	√
CPEM	×	×	×	×	×	√	√	√
Global ontology	×	×	×	×	√	×	√	√
Local ontology	×	×	×	×	×	×	×	√
Parameter	**BFS**	**OPS**	**WSE**	**ITS**	**On-ITS**	**FCSA**	**FCOSA_G**	**FCOSA_LG**
T						1.00	1.00	1.00
C						0.90	0.90	0.90
M						10	10	10
φ								0.15
η					0.15		0.15	0.15
σ	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70

“√” means that the algorithm uses the corresponding strategy, and “×” means that the algorithm does not use the corresponding strategy

Table 4 Experimental results of different algorithms about evaluation indices of Accuracy, LP, AR_{LP}, SD_{LP}, AR_{DP}, SD_{DP}, and retrieval time and when DP reaches 1000, 5000, 10 000, and 15 000

DP	Algorithm	Accuracy	LP	AR_{LP}	SD_{LP}	AR_{DP}	SD_{DP}	Time (h)
1000	BFS	0.1840	184	0.7760	0.0538	0.4122	0.2662	
	OPS	**0.7440**	**744**	**0.7769**	**0.0342**	0.6007	0.2258	
	WSE	0.4020	402			0.6500	0.1620	
	ITS	0.6960	696			**0.7027**	0.1830	
	On-ITS	0.7020	702			0.6982	0.1624	
	FCSA	0.6010	601	0.7498	0.0367	0.5819	0.2956	
	FCOSA_G	0.7140	714	0.7663	0.0651	0.6909	0.1359	
	FCOSA_LG	0.7100	710	0.7378	0.0644	0.6779	**0.1129**	
5000	BFS	0.1438	719	0.7723	0.0491	0.2856	0.2563	
	OPS	**0.7900**	**3950**	0.7782	**0.0274**	0.6736	0.1494	
	WSE	0.6130	3065			0.7000	0.1620	
	ITS	0.6580	3290			0.6577	0.1556	
	On-ITS	0.7000	3500			0.7076	0.1629	
	FCSA	0.6264	3132	0.7616	0.0449	0.5952	0.2365	
	FCOSA_G	0.7314	3657	0.7871	0.0633	0.7106	0.1478	
	FCOSA_LG	0.7620	3810	**0.7954**	0.0688	**0.7498**	**0.1199**	
10 000	BFS	0.0965	965	0.7776	0.0425	0.2927	0.2726	
	OPS	0.5376	5376	0.7784	**0.0321**	0.5716	0.2139	
	WSE	0.7000	7006			0.7250	0.1620	
	ITS	0.6600	6600			0.6436	0.2013	
	On-ITS	0.7010	7010			0.7266	0.1622	
	FCSA	0.6043	6043	0.7798	0.0472	0.6228	0.2424	
	FCOSA_G	0.7123	7123	0.7808	0.0643	0.6913	0.1693	
	FCOSA_LG	**0.7882**	**7882**	**0.8023**	0.0604	**0.7562**	**0.1287**	
15 000	BFS	0.0657	985	0.7788	0.0447	0.2262	0.2552	**8.54**
	OPS	0.4426	6639	0.7785	**0.0375**	0.5631	0.2020	9.12
	WSE	0.7330	11 002			0.7290	0.1600	12.23
	ITS	0.6364	9546			0.6627	0.1953	11.48
	On-ITS	0.7340	11 010			0.7295	0.1619	13.24
	FCSA	0.5817	8726	0.7895	0.0462	0.6463	0.2475	11.16
	FCOSA_G	0.6693	10 040	0.7906	0.0644	0.6871	0.1677	12.55
	FCOSA_LG	**0.7653**	**11 479**	**0.8095**	0.0581	**0.7511**	**0.1462**	13.12

Best results are in bold

higher than those of BFS and OPS. As a whole, the FCOSA_LG crawling algorithm stabilizes in a higher Accuracy range, and is superior to the other algorithms. The results of AR_{LP}, SD_{LP}, AR_{DP}, and SD_{DP} reflect the stability of crawlers. When DP reaches 15 000, compared with FCSA, FCOSA_G, and the other algorithms, the FCOSA_LG algorithm obtains the best results of AR_{LP}, AR_{DP}, and SD_{DP}. This indicates that the proposed FCOSA_LG algorithm is more conducive to the global retrieval of topic-relevant webpages. This also proves that the combination of the SA algorithm and the MFSLG strategy to guide crawlers to filter hyperlinks can improve the stability of focused crawlers. Table 4 lists the runtime of different algorithms when DP=15 000. As can be seen, BFS and OPS have a slightly short retrieval time, but their results are far worse than those of the proposed crawlers.

In addition, the Friedman test (Derrac et al., 2011) is a non-parametric statistical test and is commonly used to compare the performances of two or more algorithms. Here, when DP=15 000, the results obtained by eight different algorithms for the three representative indices Accuracy, $\mathrm{AR_{DP}}$, and $\mathrm{SD_{DP}}$ are converted to ranks. The best performing algorithm for each index should have the rank of 1, the second best ranks 2, and so on. Table 5 depicts the ranks of eight algorithms according to these three evaluation indices by the Friedman test. As can be seen, FCOSA_LG algorithm with the minimum average for three indices is the best performing algorithm of the eight algorithms, On-ITS ranks second, WSE ranks third, and BFS is the worst.

From Tables 4 and 5, it is not hard to see that on the whole, FCOSA_LG overmatches FCOSA_G and FCOSA_G is superior to FCSA. This further verifies the effectiveness of the improved strategies in FCOSA_LG and FCOSA_G. For selecting topic-relevant webpages, the multiple-filtering strategy based on the global ontology and local ontology in FCOSA_LG outperforms the simple filtering strategy based on the global ontology in FCOSA_G. For topic description, the domain ontology topic model used in FCOSA_LG and FCOSA_G outperforms the topic model in FCSA.

6.3 Experimental results of FCOSA_LG under different parameters

The parameter values in the focused crawlers are important. We run FCOSA_LG algorithm with different parameters to analyze the experimental results. FCOSA_LG algorithm has three important parameters, σ, φ, and η, which denote the threshold of the topic relevance of the webpage, the threshold of the topic relevance of the child-link based on the local ontology, and the threshold of the comprehensive priority of the hyperlink, respectively. To test the effects of the thresholds σ, φ, and η on the experiments, we select some representative values to execute FCOSA_LG algorithm. The algorithm is run 20 times under each parameter independently.

The threshold φ is set to 0.05, 0.10, 0.15, 0.20, and 0.25, η is set to 0.10, 0.15, and 0.20, and σ is set to 0.5, 0.6, and 0.7. Table 6 records the Accuracy and LP of FCOSA_LG algorithm based on different thresholds. The larger the values of φ and η, the better the Accuracy achieved by the algorithm. In these three cases, when both φ and η are set to 0.15, the algorithm can download 15000 pages and the Accuracy and LP are the best. If the values of thresholds are too large, their filtering capacities are excessive and the algorithm ends prematurely. In this study, φ and η are both set to 0.15. From Table 6, it can be seen that when the threshold σ is smaller, FCOSA_LG algorithm can obtain better Accuracy and higher LP. However, some actual irrelevant webpages are counted in LP. Therefore, in this study, σ is fixed to 0.7 according the threshold value in Liu WJ and Du (2014).

In addition, five independent experimental results of FCOSA_LG algorithm when σ=0.7, φ=0.15, η=0.15, and DP=15000 are shown in Table 7. The average Accuracy obtained from five independent crawlers is 0.7565. $\mathrm{AR_{LP}}$ and $\mathrm{AR_{DP}}$ obtained from the five independent runs have an average value of 0.8056 and 0.7235, respectively. The average values of $\mathrm{SD_{LP}}$ and $\mathrm{SD_{DP}}$ are low, i.e., 0.0374 and 0.1492, respectively. Through the analysis of the above experiments, when φ=0.15 and η=0.15, FCOSA_LG algorithm performs the best and has good stability.

7 Conclusions

Traditional approaches of crawlers generally pursue the maximum number of retrieved webpages, regardless of the content of the webpages. However, in most real-world searches, crawlers have an explicit target theme. It is also noticeable that most existing methods guide the search process using the domain

Table 5 Friedman test ranks of eight algorithms for the three representative evaluation indices of Accuracy, $\mathrm{AR_{DP}}$, and $\mathrm{SD_{DP}}$ when DP reaches 15000

Index	Rank							
	BFS	OPS	WSE	ITS	On-ITS	FCSA	FCOSA_G	FCOSA_LG
Accuracy	8	7	3	5	2	6	4	1
$\mathrm{AR_{DP}}$	8	7	3	5	2	6	4	1
$\mathrm{SD_{DP}}$	8	6	2	5	3	7	4	1
Average	8	6.67	2.67	5	2.33	6.33	4	**1**

Table 6 Computational results obtained by FCOSA_LG algorithm with different threshold sizes of σ, φ, and η when DP=15 000

φ	(Accuracy, LP)								
	σ=0.5			0.6			0.7		
	η=0.10	0.15	0.20	0.10	0.15	0.20	0.10	0.15	0.20
0.05	(0.4845, 7268)	(0.9050, 13 575)	–	(0.4085, 6127)	(0.8055, 12 082)	–	(0.2850, 4275)	(0.6798, 10 197)	–
0.10	(0.5305, 7958)	(0.9117, 13 675)	–	(0.4469, 6704)	(0.8490, 12 735)	–	(0.3058, 4587)	(0.7297, 10 945)	–
0.15	(0.6055, 9083)	(**0.9280**, **13 920**)	–	(0.4605, 6907)	(**0.8555**, **12 832**)	–	(0.3283, 4924)	(**0.7653**, **11 479**)	–
0.20	(0.7354, 11 031)	–	–	(0.6358, 9537)	–	–	(0.5543, 8314)	–	–
0.25	(0.8518, 12 777)	–	–	(0.7426, 11 139)	–	–	(0.6615, 9923)	–	–

"–" means that the algorithm has ended prematurely when DP has not reached 15 000

Table 7 Experimental results of FCOSA_LG algorithm over five independent times when σ=0.7, φ=0.15, η=0.15, and DP=15 000

No.	Accuracy	LP	$\mathrm{AR}_{\mathrm{LP}}$	$\mathrm{SD}_{\mathrm{LP}}$	$\mathrm{AR}_{\mathrm{DP}}$	$\mathrm{SD}_{\mathrm{DP}}$
1	0.7498	11 247	0.8029	0.0311	0.7125	0.1515
2	0.7516	11 274	0.8183	0.0389	0.7187	0.1489
3	0.7630	11 445	0.7915	0.0286	0.7110	0.1476
4	0.7528	11 292	0.8060	0.0301	0.7244	0.1517
5	0.7653	11 479	0.8095	0.0581	0.7511	0.1462
Average	0.7565	11 347	0.8056	0.0374	0.7235	0.1492

conceptual hierarchy to describe the topic benchmark model and the greedy strategy to control the search direction in the Web. In contrast, we proposed a new strategy based on a focused crawler which is related to the topic of rainstorm disasters and gives more weight to the judgment of webpage topic relevance. We constructed the ontology as the topic benchmark model, and converted the problem of evaluating hyperlinks into a single-objective optimization problem. We proposed three algorithms, FCSA, FCOSA_G, and FCOSA_LG, to direct the search of the focused crawlers. To improve the accuracy of the algorithms and prevent the phenomenon of topic drifting, we proposed a novel CPEM for unvisited hyperlinks and a novel multiple-filtering strategy based on MFSLG to find more topic-relevant webpages.

Experimental results on rainstorm disaster showed that the proposed FCSA, FCOSA_G, and FCOSA_LG algorithms are effective in implementing the focused crawler. The FCOSA_LG algorithm achieved state-of-the-art performance and was capable of finding more topic-relevant webpages. The crawler algorithms proposed here can effectively obtain relevant knowledge about rainstorm disasters from the network, and provide a reference plan for disaster warning and preventive measures. In addition, crawlers can promote the construction of ontology knowledge in the domain of rainstorm disasters.

The main challenges of the proposed focused crawlers based on ontology and the annealing strategy include automated ontology construction and the design of an annealing strategy. Although a semi-automated method of constructing ontology based on FCA was proposed in this paper, the automated ontology construction which involves automatic extraction of topic words and their relationships still needs further research. In addition, because of the complexity of the network structure, crawlers easily fall into a loop trap, and it wastes computation resources. Although the SA algorithm can partially solve this problem, it is affected by the temperature cooling rate. If the cooling rate is too low, the crawler will spend more time. If the cooling rate is too high, the crawler may skip the process of obtaining the optimal crawling direction. Therefore, further investigation is needed to ensure the global nature of the focused crawler. In addition, the topic relevance evaluation of the hyperlinks in CPEM is weighted by multiple evaluating indicators, and this is regarded as a single-objective optimization problem. However, the reasonable weight factors are hard to determine. In the future work, we plan to continue research on automated ontology construction, design of the annealing strategy, and the applications of multi-objective intelligent optimization algorithms in focused crawlers. We believe this work can achieve good results.

Contributors

Jingfa LIU designed the research. Fan LI drafted the paper, implemented the software, and performed the experiments. Ruoyao DING and Zi'ang LIU revised and finalized the paper.

Compliance with ethics guidelines

Jingfa LIU, Fan LI, Ruoyao DING, and Zi'ang LIU declare that they have no conflict of interest.

References

Bajpai N, Arora D, 2018. Domain-based search engine evaluation. In: Saeed K, Chaki N, Pati B, et al. (Eds.), Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, volume 564. Springer, Singapore, p.711-720. https://doi.org/10.1007/978-981-10-6875-1_69

Boukadi K, Rekik M, Rekik M, et al., 2018. FC4CD: a new SOA-based focused crawler for cloud service discovery. *Computing*, 100(10):1081-1107. https://doi.org/10.1007/s00607-018-0600-2

Capuano A, Rinaldi AM, Russo C, 2020. An ontology-driven multimedia focused crawler based on linked open data and deep learning techniques. *Multim Tools Appl*, 79(11): 7577-7598. https://doi.org/10.1007/s11042-019-08252-2

Chen YB, Zhang Z, Zhang T, 2011. A searching strategy in topic crawler using ant colony algorithm. *Microcomput Appl*, 30(1):53-56 (in Chinese). https://doi.org/10.19358/j.issn.1674-7720.2011.01.018

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

Colazzo D, Ghelli G, Pardini L, et al., 2013. Almost-linear inclusion for XML regular expression types. *ACM Trans Database Syst*, 38(3):15. https://doi.org/10.1145/2508020.2508022

Derrac J, García S, Molina D, et al., 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput*, 1(1):3-18. https://doi.org/10.1016/j.swevo.2011.02.002

Dong Y, Liu JF, Liu WJ, 2020. Focused crawler strategy based on multi-objective ant colony algorithm. *Comput Eng*, 46(9):274-282 (in Chinese). https://doi.org/10.19678/j.issn.1000-3428.0055967

Du YJ, Pen QQ, Gao ZQ, 2013. A topic-specific crawling strategy based on semantics similarity. *Data Knowl Eng*, 88:75-93. https://doi.org/10.1016/j.datak.2013.09.003

Du YJ, Hai YF, Xie CZ, et al., 2014. An approach for selecting seed URLs of focused crawler based on user-interest ontology. *Appl Soft Comput*, 14:663-676. https://doi.org/10.1016/j.asoc.2013.09.007

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

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

Gruber TR, 1995. Toward principles for the design of ontologies used for knowledge sharing? *Int J Human-Comput Stud*, 43(5-6):907-928. https://doi.org/10.1006/ijhc.1995.1081

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 S, Cheng JX, Cai XB, 2009. Focused crawler based on simulated anneal algorithm. *Comput Technol Dev*, 19(12): 55-58, 62 (in Chinese). https://doi.org/10.3969/j.issn.1673-629X.2009.12.015

Jia JF, Tumanian V, Li GQ, 2021. Discovering semantically related technical terms and web resources in Q&A discussions. *Front Inform Technol Electron Eng*, 22(7):969-985. https://doi.org/10.1631/FITEE.2000186

Jing WP, Wang YJ, Dong WW, 2016. Research on adaptive genetic algorithm in application of focused crawler search strategy. *Comput Sci*, 43(8):254-257 (in Chinese). https://doi.org/10.11896/j.issn.1002-137X.2016.8.051

Khadir AC, Aliane H, Guessoum A, 2021. Ontology learning: grand tour and challenges. *Comput Sci Rev*, 39:100339. https://doi.org/10.1016/j.cosrev.2020.100339

Lakzaei B, Shamsfard M, 2021. Ontology learning from relational databases. *Inform Sci*, 577:280-297. https://doi.org/10.1016/j.ins.2021.06.074

Liu B, Jiang SY, Zou Q, 2020. HITS-PR-HHblits: protein remote homology detection by combining PageRank and hyperlink-induced topic search. *Brief Bioinform*, 21(1): 298-308. https://doi.org/10.1093/bib/bby104

Liu JF, Li G, Chen DB, et al, 2010. Two-dimensional equilibrium constraint layout using simulated annealing. *Comput Ind Eng*, 59(4):530-536. https://doi.org/10.1016/j.cie.2010.06.009

Liu JF, Li F, Jiang SY, 2019a. 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, Li X, Jiang SY, 2019b. Focused crawler for rainstorm disaster strategy based on web space evolutionary algorithm. *Comput Eng*, 45(2):184-190 (in Chinese). https://doi.org/10.19678/j.issn.1000-3428.0052035

Liu JF, Gu YP, Liu WJ, 2020. Focused crawler method combining ontology and improved Tabu search for meteorological disaster. *J Comput Appl*, 40(8):2255-2261 (in Chinese).

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

Patel A, Schmidt N, 2011. Application of structured document parsing to focused web crawling. *Comput Stand Inter*, 33(3):325-331. https://doi.org/10.1016/j.csi.2010.08.002

Prakash J, Kumar R, 2015. Web crawling through shark-search using PageRank. *Proc Comput Sci*, 48:210-216. https://doi.org/10.1016/j.procs.2015.04.172

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

Rios-Alvarado AB, Lopez-Arevalo I, Sosa-Sosa VJ, 2013. Learning concept hierarchies from textual resources for ontologies construction. *Expert Syst Appl*, 40(15):5907-5915. https://doi.org/10.1016/j.eswa.2013.05.005

Tong YL, 2008. Application of focused crawler using adaptive dynamical evolutional particle swarm optimization. *Geom Inform Sci Wuhan Univ*, 33(12):1296-1299 (in Chinese).

Tsikrika T, Moumtzidou A, Vrochidis S, et al., 2016. Focussed crawling of environmental web resources based on the combination of multimedia evidence. *Multim Tools Appl*, 75(3):1563-1587. https://doi.org/10.1007/s11042-015-2624-3

Vidal MLA, da Silva AS, de Moura ES, et al., 2006. Structure-driven crawler generation by example. Proc 29th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.292-299. https://doi.org/10.1145/1148170.1148223

Wang ZG, Meng BJ, 2014. A comparison of approaches to Chinese word segmentation in Hadoop. Proc IEEE Int Conf on Data Mining Workshop, p.844-850. https://doi.org/10.1109/ICDMW.2014.43

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

Zhu G, Yang JY, Wu XH, et al., 2017. Research on construction of hierarchy relationship and ontology of meteorological disaster based on FCA. *Mod Inform*, 37(5):79-88 (in Chinese). https://doi.org/10.3969/j.issn.1008-0821.2017.05.014

List of supplementary materials

Fig. S1 A global ontology structure about the topic of rainstorm disaster

Table S1 Seed URLs