Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2022.Doi Number

A Semantic and Intelligent Focused Crawler based on BERT Semantic Vector Space Model and Hybrid Algorithm (October 2024)

Wenhao Huang[1,3], Jiahao Zhang[2], Xin Li[2], Xiao Zhou[2], Deyu Qi[3], *, Jianqing Xi[1], *, Wenjun Liu[2,4],*

[1]School of Software Engineering, South China University of Technology, Guangzhou 510006, China
[2]School of Computer and Software Engineering, XiHua University, Chengdu 610039, China
[3]Guangdong University of Foreign Studies South China Business College, Guangzhou 510545, China
[4]Sichuan Provincial Engineering Research Center of Hydroelectric Energy Power Equipment Technology, Chengdu 610039, China

* Corresponding Author: office.csasc@qq.com (Deyu Qi)

* Corresponding Author: 422582346@qq.com (Jianqing Xi)

* Corresponding Author: liuwenjun@mail.xhu.edu.cn (Wenjun Liu)

This work is supported by the National Natural Science Foundation of China (Grant No. 61070015), Guangdong Provincial Key Technology Research Center for Object Digitization and Epidemic Intelligence Prevention and Control, Guangdong Provincial Engineering Technology Research Center for General Colleges and Universities, Guangdong Provincial Department of Education (Grant No. 2022GCZX013), the Intelligent Policing Key Laboratory of Sichuan Province (Grant No. ZNJW2024KFQN012), the Open Research Subject of Sichuan Provincial Engineering Research Center of Hydroelectric Energy Power Equipment Technology (Grant No. SDNY-2024001), and the College Student Innovation and Entrepreneurship Training Project of Sichuan Province (Grant No. D202405310751003590).

ABSTRACT The goal of a focused crawler is to selectively fetch pages that are relevant to a given topic. Previous crawlers use text content to determine text topic relevance and manually determined weighting factors to predict the priority of unvisited URLs. However, there are still some problems in the above focused crawler methods, the calculation formula of semantic similarity between words is flawed. The weighting factor for the priority of unvisited URLs is determined arbitrarily. In order to solve the above problems, this paper proposes a semantic and intelligent focused crawler based on BERT semantic vector space model and hybrid algorithm. This method used BERT semantic vector space model to calculate the topic relevance of documents, and used a hybrid algorithm to optimize the weighting factor of unvisited URL priority. The experimental results show that the proposed BSVSM-HA crawler can obtain better evaluation indicators compared with the other three crawlers including Word2vec crawler, ELMO crawler and BSVSM crawler. In conclusion, the semantic and intelligent crawler proposed in this paper makes the semantic similarity between terms more accurate, and improves the topic relevance of the text, and the optimized weighting factor makes the priority evaluation of unvisited URLs more accurate.

INDEX TERMS Focused Crawler, Semantic Vector Space Model, Hybrid Algorithm

I. INTRODUCTION

With the rapid advancement of the Internet, the volume of information on the network is increasing sharply. People require valuable content from massive data, and traditional manual methods are no longer sufficient to meet this demand. As an automated data acquisition tool, the essence of a crawler lies in subject screening during webpage crawling, ensuring that only web page information related to the subject is captured as much as possible, and it can swiftly and efficiently capture required data from the Internet. Therefore, crawlers play a crucial role in information collection, data analysis, and other fields [1]. In daily life, with growing demands for data, the application scenarios of crawler technology are becoming increasingly extensive. For instance, search engines rely on crawlers to gather and update web page information, e-commerce websites depend on crawlers to collect product information and prices, news media utilize crawlers to gather news articles and comments, etc [2-4].

Focused crawlers demonstrate superior efficiency and yield higher quality data. In contrast, ordinary crawlers may expend time and resources on irrelevant web pages [5]. Currently, mainstream topic crawlers can be categorized as non-learning or learning-based. Non-learning focused crawlers encompass traditional and semantic variants. Classic focused crawlers are further classified based on text content evaluation, link structure evaluation, and combined content-link structure evaluation [6]. An example of a traditional topic crawler that assesses the relevance of unvisited hyperlinks based on content and link is the Baby-crawler, which utilizes a priority score for URL downloads [7]. The Semantic Disambiguation Space Vector Model (SDVSM) represents a core technology in semantic crawling, integrating Semantic Disambiguation Graph (SDG) with Semantic Vector Space Model (SVSM) to address limitations in similarity accuracy between text and given topics encountered by SSRM models, thereby enhancing crawler performance [8-9]. Learning focused crawlers leverage machine learning methods to predict the priority of unvisited URLs, while ontology-based learning focused crawlers integrate learning technology with semantic technology. A typical example is the learning-based crawler using URL knowledge base [10], enabling continuous updating of URL content to enhance topic similarity accuracy. Additionally, there are topic crawlers based on ontology learning that utilize text and multimedia web content, optimizing crawling tasks through semantic-based technology and associating different topics using ontology to understand their relationships. These topic-focused approaches use text content and link structure to determine page relevance to a given topic, predict access priority within the URL queue, ultimately obtaining relevant web pages accurately [11-12]. However, the above topic crawler method still has some problems, specifically as follows:

(1) The semantic similarity calculation formula between words is flawed. The previous method of evaluating semantic similarity is to extract sentence similarity matrix based on WordNet to calculate the semantic similarity between words. However, this method ignores the order of words in the sentence and the deep semantics of the sentence, which is easy to cause the distortion of the sentence semantics. For example, the sentences "I am going from Chengdu to Beijing this holiday" and "I am going from Beijing to Chengdu this holiday" are easy to cause misjudgments, resulting in inaccurate semantic similarity between words.

(2) The weighting factor of the priority of unvisited URLs is random. In traditional focused crawlers, these weighting factors are obtained through personal experience. Because the priority obtained by integrating these weighting factors and different text similarity cannot objectively and truly show the contribution degree of different texts to the hyperlink priority. Therefore, these inaccurate weighting factors will misguide the focused crawler to collect a large number of irrelevant web pages, and the priority of unvisited hyperlinks will be greatly biased, resulting in the performance degradation of the focused crawler.

In order to solve the above problems, this paper proposes a semantic and intelligent focused crawler based on BERT semantic vector space model and hybrid algorithm. This method used BERT semantic vector space model to calculate the topic relevance of documents, and used a hybrid algorithm to optimize the weighting factor of unvisited URL priority. Firstly, BERT is used to construct word vectors and evaluate the semantic similarity between words. The document semantic vector and the topic semantic vector are constructed respectively, and the cosine similarity between them is used to calculate the topic relevance of the document more accurately. Then, by randomly generating a population and calculating the fitness of each individual in the population, the hybrid algorithm was used to determine the optimal values of the four weighting factors. Finally, the priority of unvisited hyperlinks was obtained by linear integration of document topic relevance and optimal weighting factor, and the priority of URLs was predicted. The experimental results show that the proposed BSVSM-HA crawler can obtain better evaluation indicators compared with the other three crawlers. In conclusion, the semantic and intelligent crawler proposed in this paper makes the semantic similarity between terms more accurate, and improves the topic relevance of the text, and the optimized weighting factor makes the priority evaluation of unvisited URLs more accurate.

The contributions of this paper are summarized as follows:

(1) BSVSM model is proposed to compute the topic-relevance of documents. This model uses BERT to construct word vectors, and uses the word vectors generated by BERT to evaluate the semantic relevance between words, then constructs the document semantic vector and the topic semantic vector, and finally calculates the topic relevance of the document by using the cosine similarity of the two vectors.

(2) The hybrid algorithm is used to optimize the weighting factor of the priority of unvisited URLs. The algorithm intelligently obtains four optimal weighting factors of full text, anchor text, title text and text around paragraphs through three rules of selection, crossover and mutation and simulated annealing iteration. These four weighting factors are used to predict the priority of unvisited URLs.

The remaining contents of this paper are organized as follows. The second part introduces two methods to calculate the topic relevance of documents. The third part proposes a semantic similarity intelligent crawler based on BERT semantic vector space model and hybrid algorithm. Section 4 shows and analyzes the experimental results. Section 5 presents the conclusion and future work.

II. RELATED WORKS

Topic crawler uses the relationship between text content and various URL link structures to obtain the web pages with high topic relevance. Focused crawlers need to predict the access priority of the URL queue, and the priority of unvisited URLs is related to the topic relevance of the text, the anchor text, and the context of the article [13]. According to whether the training set is used to train, the topic crawler can be divided into two categories: learning topic crawler and non-learning topic crawler.

A. Non-Learning Focused Crawlers

Non-learning focused crawlers can be divided into traditional focused crawlers and semantic focused crawlers, which retrieve more web pages related to a given topic through text content, link structure and term similarity respectively.

Traditional focused crawlers can be divided into three types: evaluation based on text content, evaluation based on link structure, and evaluation based on content and link structure. The search strategy based on content evaluation mainly uses the correlation between the text content of the web page and the topic content of the hyperlink to evaluate the topic relevance of the unvisited hyperlink [14]. The search strategy based on link structure evaluation determines the topic relevance of unvisited hyperlinks by analyzing the coreference relationship between web pages [15]. The evaluation based on content and link structure considers the above two methods simultaneously to calculate the topic relevance of unvisited hyperlinks [16].

The typical Crawler that evaluates the topic relevance of unvisited hyperlinks based on content and links is Baby-crawler [17]. By using the hierarchical structure of T-Graph, the Baby-crawler can assign an appropriate priority score to each unvisited link, and download URLs according to this priority. The Treasure Crawler is evaluated against specific information retrieval criteria, such as recall and precision, both of which have values close to 50%. Obtaining such results affirms the importance of the proposed method [18].

Semantic Disambiguation Space Vector Model (SDVSM) is one of the core technologies of semantic crawler. Different from the traditional SSRM model, in the SSRM based topic crawler, if the text and the topic words are the same or synonymous, and the TF*IDF weights of the topic words are very different, the accurate similarity between the text and the given topic cannot be obtained, thus reducing the performance of the crawler [19]. The SDVSM method combines the Semantic Disambiguation Graph (SDG) and the semantic Vector Space Model (SVSM). SDG is used to remove ambiguous terms not related to a given topic from the presentation terms of the retrieved web pages [20]. The SVSM algorithm constructs text and topic semantic vectors according to the TF*IDF weight of terms and the semantic similarity between terms, and calculates the topic similarity of the text.

B. Learning Focused Crawlers

Learning focused crawlers learn and predict the priority of unvisited URLs by using machine learning methods. This method needs to train through a large number of training sets, and uses the relevant algorithm to calculate the topic relevance of the text.

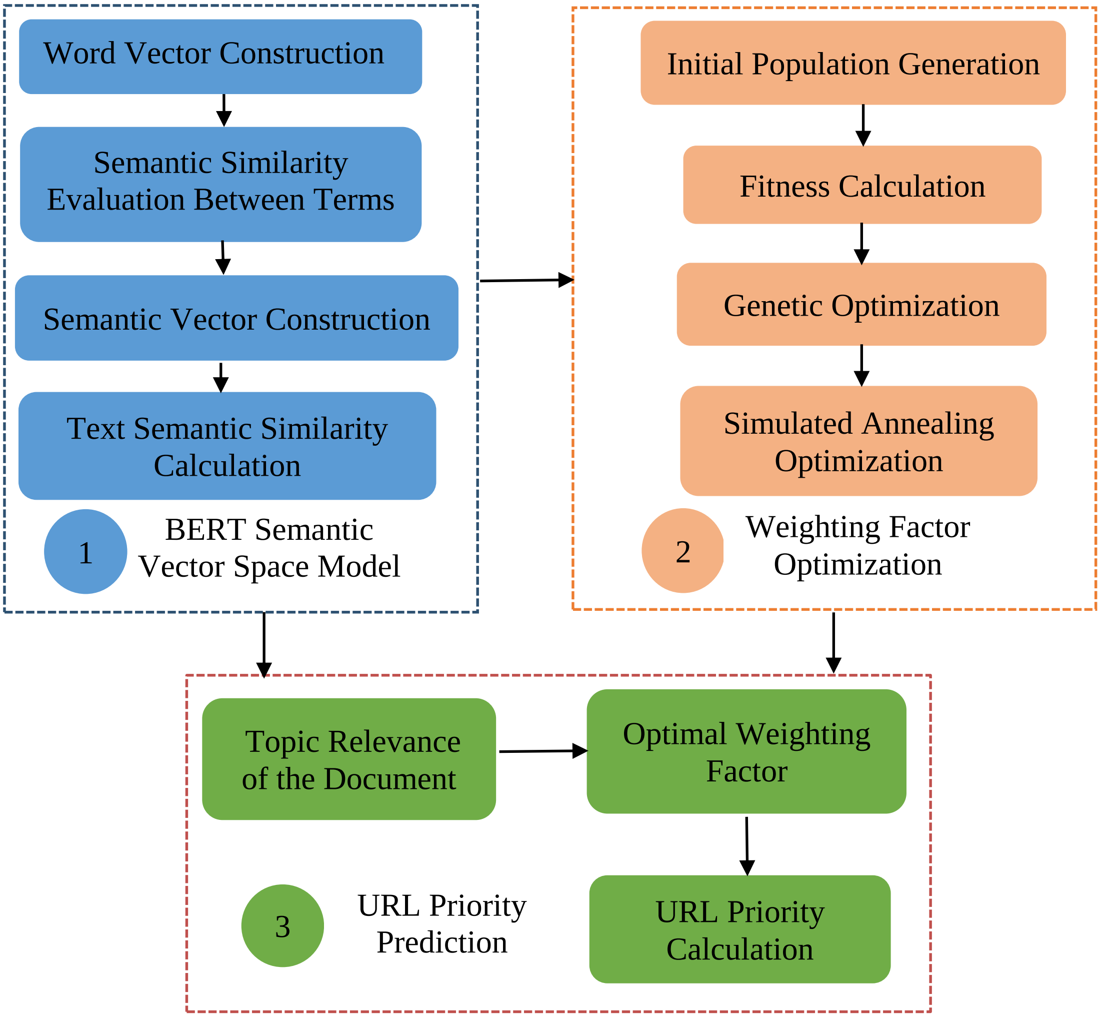
Learning-based crawlers using URL knowledge bases utilize parent page content, anchor information as well as URL content to evaluate topic similarity [21]. This method can make the focused crawler have the ability to learn and constantly update the URL content, so as to improve the accuracy of topic similarity [22]. proposed a method of embedding words to calculate the topic relevance, and used the cosine similarity between the topic vector of the topic and the content of the web page as the input of the random forest classifier to predict the topic relevance of the web page [23]. This paper proposed a text classification model based on Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) with word embedding to improve the accuracy of web page classification. Ontology-driven multimedia crawler based on linked open data and deep learning technology uses word embedding based on Adagrad optimized Skip Gram Negative Sampling (a-SGNS) and Recurrent Neural Network (RNN), uses word embedding matrix to calculate cosine similarity to construct feature vector [24]. This feature vector is used as the input of the RNN to predict the relevance of the website [25].

proposed an ontology learning focused crawler, which computes the relevance score of web pages by integrating text and multimedia content [26]. Applying ontology learning and multi-objective ant colony optimization method to meteorological disaster domain knowledge, this paper proposed a semi-automatic domain ontology construction method based on ontology learning technology. This method combined latent Dirichlet allocation and Apriori algorithm. A multi-objective optimization model for link evaluation and a multi-objective ant colony optimization algorithm (MOACO) were proposed to select the optimal hyperlinks [27]. proposed a focused crawler based on text and multimedia web content, which uses semantically based techniques to improve the crawling task and combines the results with new techniques such as convolutional neural networks and linked open data, and uses usage ontologies to relate different topics and understand their relationships [28]. proposed a focused crawler based on ontology and host information and improved tabu search algorithm to solve the problems of incomplete topic description and repeated crawling of access hyperlinks in traditional focused crawling methods.

III. FOCUSED CRAWLER BASED ON BSVSM AND HA

This paper proposes a semantic and intelligent focused crawler based on BERT semantic vector space model and hybrid algorithm. This method used BERT semantic vector space model to calculate the topic relevance of documents, and used a hybrid algorithm to optimize the weighting factor of unvisited URL priority. Fig. 1 shows the framework diagram of semantic and intelligent focused crawler based on BERT semantic vector space model and hybrid algorithm. In Fig. 1,

FIGURE 1. The flowchart of a focused crawler based on BSVSM and HA


Word Vector Construction
Semantic Similarity Evaluation Between Terms
Semantic Vector Construction
Text Semantic Similarity Calculation
1
BERT Semantic Vector Space Model
Initial Population Generation
Fitness Calculation
Genetic Optimization
Simulated Annealing Optimization
2
Weighting Factor Optimization
Topic Relevance of the Document
Optimal Weighting Factor
URL Priority Calculation
3
URL Priority Prediction

the method is divided into three main modules: BERT semantic vector space model, weighting factor optimization, URL priority prediction. The BERT semantic vector space model uses BERT to construct word vectors and evaluate the semantic similarity between words, then constructs the document semantic vector and the topic semantic vector, and uses the cosine similarity of the two vectors to calculate the topic relevance of the document. Firstly, the population is generated randomly, and the fitness of the individual is calculated. Then, the three genetic rules of selection, crossover and mutation and simulated annealing are used to continuously optimize the weighting factor. URL priority prediction uses BSVSM to obtain the topic relevance of the document, and uses the hybrid algorithm to obtain the optimal weighting factor, and then linearly integrates the two as the priority of the unvisited URLs.

A. BERT Semantic Vector Space Model

BERT semantic vector space model uses BERT model to calculate the semantic similarity between words, constructs semantic vectors, and realizes the topic relevance evaluation of documents. Firstly, BERT semantic vector space model uses BERT to generate word vectors. Second, the module uses the cosine similarity of two vectors to calculate the semantic similarity between terms. This module then builds the text semantic vector and the topic semantic vector. Finally, the cosine similarity between the document semantic vector and the topic semantic vector was used to calculate the topic relevance of the document. The BERT semantic vector space model includes word vector construction, semantic similarity evaluation between words, semantic vector construction, and topic relevance calculation of documents. The above four are described below.

1) WORD VECTOR CONSTRUCTION

Word vector construction uses BERT to generate word vectors with semantics and word order. This module is divided into four processes: word embedding, multi-head attention mechanism, feedforward neural network, encoding layer, and output layer. Word embedding is to calculate and sum the input words by three embedding layers to obtain the input vector. The multi-head attention mechanism is the process of concatenating the Q, K, and V matrices obtained after projection of the input vectors after normalization. A feed-forward neural network is a one-way propagation neural network that takes the output of the previous layer and feeds this output to the next layer. The encoding layer is an important process that uses the multi-head attention mechanism and feed-forward neural network to calculate the input word vector so that the word vector has the characteristics of semantics, position and order. The output layer is the process of obtaining the final output vector after linear transformation and normalization of the vectors obtained by the encoding layer.

The formula for word vector construction is as follows:

$$\begin{aligned} &OV = BERT(IV) \\ &IV \in R^{n\times h} \quad OV \in R^{n\times z} \end{aligned} \tag{1}$$

where OV represents the output matrix consisted of word vectors generated by the BERT model, IV represents the input matrix obtained by token embeddings, segment embeddings and position embeddings, n represents the number of words in the text, h represents the dimension of the input vector of each word, z represents the dimension of the output vector of each word.

2) SEMANTIC SIMILARITY EVALUATION BETWEEN TERMS

The semantic similarity between terms is calculated using the word vectors generated by Bert. This module uses BERT to generate the text term vector and the topic term vector, and uses the cosine similarity between the two vectors as the semantic similarity between the two terms, that is, it can obtain the semantic similarity between each document term and each topic term. The closer the value is to 1, the more similar the two terms are. Conversely, the closer the result is to 0, the lower the semantic similarity between two terms. The formula for evaluating semantic similarity between terms is as follows:

$$Sem(w_d, w_t) = \frac{\sum_{i=1}^{k} w_{di} w_{ti}}{\sqrt{\sum_{i=1}^{k} (w_{di})^2}\sqrt{\sum_{i=1}^{k} (w_{ti})^2}} \tag{2}$$

where $Sem(w_d, w_t)$ represents the semantic similarity between two terms w_d and w_t ,Where w_d and w_t represent document terms and topic terms respectively, w_{di} and w_{ti} denote the one-dimensional component of the document term vector and the topic term vector respectively, w_{dk} and w_{tk} are the k-dimensional component of the document term vector and the subject term vector respectively.

3) SEMANTIC VECTOR CONSTRUCTION

The semantic vector construction is to obtain the document and topic semantic vector by using the TF*IDF weight of the words and the semantic similarity between the words. This module is divided into two processes: document semantic vector construction and topic semantic vector construction. The construction of document semantic vector is to construct the document semantic vector by using multiple dimensions of the document semantic vector components, and the document semantic vector components of each dimension are obtained by multiplying the TF*IDF weight of the document terms by the semantic similarity between the text terms and the topic terms. The topic semantic vector construction is to construct the document semantic vector by using the multi-dimensional topic semantic vector components, and the topic semantic vector components of each dimension are obtained by multiplying the TF*IDF weight of the topic terms by the semantic similarity between the document terms and the topic terms. The semantic vector construction is calculated as follows:

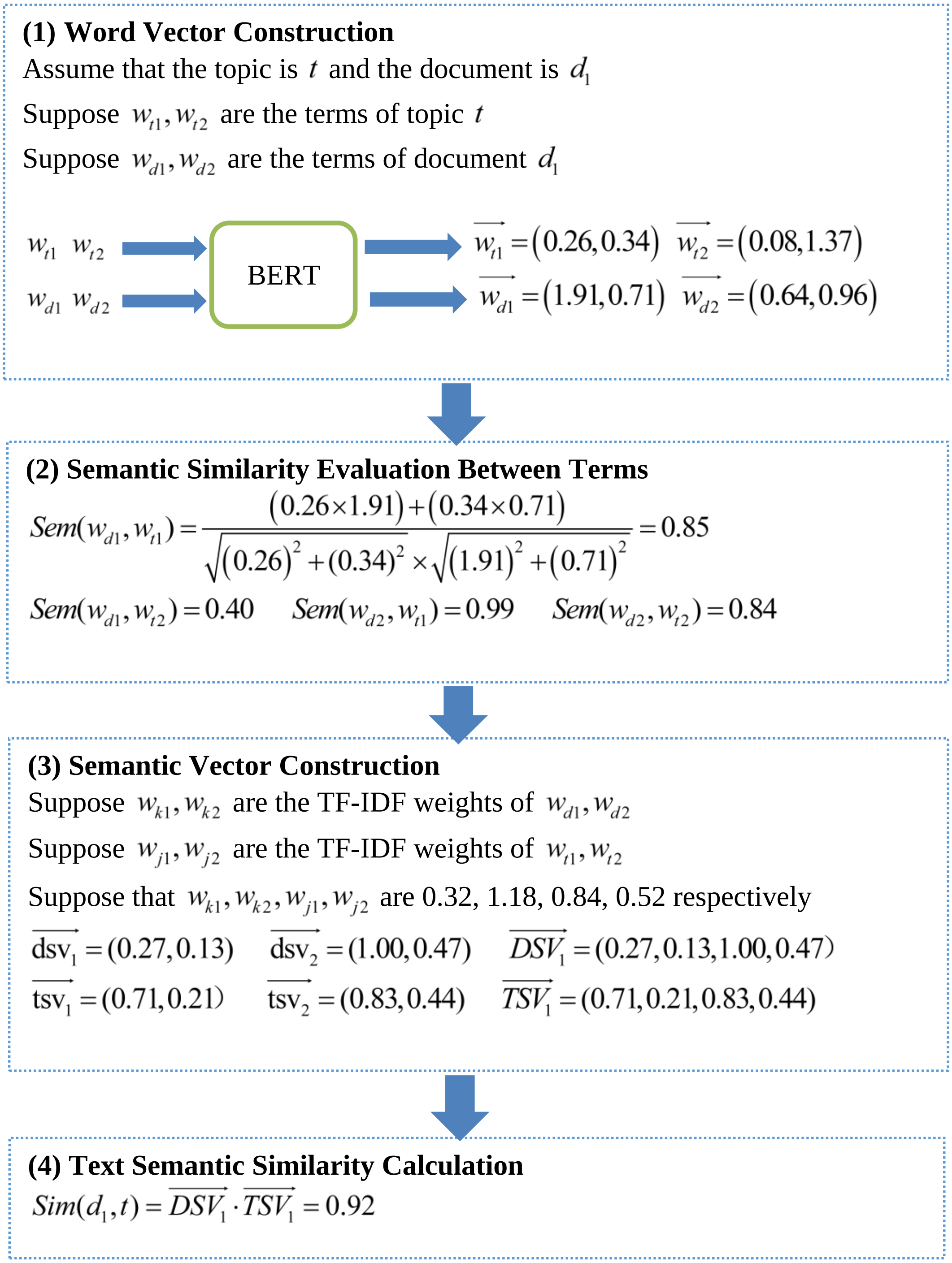
$$\begin{aligned} &\overrightarrow{dsv_i} = (w_{ki} \bullet sem_{i1}^{k},\ w_{ki} \bullet sem_{i2}^{k}, \cdots, w_{ki} \bullet sem_{in}^{k}) \\ &\overrightarrow{DSV_k} = (\overrightarrow{dsv_1}, \overrightarrow{dsv_2}, \cdots, \overrightarrow{dsv_m}) \\ &(1 \le i \le m) \\ &\overrightarrow{tsv_j} = (w_{t1} \bullet sem_{j1}^{k}, w_{t2} \bullet sem_{j2}^{k}, \cdots, w_{tn} \bullet sem_{jn}^{k}) \\ &\overrightarrow{TSV_k} = (\overrightarrow{\mathrm{t}sv_1}, \overrightarrow{\mathrm{t}sv_2}, \cdots, \overrightarrow{\mathrm{t}sv_m}) \\ &(1 \le j \le m) \end{aligned} \tag{3}$$

where dsv_i represents the i-th dimension component of the document semantic vector, w_{ki} represents the TF*IDF weight of term i in document k, sem_{ij}^{k} represents the semantic similarity between document term i and topic term j in document k and topic t, m represents the number of terms in document k, n represents the number of terms in topic t, DSV_k represents the document semantic vector, tsv_j represents the j-th dimension component of the topic semantic vector, and w_{tj} represents the number of terms in the topic semantic vector. TSV_k denote the TF*IDF weight of term j in topic t and denote the topic semantic vector.

4) TEXT SEMATIC SIMILARITY CALCULATION

The topic relevance of a document is calculated by using the cosine similarity between the semantic vector of the document and the semantic vector of the topic. SVSM model uses WordNet to calculate the semantic similarity between words, which ignores the word order and the deep meaning of the sentence. The BSVSM model uses BERT to generate word vectors with word order and semantics, and uses this word vector to calculate the semantic similarity between words. Finally, the topic relevance obtained by this model is more accurate. The formula for calculating topic relevance is as follows:

FIGURE 2. An example of BERT Semantic Vector Space Model


(1) Word Vector Construction
Assume that the topic is t and the document is d_1
Suppose w_{t1}, w_{t2} are the terms of topic t
Suppose w_{d1}, w_{d2} are the terms of document d_1
w_{t1} w_{t2}
w_{d1} w_{d2}
BERT
$\overrightarrow{w_{t1}} = (0.26, 0.34)$ $\overrightarrow{w_{t2}} = (0.08, 1.37)$
$\overrightarrow{w_{d1}} = (1.91, 0.71)$ $\overrightarrow{w_{d2}} = (0.64, 0.96)$
(2) Semantic Similarity Evaluation Between Terms
$Sem(w_{d1}, w_{t1}) = \frac{(0.26 \times 1.91) + (0.34 \times 0.71)}{\sqrt{(0.26)^2 + (0.34)^2} \times \sqrt{(1.91)^2 + (0.71)^2}} = 0.85$
$Sem(w_{d1}, w_{t2}) = 0.40$ $Sem(w_{d2}, w_{t1}) = 0.99$ $Sem(w_{d2}, w_{t2}) = 0.84$
(3) Semantic Vector Construction
Suppose w_{k1}, w_{k2} are the TF-IDF weights of w_{d1}, w_{d2}
Suppose w_{j1}, w_{j2} are the TF-IDF weights of w_{t1}, w_{t2}
Suppose that $w_{k1}, w_{k2}, w_{j1}, w_{j2}$ are 0.32, 1.18, 0.84, 0.52 respectively
$\overrightarrow{\mathrm{dsv}_1} = (0.27, 0.13)$ $\overrightarrow{\mathrm{dsv}_2} = (1.00, 0.47)$ $\overrightarrow{DSV_1} = (0.27, 0.13, 1.00, 0.47)$
$\overrightarrow{\mathrm{tsv}_1} = (0.71, 0.21)$ $\overrightarrow{\mathrm{tsv}_2} = (0.83, 0.44)$ $\overrightarrow{TSV_1} = (0.71, 0.21, 0.83, 0.44)$
(4) Text Semantic Similarity Calculation
$Sim(d_1, t) = \overrightarrow{DSV_1} \cdot \overrightarrow{TSV_1} = 0.92$

$$Sim(d_k, t) = \overrightarrow{DSV_k} \bullet \overrightarrow{TSV_k} = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} w_{ki} w_{tj} (sem_{ij}^{k})^2}{\sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (w_{ki} sem_{ij}^{k})^2} \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (w_{ij} sem_{ij}^{k})^2}} \tag{4}$$

where $Sim(d_k, t)$ represents the degree of document topic relevance. $\overrightarrow{DSV_k}$ is the document semantic vector, $\overrightarrow{TSV_k}$ is the topic semantic vector. m and n are the number of terms in document k and topic t, respectively. w_{ki} is the TF*IDF weight of item i in document k, w_{tj} is the TF*IDF weight of item j in document t, sem_{ij}^{k} is the semantic similarity between item i in document k and item j in topic t.

Fig. 2 shows an example of BERT Semantic Vector Space Model. In Fig.2, the topic and documents are t and d_1, w_{t1}, w_{t2} are terms of t, w_{d1}, w_{d2} are terms of document d_1, $\overrightarrow{w_{t1}}$, $\overrightarrow{w_{t2}}$, $\overrightarrow{w_{d1}}$, $\overrightarrow{w_{d2}}$ are the word vector derived by BERT, $Sem(w_{d1}, w_{t1})$ is the semantic similarity between term w_{d1} and w_{t1}, w_{k1} w_{k2} are the TF-IDF weights of w_{d1}, w_{d2}, w_{j1}, w_{j2} are the TF-IDF weights of w_{t1}, w_{t2}, dsv_1 represents the i-th component of the document semantic vector , the value of dsv_1 is $(0.27, 0.13)$, and the value of dsv_2 is $(1.00, 0.47)$, DSV_1 and TSV_1 represent document semantic vector and topic semantic vector respectively. $Sim(d_1, t)$ indicates the topic relevance of the document d_1 , it is calculated to be 0.92.

B. Weighting Factor Optimization

Weighting factor optimization uses a hybrid algorithm to obtain the four optimal weighting factors for the full text, anchor text, heading text, and text around the paragraph. First, the weighting factor optimization randomly generates multiple different individuals, which constitute an initial population, and the genetic algorithm will iteratively optimize this initial population until convergence. Second, this module utilizes the true and predicted values of the topic relevance of the training hyperlinks to determine the fitness function and calculate the fitness of each individual within the population. Then, this module continuously optimizes the population using the three major genetic rules of selection, crossover, and mutation. Finally, the weighting factors obtained by the genetic algorithm are annealed and optimized using the simulated annealing algorithm, which in turn optimizes the four weighting factors of the full text, anchor text, headline text, and text around the paragraph. Weighting factor optimization contains initial population generation, fitness calculation, genetic rules, and simulated annealing optimization. The following section describes the content of the above three modules.

1) INITIAL POPULATION GENERATION

Initial population generation is to randomly generate multiple individuals to form an initial population. Each individual in the initial population is not the same as each other, if the same individual in the generated initial population is not less than half of the total number of individuals in the population, it is necessary to re-generate the initial population, and vice versa, the same individuals are re-generated until each individual in the population is not the same as each other. Individual is a four-dimensional vector composed of full text weighting factor, anchor text weighting factor, title text, text around the paragraph, and the value range of each vector dimension is [0,1]. The initial population generated in this paper has diversity, and the initial population produced in this way can enhance the exploration ability of the genetic algorithm to the global optimal solution.

The representation of initial population generation is as follows:

$$\begin{aligned} & i_k = (\lambda_{k1}, \lambda_{k2}, \lambda_{k3}, \lambda_{k4}) \qquad 0 \le \lambda_{k1}, \lambda_{k2}, \lambda_{k3}, \lambda_{k4} \le 1 \\ & I_{\text{rand}} = \{i_1,\ i_2, \dots,\ i_{\mathrm{n}}\} \qquad \mathrm{n} \ge 2 \end{aligned} \tag{5}$$

where i_k denotes the kth individual in the population, λ_{k1}, λ_{k2}, λ_{k3}, λ_{k4} denotes the full-text weighting factor, anchor text weighting factor, headline text weighting factor, and text weighting factor around the paragraph, respectively, in the kth individual, denotes the initial randomly generated collection of the population, and n denotes the number of individuals in the population and n is greater than or equal to 2.

2) FITNESS CALCULATION

Adaptation calculation is to utilize the real and predicted values of training hyperlink topic similarity to calculate the adaptation of that individual. Adaptation calculation is divided into two processes: prediction value calculation and individual adaptation evaluation. Predictive value calculation is obtained by linearly integrating the four weighting factors corresponding to the current individual with the two true values of text similarity of the training hyperlinks. Individual fitness assessment is calculated by utilizing the difference between the true value of topic similarity and the predicted value for each training hyperlink. Each individual in the population has a fitness value, and the fitness value represents the individual's ability to adapt to the environment; the larger the fitness value, the stronger the individual's ability to adapt to the environment, i.e., the weighting factor corresponding to the individual is good. Conversely, the weighting factor corresponding to the individual is poor.

The formula for calculating the fitness is as follows:

$$\begin{aligned} & y_{ic} = \lambda_{k1} Sim_{k1} + \lambda_{k2} Sim_{k2} + \lambda_{k3} Sim_{k3} + \lambda_{k4} Sim_{k4} \\ & fit(i_k) = \frac{1}{\sqrt{\sum_{j=1}^{n} (y_{it} - y_{ic})^2}} \end{aligned} \tag{6}$$

where y_{ic} denotes the predicted value of the topic relevance of training hyperlink i, λ_{k1}, λ_{k2}, λ_{k3}, λ_{k4} are the optimal weighting factors for the full text and anchor text, respectively, Sim_{k1}, Sim_{k2}, Sim_{k3} and Sim_{k4} are the true value of the topic similarity of the full text, the true value of the topic similarity of the anchor text, the true value of the topic similarity of the header text, and the true value of the topic similarity of the text around the paragraph, respectively. $fit(i_k)$ denotes the individual fitness function, n is the total number of training hyperlinks, and y_{it} denotes the true value of the topic relevance of training hyperlink i.

3) GENETIC OPTIMIZATION

Genetic algorithm is an optimization algorithm based on the principle of natural evolution, which is mainly used to solve complex optimization problems. It simulates the mechanisms of selection, crossover and mutation in the process of biological evolution, and searches for the optimal solution through the continuous evolution and competition of multiple candidate solutions. Firstly, the selection rule selects the more adapted individuals of the parent generation into the offspring by utilizing the rules of roulette. Then, the crossover rule obtains two new offspring individuals with different fitness by exchanging the gene segments of two individuals of the parent generation. Finally, the mutation rule obtains offspring individuals with different fitness by changing the gene segments of the parent individuals. Genetic optimization involves three processes: selection, crossover, and mutation.

(1) Selection Rule

The selection rule is to use the roulette rule to select the individual with higher fitness from the parent generation into the offspring, which is used to select the individual with higher fitness in the parent generation into the offspring i.e. to pass its excellent characteristics to the offspring. This rule uses the ratio of the fitness of a single individual to the sum of the fitness of all individuals as the selection probability of that individual, i.e., the ratio of the area of the roulette wheel to the total area of the wheel in the roulette rule to obtain the stake. When the selection probability of that individual is not less than the random probability, the current parent's individual is selected to the offspring. Conversely, the current individual is discarded. This rule is to select the excellent parent individuals into the offspring. The formula for the selection rule is as follows:

$$P_{sj}=\frac{fit_j}{\sum_{n=1}^{N} fit_n}$$
$$if \quad P_{rand}\le P_{sj}, then \quad Parent \quad Generation[i_j]\rightarrow Filial \quad Generation[i_j] \tag{7}$$

where P_{sj} is the selection probability of an individual i_j , fit_j is the fitness value of an individual i_j , fit_n is the fitness value of an individual fit_n , P_{srand} is a random number in the range [0, 1], $Parent \quad Generation[i_j]$ denotes an individual i_j in the parent generation, and denotes $Filial \quad Generation[i_j]$ an individual i_j selected into the offspring.

(2) Crossover Rule

The crossover rule utilizes the exchange of gene fragments from two individuals to obtain offspring individuals with different fitness. This rule simulates the genetic recombination process in nature so that the two individuals cross to generate two new individuals with different fitness with a certain probability. This rule improves the diversity of the algorithm and ensures that the algorithm does not fall into local optimal solutions. The crossover rule is divided into three processes: crossover parameter calculation, crossover probability determination, and crossover operation. The crossover parameter calculation firstly obtains the category probability through the ratio of the number of individuals in the category to the total number of individuals in the current population and then calculates the population entropy of the current population from this probability to obtain the function, and utilizes the maximum crossover probability, the minimum crossover probability, and the function to obtain the crossover parameter in the population. The crossover probability is determined by the relationship between the individual fitness value and the average fitness of the individuals in the population. Crossover operation is the operation of linear recombination of gene segments of two individuals to obtain a new individual when the crossover probability of two individuals is not less than the random probability. The crossover rule is calculated by the following formula:

$$\mathrm{P_d}=\frac{\mathrm{n_d}}{\mathrm{N}} \qquad \mathrm{H}=-\sum_{\mathrm{d}=1}^{\mathrm{C}}\mathrm{P_d}\log_2\mathrm{P_d} \quad （\mathrm{C}\le\mathrm{N}）$$
$$\mathrm{U}(\mathrm{t})=\frac{2\mathrm{H}}{\log_2(1+\mathrm{t})\log_2\mathrm{N}}-1 \qquad \mathrm{r_c}=\frac{\mathrm{P_{c\max}}-\mathrm{P_{c\min}}}{1+\exp（-\mathrm{aU}(\mathrm{t})）}+\mathrm{P_{c\ min}} \tag{8}$$
$$\overline{fit}=\frac{1}{\mathrm{N}}\sum_{\mathrm{n}=1}^{\mathrm{N}} fit_{\mathrm{n}} \qquad fit_{\max}=\max_{1\le\mathrm{n}\le\mathrm{N}} fit_n$$
$$P_{c\mathrm{j}}=\begin{cases} r_c\dfrac{fit_{\max}-fit_{\mathrm{j}}}{fit_{\max}-\overline{fit}} & fit_j\ge\overline{fit} \\ r_c & fit_j<\overline{fit}\end{cases} \quad P_{ck}=\begin{cases} r_c\dfrac{fit_{\max}-fit_{\mathrm{k}}}{fit_{\max}-\overline{fit}} & fit_k\ge\overline{fit} \\ r_c & fit_k<\overline{fit}\end{cases}$$
$$\mathrm{if} \quad \mathrm{P_{crand}}\le\mathrm{P_{cj}}，\ \mathrm{then}\ \ \mathrm{i_j}\rightarrow\mathrm{i_j'} \qquad \mathrm{if} \quad \mathrm{P_{crand}}\le\mathrm{P_{ck}}，\ \mathrm{then}\ \ \mathrm{i}_k\rightarrow\mathrm{i}_k'$$
$$\mathrm{u_j'}=\alpha\mathrm{u_j}+(1-\alpha)\mathrm{u_k} \qquad \mathrm{u}_k'=\alpha\mathrm{u}_k+(1-\alpha)\mathrm{u_j}$$

where P_d is the category probability, which is equal to n_d divided by N, n_d is the number of individuals in category d, H is the population entropy of all individuals in the population, U (t) is the value of the function, a is a given constant, t is the number of current evolutionary generations, $\mathrm{r_c}$ is a parameter in the crossover rule, $P_{c\max}$ and $P_{c\min}$ are the maximum crossover probability and the minimum crossover probability, fit is the average fitness value of all individuals in the population, $\mathrm{fit}_{\max}$ is the maximum fitness of all individuals in the population values, $P_{c\mathrm{j}}$ and P_{ck} are the crossover probabilities of two individual sums, respectively, P_{crand} is random numbers in the range [0, 1], if $P_{c\mathrm{j}}$ and P_{ck} are not less than P_{crand} , then the two individuals i_j' and i_k crossover in order to generate two new individuals $_j$ and i_k , and α are random numbers in the range [0, 1].

(3) Mutation Rule

The mutation rule generates new offspring by mutating the parent individual. This rule simulates the process of genetic mutation in nature to generate offspring with a lower mutation rate than the parent. This rule allows the introduction of offspring with different individual fitness and also increases the diversity of the algorithm, ensuring that the algorithm does not fall into a local optimum. The mutation rule is divided into three processes: mutation parameter calculation, mutation probability determination, and mutation operation. Calculation of mutation parameter firstly obtains the category probability through the ratio of the number of individuals in the category to the number of individuals in the current population, then obtains the group entropy of all individuals in the current population by probability calculation, and then obtains the function, and obtains the mutation parameter by utilizing the maximum probability of mutation, minimum probability of mutation, and the function of mutation in the population. The variation probability is determined by the relationship between the individual fitness value and the average fitness in the population. The mutation operation is to let each gene segment on an individual get the same increment to get a new individual when the probability of mutation of the individual is not less than the random probability. The formula for the mutation rule is as follows:

$$
\begin{gathered}
P_d=\frac{n_d}{N} \qquad H=-\sum_{d=1}^{C} P_d \log_2 P_d \quad （C \le N）\\
U(t)=\frac{2H}{\log_2（1+t）\log_2 N}-1 \qquad r_m=\frac{P_{m\ \max}-P_{m\ \min}}{1+\exp（-aU(t)）}+P_{m\ \min} \qquad (9)\\
\overline{fit}=\frac{1}{N}\sum_{n=1}^{N} fit_n \qquad fit_{\max}=\max_{1\le n\le N} fit_n\\
P_{mj}=\begin{cases} r_m \dfrac{fit_{\max}-fit_j}{fit_{\max}-\overline{fit}} & fit_j \ge \overline{fit}\\ r_m & fit_j < \overline{fit} \end{cases}\\
\text{if} \quad P_{mrand} \le P_{mj}, \quad \text{then} \quad i_j \to i_j^{'}\\
i_j^{'}=i_j+\beta(1,1,1,1)_{1\times 4}
\end{gathered}
$$

where P_d is the category probability, which is equal to n_d dividing by N, n_d is the number of individuals in category d, H is the population entropy of all individuals in the population, U(t) is the function value, a is a given constant, t is the current evolutionary generation, r_m is a parameter in the mutation rule, fit is the average fitness value of all individuals in the population, fit_{max} is the maximum fitness value of all individuals in the population, P_{mi} is the mutation probability of an individual u_i , $P_{m\max}$ and $P_{m\min}$ are the maximum mutation probability and minimum mutation probability, if P_{mi} not less than P_{mrand} , the individual u_i is mutated to generate a new individual u_i , β is a random number in the range [0, 1].

4) SIMULATED ANNEALING OPTIMIZATION

Simulated annealing optimization is the use of simulated annealing algorithm to optimize the four optimal weighting factors obtained by the genetic algorithm. Simulated annealing optimization is divided into three processes: parameter initialization, iterative search, and termination of output. The parameter initialization process sets the initial temperature, the number of iterations, the cooling factor, and the termination temperature. The iterative search process perturbs the current individual at the current temperature to obtain a new individual, calculates the energy difference between the old and new individuals, and decides whether to accept the new solution as the current solution according to the Metropolis criterion. The termination output process reduces the temperature at the end of the iteration, repeats the iterative search process, terminates the algorithm when the current temperature is less than the minimum temperature and outputs the current solution as the optimal solution. Using simulated annealing algorithm to optimize the initial population can effectively avoid the population from falling into local optimum.

Optimization of four weighting factors which have been optimized by genetic algorithm using simulated annealing algorithm is given in Algorithm 1. Line 1 performs initialization operations on the parameters such as initial temperature $initial\_temp$, annealing factors $cooling\_rate$, minimum temperature $cooling\_rate$, and number of iterations $search\_times\_per\_temp$. In lines 3 to 9, first set the initial temperature to the current temperature, if the current temperature is greater than the minimum temperature, then the current individual $i_{current}$ is perturbed to get the neighboring individual i_{new} , the new individual corresponding to the weighting factor new_weights and the theme of the text similarities to get the energy of the new individual new_power. if the energy of the new individual is greater than the current individual or the generated random number random is less than the probability exp ($\Delta power$ / temp), then the weighting factor of the new individual and the individual energy are assigned to the current individual. Lines 11 to 15, cool the temperature temp by cooling_rate factor, if the energy of the new individual new_power is greater than the optimal individual best_power, then the weighting factor and individual energy of the new individual will be assigned to the optimal individual. Finally, the optimal weighting factor best_weights and the maximum energy best_power are output when the current temperature temp is lower than the minimum temperature min_temp. $\lambda_{k1}, \lambda_{k2}, \lambda_{k3}, \lambda_{k4}$ is the four weighting factors corresponding to the individual, $power$ is the objective function, $temp$ is the temperature in the simulated annealing, $power_{current}$ is the minimum temperature, $power_{current}$ and $power_{current}$ represent the objective function values of the current individual and the perturbed neighboring individual, respectively, $\Delta power$ is the energy difference between the current individual and the perturbed domain individual, i_{best} and $power_{best}$ represent the optimal individual and its corresponding objective function value, respectively.

Algorithm 1: The SA method optimizing four factors

```
Input: (1) The individual optimized by genetic algorithm is $i_{current}$ , corresponding to four weighting factors: $\lambda_{k1}, \lambda_{k2}, \lambda_{k3}$ and $\lambda_{k4}$

(2) The topic similarities of the four texts corresponding to full texts, anchor texts, title texts and context texts including these N training hyperlinks are respectively described as $sim_1, sim_2, sim_3$ and $sim_4$

Output: The four optimal weighting factors included in individual $i_{best}$ are $\lambda_1, \lambda_2, \lambda_3$ and $\lambda_4$
01.Initialize initial_temp, cooling_rate, min_temp, search_times_per_temp
02. #step1: Parameter initialization
03. temp = initial_temp
04. while temp > min_temp:
```

```
$i_{new}$ = neighbor_solution ( $i_{current}$ )
new_power = objective_function(new_weights, similarities)
if new_power > current_power or random< exp ( $\Delta power$ / temp):
current_weights = new_weights
current_power = new_power
#step2: Iterative search
if new_power > best_power:
best_weights = new_weights
best_power = new_power
temp *= cooling_rate
return best_weights, best_power
```

C. URL Priority Prediction

URL priority prediction is to calculate the priority of unvisited URLs by linear integration of two optimal weighting factors and document topic relevance. Firstly, URL priority prediction is to calculate the document topic relevance by using BERT vector space model. Then, the genetic algorithm was used to obtain four optimal weighting factors: the full text, the anchor text, the title text and the text around the paragraph. Finally, the URL priority was predicted by linearly weighting the four optimal weighting factors and the document topic relevance.URL priority prediction is calculated as follows:

$$priority = \lambda_1 sim_1 + \lambda_2 sim_2 + \lambda_3 sim_3 + \lambda_4 sim_4 \quad (10)$$

where $priority$ is the priority value of the URL, λ_1 is the optimal weighting factor of the full text, sim_1 is the document topic relevance of the full text, λ_2 is the optimal weighting factor of the anchor text, sim_2 is the document topic relevance of the anchor text, λ_3 is the optimal weighting factor of the title text, sim_3 is the document topic relevance of the title text, λ_4 is the optimal weighting factor of the text around the paragraph, sim_4 is the document topic relevance of the text around the paragraph.

IV. Experiment

This experiment designs and implements four kinds of focused crawlers: Word2vec based crawler, Elmo based crawler, BERT based crawler and BSVSM crawler. The experimental results further show that the use of BSVSM method can improve the performance of the focused crawler. This part includes two parts: experimental design and experimental results. The experimental design includes the experimental theme crawler, the initial data of the experiment, and the experimental evaluation index. The experimental results include the comparison of course scheduling algorithms, the comparison of communication algebra, and the visualization results. The experimental results show that BSVSM can better obtain the web information with high relevance to the topic.

A. Experimental Design

Experimental design is the design scheme that illustrates the whole experiment. This part includes three contents: experimental theme crawler, experimental initial data, and experimental evaluation index. Three elements of the experimental design are detailed below.

1) EXPERIMENTAL FOCUSED CRAWLER

In this experiment, four different theme crawlers are designed, including Word2vec based crawler, Elmo based crawler, BERT based crawler and BSVSM crawler. The performance of the four theme crawlers is compared through the experimental results. The four focused crawlers are described as follows:

(1) Word2vec Crawler

Word2vec Crawler is a crawler that utilizes the Word2vec model. Word2vec is a model based on neural network training to obtain word vectors. Word2Vec models typically employ two neural network architectures: Skip-gram and Continuous Bag of Words (CBOW). The Skip-gram model tries to predict the context word, while the CBOW model tries to predict the target word. The Word2Vec model maps words from the original space to a new low-dimensional space, so that semantically similar words are close to each other in this space. Therefore, the Word2Vec word vector can be used to measure the similarity between words. Since the distribution of semantically similar words in the vector space is relatively close, the spatial distance between word vectors can be calculated to represent the semantic similarity between words.

(2) ELMO Crawler

ELMO Crawler is a crawler based on the pre-trained language model ELMO. ELMO is a pre-trained language model for solving the polysemy problem. In word2vec and Glove, each word corresponds to a certain fixed vector, so the problem of polysemy cannot be solved. In ELMO, each word no longer corresponds to a fixed vector, and the pre-trained model is no longer just the correspondence between words and vectors. Instead, a model is trained to input a sentence or a paragraph when it is used, and the model will obtain the word vector of a certain word according to the semantic information of the context. One of the advantages of this approach is that it can solve the problem of polysemy of a word, so as to optimize the semantic similarity between terms, so as to improve the text relevance.

(3) BSVSM Crawler

BSVSM Crawler is a crawler based on pre-trained language model BERT. BERT uses the Encoder structure of Transformer to learn language representation and semantic relations by pre-training large-scale corpora. In the pre-training phase, BERT uses a large corpus for training, with the goal of learning language representation and semantic relations. In the fine-tuning phase, BERT is trained using datasets of downstream tasks to adapt to specific task requirements. At this stage, BERT updates only the parameters relevant to the task, leaving the language representation learned during the pre-training phase unchanged. BERT uses a large corpus for pre-training, and it can learn richer language representations and semantic relations, thus making semantic similarity and text relevancy more accurate.

(4) BSVSM-HA Crawler

BSVSM-HA crawler is a semantic and intelligent crawler based on BERT model and hybrid algorithm. BSVSM-HA crawler uses BERT semantic vector space model to calculate the topic relevance of documents, and uses hybrid algorithm to optimize the weighting factor of unvisited URLs priority. First, BERT is used to construct word vectors and evaluate the semantic similarity between terms. The document semantic vector and topic semantic vector are constructed respectively, and the cosine similarity of the two is used to calculate the topic relevance of the document more accurately. Then, by randomly generating a population and calculating the fitness of each individual in the population, the method uses a hybrid algorithm to determine the optimal value of the two weighting factors. Finally, this method obtains the priority of unvisited hyperlinks and predicts the priority of URLs by linearly integrating the document topic relevance and the optimal weighting factor.

2) EXPERIMENTAL INITIAL DATA

The performance of the five focused crawlers was compared by giving them the same subject set and initial data. The more topics a topic set contains, the more convincing the results. In the experiment, the topic set includes 5 different topics: virtual reality, blockchain, artificial intelligence, cloud computing, cybersecurity. The initial data includes a theme page set, a crawling data set, and a training data set. The theme page set describes the topic information, and the content of the page is related to the topic. The crawler dataset contains the initial URLs of different topics, and the initial URL of each topic initiates the focused crawler to crawl the web pages related to that topic. The training data set includes training data, test data and training parameters for different topics. The maximum number of downloaded web pages is not more than 5000.

A theme page set can be used to calculate the topic similarity between a web page and a theme. This collection of pages can be accessed by web crawlers. In the experiment, in order to reduce the time complexity, the size of the theme web page was set to 20. First, enter 5 different topics into a general search engine such as Bing or Google to get a large number of web pages related to the topic. These retrieved pages have been sorted by topic similarity, and the most relevant pages will be at the top of the results list. The experiment then directly logs the top 20 URLs in the results list into the specified file. Finally, use the crawler to download the theme web page that is set for each theme by specifying the file.

The crawling dataset includes the initial URLs for different topics. This dataset can be used to extract new hyperlinks so that centralized crawlers constantly download web pages from the Internet. Table 1 shows the initial URLs for five different topics, with three different initial URLs for each topic.

The training data set includes training data, test data and training parameters for different topics. The four optimal weighting factors are obtained by using the training data set.

The training data and test data for each topic include the topic similarity of the web pages corresponding to the training and test hyperlinks, as well as the topic similarity of the four texts corresponding to the full text, anchor text, title text, and context text containing these hyperlinks. The experiment obtained 1000 sets of topic similarity for all training and test hyperlinks. Each hyperlink corresponds to a set of topic similarities, with five values for five different texts. In addition, the five different texts of each hyperlink contain the web page corresponding to that hyperlink, and the four text corresponding to the full text, anchor text, title text, and context text in the parent web page containing the hyperlink. The 1000 groups of topic similarity were divided into training data and test data, and the first 500 groups of data and the last 500 groups of data were regarded as training data and test data respectively.

The training data is utilized in the hybrid algorithm to set the training parameters in order to obtain the optimal weighting factors. In the experiment, all the training parameters are given as follows: $P_{c\max}$ =0.7, $P_{c\min}$ =0.3, $P_{m\max}$ =0.5, $P_{m\min}$ =0.1, $initial\_temp$ =100, $cooling\_rate$ =0.99, $search\_times\_per\_temp$ =1, min_temp=0.1 in formula (12) , formula (13) and Algorithm. The maximum number of evolutionary generations of the hybrid algorithm is set to 3000 to achieve convergence. In addition, the individuals in the initial population are randomly generated and each individual corresponds to four weight factors, each ranging from 0 to 1.

3) EXPERIMENTAL EVALUATION INDICATOR

In this experiment, all focused crawler performance is mainly evaluated by the relevant number of web pages, accuracy, and average topic relevance. The relevant number of web pages refers to the number of relevant web pages crawled by the topic crawler, and the accuracy is the ratio of the number of topic-related pages crawled by the topic crawler to the total number of crawled web pages. The average topic relevance is used to evaluate the relevance performance of crawlers crawling web pages. The higher the average topic relevance, the higher the average topic relevance of the crawler, that is, the better the performance of the crawler.

The above three indicators can well evaluate the performance of the course scheduling algorithm, and the calculation expression of these indicators is as follows:

$$R_i \geq th(1 \leq i \leq SP) \qquad AC = \frac{SP}{DP} \qquad AR = \frac{1}{DP} \times \sum_{j=1}^{DP} R(P_j) \tag{11}$$

where R_i is the topic similarity of the page i- th is the parameter that determines whether the web page is relevant to the topic, SP is the number of pages related to the topic crawled by the crawler, DP is the total number of pages crawled by the crawler, AR is the average topic relevancy of the page, $R(P_j)$ is the topic relevancy of the page P_j.

B. Experimental Training Results

This experiment provides training data to obtain the optimal four weight factors using a hybrid algorithm. The hybrid algorithm is based on the training data and convergent evolutionary generation from the experiments using genetic optimization and simulated annealing optimization for the four weighting factors for four different texts. The hybrid algorithm gives convergent evolutionary generation for all the different topics. In the experiment, the optimal object is considered as the optimal four weight factors for each topic when the number of evolutionary generations reaches 3000. Table 1 shows the optimal weight factors for all five topics based on the hybrid algorithm. In Table 1, for the topic “Virtual Reality”, the optimal weight factors corresponding to the topic contributions of full text, anchor text, title text and context text are (0.053788, 0.028059, 0.012684, 0.286651).

TABLE 1. **Optimal weighting factors for all 5 topics based on hybrid algorithm**

Topics	λ_1	λ_2	λ_3	λ_4
1.Virtual Reality	0.053788	0.028059	0.012684	0.286651
2.Artificial Intelligence	0.253796	0.051969	0.006254	0.042144
3.Blockchain	0.028392	0.058458	0.197910	0.371659
4.Cloud Computing	0.032886	0.133501	0.010908	0.141462
5.Network Security	0.052849	0.060498	0.058671	0.192233

The experiment utilizes test data to compare the above optimal four weighting factors determined by the hybrid algorithm with the four weighting factors determined by the manual method. The manual method utilizes the researcher's experience to determine the four weighting factors, which is arbitrary and subjective. Table 1 shows the four weighting factors determined by the hybrid algorithm for all five topics, while the four weighting factors determined by the manual method are all equal at 0.25. Test data were utilized to compare the four weighting factors determined by the hybrid algorithm and the manual method respectively. The test data contained 500 sets with 500 absolute errors per subject for each method. In order to compare the two methods more clearly, the experiment divided the above 500 groups of test data into 50 groups for each topic, every 10 groups of test data for each topic were combined into one group. Then, each group of test data after combination corresponds to 10 absolute errors, and the average of these absolute errors is regarded as the average absolute error of the group for each topic. Finally, each method obtains 50 mean absolute errors for each topic after combining the test data.

Fig. 3 shows the mean absolute errors based on the two methods for all the different topics. In Fig. 3, for all 50 sets of test data, the average absolute error determined by the hybrid algorithm is significantly smaller than the average absolute error determined by the manual method. Thus, the test results indicate that the four weighting factors determined by the hybrid algorithm are more effective than the manual method. The training results in the experiment were used to obtain the optimal weighting factors. First, the training results show the optimal weighting factors for all the different topics using the hybrid algorithm. In addition, the test results show that the hybrid algorithm is able to obtain more accurate four weighting factors than the manual method. The optimal weighting factors obtained based on the hybrid algorithm will be used to predict the priority of unvisited hyperlinks for all different topics.

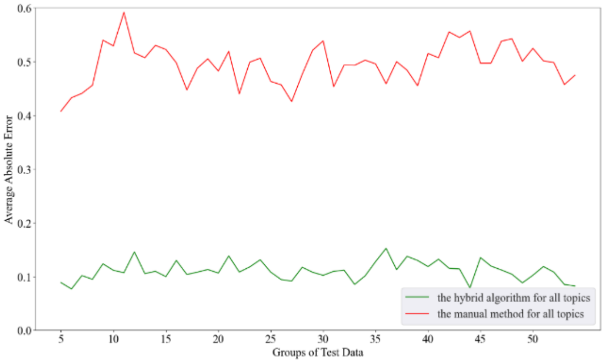


FIGURE 3. **The average absolute errors for all different topics based on the hybrid algorithm and the manual method**

C. Experimental Crawling Results

In this paper, four focused crawlers are used to crawl the initial URLs of five different topics to obtain the experimental crawling results. Table 2 shows the crawling results of four focused crawlers. In Table 2, the relevant number of Web pages (RN), the accuracy of crawled Web pages (AC), and the average topic relevance of crawled Web pages (AR) are used as the evaluation indexes of crawling results. In addition, for all five topics in Table 2, the number of retrieved pages starts from 100 until 5000 web pages are retrieved, and the trial metrics of each topic crawler are recorded separately to detect the crawling results.

TABLE 2. **The crawling results including AC, HR and AR for five focused crawler**

Numbers of retrieved pages	Word2vec Crawler			ELMO Crawler			BSVSM Crawler			BSVSM-HA Crawler		
	RN	AC	AR	RN	AC	AR	RN	AC	AR	RN	AC	AR
100	76	0.76	0.580	95	0.95	0.703	82	0.82	0.604	67	0.67	0.513
200	150	0.75	0.586	157	0.785	0.591	156	0.78	0.606	153	0.765	0.602
300	196	0.653	0.537	201	0.67	0.531	199	0.663	0.550	202	0.673	0.546
400	245	0.612	0.510	248	0.62	0.504	246	0.615	0.516	268	0.67	0.531
500	317	0.634	0.510	319	0.638	0.495	305	0.61	0.505	325	0.65	0.516
600	391	0.651	0.519	396	0.66	0.496	383	0.638	0.502	375	0.625	0.501
700	453	0.647	0.512	482	0.688	0.506	439	0.627	0.492	468	0.668	0.500
800	535	0.668	0.510	540	0.675	0.501	533	0.666	0.514	568	0.71	0.498
900	605	0.672	0.507	612	0.68	0.498	611	0.678	0.511	649	0.721	0.497
1000	685	0.685	0.506	707	0.707	0.502	699	0.699	0.509	718	0.718	0.496
1500	1047	0.698	0.492	1043	0.695	0.484	1164	0.776	0.505	1183	0.788	0.515
2000	1233	0.616	0.460	1258	0.629	0.455	1361	0.680	0.462	1451	0.725	0.491
2500	1463	0.585	0.445	1486	0.594	0.441	1602	0.640	0.448	1713	0.685	0.475
3000	1732	0.577	0.440	1721	0.573	0.431	1855	0.618	0.441	1985	0.661	0.464
3500	2102	0.600	0.442	1982	0.566	0.427	2120	0.605	0.438	2267	0.647	0.459
4000	2385	0.596	0.439	2243	0.560	0.423	2379	0.594	0.436	2519	0.629	0.452
4500	2687	0.597	0.440	2521	0.560	0.422	2657	0.590	0.436	2770	0.615	0.447
5000	3001	0.600	0.440	2805	0.561	0.421	2930	0.586	0.436	3048	0.609	0.445

Fig.4 shows the acquisition of themed web pages by four themed crawlers. In Fig.4, BSVMS-HA crawlers obtained more related web pages than the other three themed crawlers. It can be concluded that BSVSM-HA crawlers can obtain more themed web pages. Fig. 5 shows the accuracy rate of the four theme crawlers to crawl the theme web page. In Fig. 5, the accuracy rate of the BSVSM-HA crawler to crawl the theme web page is higher than that of the other three theme crawlers, so the BSVSM-HA crawler can crawl the theme web page more accurately. Fig.6 shows the average topic relevance of the four theme crawlers to crawl web pages. In Fig. 6, the average relevance of BSVSM-HA crawlers to crawl web pages with higher topic relevance is the highest. It can be seen that BSVSM-HA crawlers can crawl web pages with higher topic relevance.

The crawling results of four kinds of focused crawlers were obtained. First of all, the crawling results show that BSVSM-HA crawlers can obtain more subject-related web pages and have higher subject-related degree than W2V, ELMO and BSVSM crawlers. Secondly, the crawling results show that BSVSM-HA crawler uses BERT model to calculate topic relevance, which can make crawler have better semantic, so that crawler can obtain more and better topic related web pages. Finally, the crawling results show that the BSVSM-ha crawler with 4 weight factors optimized by the hybrid algorithm can obtain more and better topic-related web pages than the other three crawlers with 4 weight factors determined by the manual method.

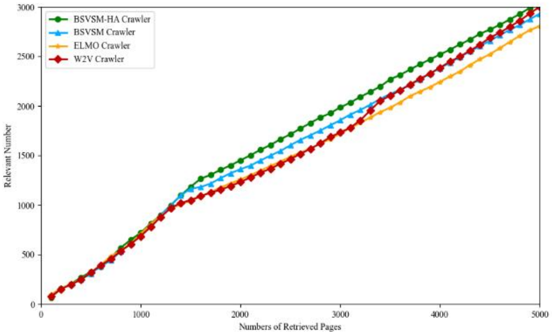


FIGURE 4. **Comparison of the relevant number of web pages of four focused crawlers**

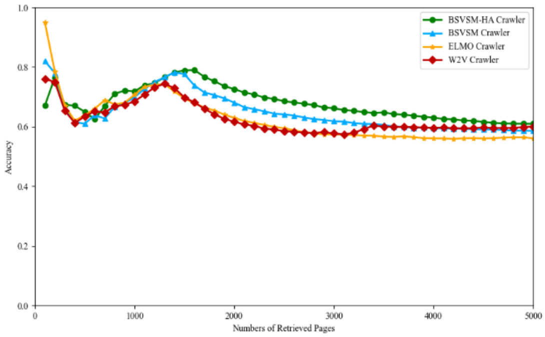


FIGURE 5. Comparison of the accuracy of crawling web pages by four focused crawlers

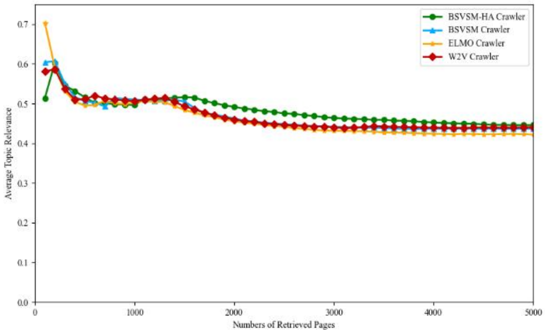


FIGURE 6. Comparison of the average topic relevance of web pages crawled by four focused crawlers

V. CONCLUSION AND FUTURE WORK

In this paper, a semantic and intelligent focused crawler based on BERT semantic vector space model and hybrid algorithm is proposed. This method used BERT semantic vector space model to calculate the topic relevance of documents, and used a hybrid algorithm to optimize the weighting factor of unvisited URL priority. Firstly, BERT is used to construct word vectors and evaluate the semantic similarity between words. The document semantic vector and the topic semantic vector are constructed respectively, and the cosine similarity between them is used to calculate the topic relevance of the document more accurately. Then, by randomly generating a population and calculating the fitness of each individual in the population, the hybrid algorithm was used to determine the optimal values of the four weighting factors. Finally, the priority of unvisited hyperlinks was obtained by linear integration of document topic relevance and optimal weighting factor, and the priority of URLs was predicted. The experimental results show that BSVSM-HA crawler can crawl more subject-related web pages than Word2vec crawler, ELMO crawler and BSVSM crawler. In conclusion, the semantic and intelligent crawler proposed in this paper makes the semantic similarity between terms more accurate, and improves the topic relevance of the text, and the optimized weighting factor makes the priority evaluation of unvisited URLs more accurate.

There are still worthy of further research work in this paper. In this paper, BERT is used to calculate the semantic similarity between words, and the semantic vector and document vector are constructed. The semantic vector and document vector constructed by this method depend on the BERT model, and the effect of using BERT to evaluate the semantic similarity of words is not as good as using BERT to evaluate the semantic similarity between sentences. Therefore, it is better to use a large language model to calculate the document vector for the entire document. In addition, the method of using the hybrid algorithm to optimize the four weighting factors in this paper still has the problem of falling into the local optimal solution, and the cuckoo algorithm can be introduced into the process of population initialization. This prevents the algorithm from falling into local optimal solutions and improves the search efficiency.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61070015), Guangdong Provincial Key Technology Research Center for Object Digitization and Epidemic Intelligence Prevention and Control, Guangdong Provincial Engineering Technology Research Center for General Colleges and Universities, Guangdong Provincial Department of Education (Grant No. 2022GCZX013), the Intelligent Policing Key Laboratory of Sichuan Province (Grant No. ZNJW2024KFQN012), the Open Research Subject of Sichuan Provincial Engineering Research Center of Hydroelectric Energy Power Equipment Technology (Grant No. SDNY-2024001), and the College Student Innovation and Entrepreneurship Training Project of Sichuan Province (Grant No. D202405310751003590).

REFERENCES

[1] P. R. J. Dhanith, K. Saeed, G. Rohith, et al. Weakly supervised learning for an effective focused web crawler. Engineering Applications of Artificial Intelligence. 2024, 132.

[2] S. Jamil, A. M. Roy. An efficient and robust phonocardiography (PCG)-based valvular heart diseases (VHD) detection framework using vision transformer (ViT). Computers In Biology And Medicine. 2023, 158.

[3] X. Wang, Z. C. Chen, M. M. Kong, et al. A hunger-based scheduling strategy for distributed crawler. Expert Systems With Applications. 2023, 222.

[4] N. Kumar, D. Aggarwal. LEARNING-Based focused WEB crawler. Iete Journal Of Research. 2023, 69(4): 2037-2045.

[5] S. Masters, B. Anthoons, P. Madesis, et al. Quantifying an online wildlife trade using a web crawler. Biodiversity And Conservation, 2022, 31(3): 855-869.

[6] S. S. Alqarale, H. M. N. Sirin. A Topic-Specific Web Crawler using Deep Convolutional Networks. International Arab Journal of Information Technology, 2022, 20(3): 310-318.

[7] W. J. Liu, Z. R. Gan, T. J. Xi, et al. A Semantic and Intelligent Focused Crawler based on Semantic Vector Space Model and Membrane Computing Optimization Algorithm. Applied Intelligence, 2022, 53(7): 7390-7407.

[8] J. D. P. N. R. Mary, S. Balasubramanian, R. S. P. Raj. A Critique Empirical Evaluation of Relevance Computation for Focused Web Crawlers. Brazilian Archives of Biology and Technology, 2021, 64.

[9] W. J. Liu, Y. He, J. Wu, et al. A Focused Crawler based on Semantic Disambiguation Vector Space Model. Complex & Intelligent Systems, 2022, 9(1): 345-366.

[10] P. R. J. Dhanith, B. Surendiran, SP. Raja. A Word Embedding Based Approach for Focused Web Crawling Using the Recurrent Neural Network. International Journal of Interactive Multimedia and Artificial Intelligence, 2021, 6(6): 122-132.

[11] W. Wang, L. H. Yu. UCrawler: A learning-based web crawler using a URL Knowledge base. Journal of Computational Methods in Sciences and Engineering, 2021, 21(2): 461-474.

[12] J. D. P. N. R. Mary, S. Balasubramanian, RSP. Raj. A Critique Empirical Evaluation of Relevance Computation for Focused Web Crawlers. Brazilian Archives of Biology and Technology, 2021, 64.

[13] J. F. Liu, Z. H. Yang, X. M. Yan, et al. Applying particle swarm optimization-based dynamic adaptive hyperlink evaluation to focused crawler for meteorological disasters. Complex & Intelligent Systems. 2024, 10(1): 233-255.

[14] A. Lagopoulos, G. Tsoumakas. Content-aware web robot detection. Applied Intelligence, 2020, 50(11): 4017-4028.

[15] W. J. Liu, Z. R. Gan, T. J. Xi, et al. A Semantic and Intelligent Focused Crawler based on Semantic Vector Space Model and Membrane Computing Optimization Algorithm. Applied Intelligence, 2022, 53(7): 7390-7407.

[16] K. S. S. Prabha, C. Mahesh, SP. Raja. An Enhanced Semantic Focused Web Crawler Based on Hybrid String Matching Algorithm. Cybernetics and Information Technologies, 2021(2): 105–120.

[17] A. M. Roy, J. Bhaduri. DenseSPH-YOLOv5: an automated damage detection model based on DenseNet and Swin-Transformer prediction head-enabled YOLOv5 with attention mechanism. Advanced Engineering Informatics. 2023, 56.

[18] W. Wang, L. H. Yu. UCrawler: A learning-based web crawler using a URL knowledge base. Journal Of Computational Methods in Sciences and Engineering, 2021, 21(2): 461-474.

[19] J. F. Liu, F. Li, R. Y. Ding, Focused crawling strategies based on ontologies and simulated annealing methods for rainstorm disaster domain knowledge. Frontiers of Information Technology & Electronic Engineering, 2022, 23(8): 1189-1204.

[20] W. J. Liu, Y. He, J. Wu, et al. A Focused Crawler based on Semantic Disambiguation Vector Space Model. Complex & Intelligent Systems, 2022, 9(1): 345-366.

[21] W. Wang, L. H. Yu. UCrawler: A learning-based web crawler using a URL

Knowledge base. Journal of Computational Methods in Sciences and Engineering, 2021, 21(2): 461-474.

[22] S. Rajiv, C. Navaneethan. A supervised learning-based approach for focused web crawling for IoMT using global co-occurrence matrix. Expert Systems. 2023, 40(4).

[23] G. K. Shrivastava, R. K. Pateriya, P. Kaushik. An efficient focused crawler using LSTM-CNN based deep learning. International Journal of System Assurance Engineering and Management. 2023, 14(1): 391-407.

[24] P. R. J. Dhanith, B. Surendiran, SP. Raja. A Word Embedding Based Approach for Focused Web Crawling Using the Recurrent Neural Network. International Journal of Interactive Multimedia and Artificial Intelligence, 2021, 6(6): 122-132.

[25] A. M. Roy, R. Bose, V. Sundararaghavan, et al. Deep learning-accelerated computational framework based on physics informed neural network for the solution of linear elasticity. Neural Networks. 2023, 162: 472-489.

[26] J. F. Liu, Y. Dong, Z. X. Liu, et al. Applying ontology learning and multi-objective ant colony optimization method for focused crawling to meteorological disasters domain knowledge. Expert Systems with Applications. 2022, 198.

[27] C. H. Liu, S. D. You, Y. C. Chiu. A Reinforcement Learning Approach to Guide Web Crawler to Explore Web Applications for Improving Code Coverage. ELECTRONICS. 2024, 13(2).

[28] J. F. Liu, F. Li, R. Y. Ding, Focused crawling strategies based on ontologies and simulated annealing methods for rainstorm disaster domain knowledge. Frontiers Of Information Technology & Electronic Engineering, 2022, 23(8): 1189-1204.

WenHao Huang. the PhD candidate of South China University of Technology, received the M.E degree from South China University of Technology and B.S degree from Yangtze University. His research interests are mainly on Intelligent manufacturing, Internet of Things, Deep learning, machine learning and software engineering.

Jiahao Zhang.currently studying computer science and technology at Xihua University. His research interests include Text Clustering and Deep Learning.

Xin Li. currently studying computer science and technology at Xihua University, mainly focuses on Artificial Intelligence and Heuristic Optimization Algorithm.

Xiao Zhou is currently studying in computer science and technology at Xihua University. His main research fields include Ontology Construction and Artificial Intelligence.

Deyu Qi. received the M.S. degree from National University of Defense Technology and Ph.D. degree from South University of Technology. He received the M.S. degree from the National University of Defense Technology, in 1988, and the Ph.D. degree, in 1992. He is currently a Full Professor with the South China University of Technology and the Head of the Infrastructure Software and Application Construction Technology Laboratory of Guangdong Province. His research interests include cloud computing platform, parallel scheduling, and software architecture.

JianQin Xi. received the M.S. degree from the National University of Defense Technology, in 1988, and the Ph.D. degree, in 1992. He is currently a Full Professor with the South China University of Technology and the Head of the Infrastructure Software and Application Construction Technology Laboratory of Guangdong Province. His research interests include cloud computing platform, parallel scheduling, and software architecture.

Liu Wenjun. currently studying for a doctor's degree in computer science and technology at the University of Electronic Science and Technology of China, graduated from Xihua University with a master's degree in computer software and theory, and graduated from Yangtze University with a bachelor's degree. His research fields include artificial intelligence, natural language processing, and social network analysis.