Available online at www.ijpe-online.com

**International Journal of Performability Engineering**

vol. 15, no. 10, October 2019, pp. 2645-2656
DOI: 10.23940/ijpe.19.10.p10.26452656

# An Improved Focused Web Crawler based on Hybrid Similarity

Songtao Shang*, Huaiguang Wu, and Jiangtao Ma

*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou, 450002, China*

**Abstract**

Web crawler is an efficient strategy for downloading data automatically from the Internet. Focused web crawler is a special kind of web crawler that is responsible for getting certain information from webpages and making them available to users. The most important problem of focused web crawler is to confirm the similarity between the target webpages and the topics. Therefore, this paper proposes an improved focused web crawler algorithm, whose similarity calculating methods derive from three aspects: anchor text, content, and structure of the webpages. This improved algorithm is called hybrid similarity. If the anchor text similarity is bigger than the threshold, the target webpages are downloaded directly; otherwise, the target webpages' similarity is analyzed by using the TF-Gini feature weighting algorithm and the improved cosine similarity algorithm. The experimental results in this paper have proven that the hybrid similarity algorithm is more effective than the traditional algorithm. The precision increases by nearly 10% compared with the traditional algorithm.

*Keywords*: focused web crawler; TF-Gini; similarity; hybrid similarity

(Submitted on June 21, 2019; Revised on July 16, 2019; Accepted on August 11, 2019)



## 1. Introduction

With the development of the Internet and social media, accessing needed information is becoming more convenient as long as one is connected to the Internet. However, in the age of information explosion, how to get the exact information that is needed is very difficult. In China, there are 5.06 million websites [1] that provide different services for different users. Therefore, a quick, convenient, and automatic information retrieval method is necessary for users. General web crawler (GWC), which is widely used in search engines, retrieves all hyperlinks start from the seed page, automatically downloads information from websites, and makes the web easier to use for millions of people [2]. However, GWC is not suitable for most users, since it retrieves every hyperlink from the hyperlink list regardless of users' specific needs. Focused web crawler (FWC) [3-4] is a special GWC that retrieves the hyperlinks that relate to some certain topics. Hence, the most important aspect of FWC is to confirm the similarity between target webpages and topics.

The working theory of FWC [5] is that the crawler is guided by a list of hyperlink seeds, and new hyperlinks related to the topic are continuously put into the list while the crawler crawls recursively on the hyperlink list. The FWC does not stop working until the hyperlink list is empty. Therefore, the key step of FWC is to calculate the similarity between target webpages and topics. The pages that have a value of similarity larger than the threshold should be downloaded by FWC.

Fish Search [6] is a classical FWC algorithm that uses Boolean variables to express the similarity, i.e., 0 is irrelevant and 1 is relevant. The Shark Search [7] algorithm is an aggressive version of Fish Search. It has a sophisticated concept of potential scores for the links in the algorithm, and it is influenced by anchor text, link text, and inherited score from ancestors. These two algorithms only take into consideration the page's contents and anchor text, ignoring the influence of the hyperlink structure of the web pages. Zhang [8] and Wang [9] improved the algorithms by taking into account the hyperlink structure of the web page and splitting the web page into blocks based on the page's structure. The similarity was comprised of the parent page's value, the hyperlink's value, and the block's value.

* Corresponding author.
*E-mail address*: songtao.shang@zzuli.edu.cn

PageRank [10] is another FWC algorithm. This algorithm is a centrality measure that represents the probability of walking along a hyperlink from one page to another. The value of PageRank presents the similarity between the target pages and topics. The crawler will crawl the hyperlinks whose PageRank values are larger than the threshold. PageRank takes the Internet into consideration as a network and improves the performance of the Shark Search algorithm. However, the crawler does not exactly understand what the hyperlink represents. This will lead to a *theme-drift* phenomenon when the crawler uses PageRank crawling hyperlinks.

Learning crawler is another kind of FWC that uses machine learning algorithms to calculate the similarity between target webpages and topics. Naïve Bayesian is an excellent machine learning algorithm for text classification. Zhou [11] and Altigovde [12] adopted Naïve Bayesian to compute the similarity in FWC and achieved good performance. Other machine learning algorithms are also used in learning crawlers, including support vector machines [13], decision trees [14], neural networks [15], and so on. Learning crawler can effectively infer topics from webpages through machine learning algorithms, but it neglects the influence of hyperlink and structure of webpages.

This paper proposes an improved FWC algorithm based on hybrid similarity. The hybrid similarity consists of three aspects: anchor text, content, and structure of the webpage. Anchor text, the brief of target webpages, is the most important for FWC. This paper uses semantic similarity to calculate the similarity between anchor text and topics. Most of the webpages' topics can be recognized through anchor text. As for the other complex webpages, this paper analyzes the entire content to find the webpages' topics. Among them, feature weighting is the key step for similarity computing. This paper proposes a TF-Gini feature weighting algorithm for webpages' similarity computing. In addition, the structure of webpages is also important for similarity computing. We should dynamically regulate the features' weights according to their position on the webpage. Therefore, this paper combines these three algorithms to form hybrid similarity. The experimental results in this paper prove the effectiveness of hybrid similarity.

## 2. The Improved Hybrid Similarity

### *2.1. Anchor Text Similarity*

Anchor text is the visible, clickable text in a hyperlink. In webpages, it is often under-lined, visible characters and words that hyperlinks display when linking to another location on the web, which can provide users relevant contextual information about the content of the link's destination. In other words, anchor text is a brief summary of the target webpage that points out the topic of the webpage.

In modern browsers, anchor text can be highly ranked in FWC algorithms, because the anchor text is usually relevant to the target webpages. The hyperlink's key words of anchor text serve the purpose of directing users to the target websites. Hence, in order to take full advantage of the anchor text, we use the semantic similarity algorithm [16]. Semantic similarity is developed from linguistic research, which is based on topology of ontologies. HowNet [17] is an ontology vocabulary offering a taxonomic hierarchy of natural language terms, which provides broad coverage of the English and Chinese vocabulary. It provides corresponding concepts for as many key words as possible, which is convenient for word sense disambiguation. Therefore, semantic similarity is the best algorithm for similarity computing, especially under the circumstance that the length of anchor text is limited.

According to Resnik's [18] information theory model, there are two main concepts in HowNet, *sememe* and *concept*. *Sememe* is the basic semantic unit that cannot be reduced further. *Concept* is described by a specific knowledge representation language and is constituted by *sememe* and symbol. Symbol is a special symbol defined by HowNet. Therefore, the semantic similarity of two *sememes* is defined in Equation (1).

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} (-\log(p(c)) \tag{1}$$

Where $c_1$ and $c_2$ are *sememes* in HowNet, $S(c_1, c_2)$ is a set containing $c_1$ and $c_2$, and $p(c)$ is the probability of $c$ being in the corpus.

In Equation (1), the probability computing method still relies on the corpus. The cost of computing will be expensive when the corpus is huge. Therefore, this paper improves the computing method. We only use HowNet to compute the probability, as shown in Equation (2).

$$p(c) = 1 - \frac{\log(hy(c)+1)}{\log(\max_{hn})} \tag{2}$$

Where $hy(c)$ is the number of *sememes* that *c* contains and $\max_{hn}$ is the total number of *sememes* that HowNet contains. In this way, the similarity between two *sememes* only relies on the HowNet. The computation cost sharply decreases.

*Concept* usually contains a number of *sememes*. Therefore, the semantic similarity of *concept* can be computed by the *sememes*.

Suppose *concept* $p_1$ contains *n sememes*, i.e., $p_1 = \{c_{11}, c_{12}, \cdots, c_{1n}\}$, and *concept* $p_2$ contains *m sememes*, i.e., $p_2 = \{c_{21}, c_{22}, \cdots, c_{2m}\}$. The similarity between $p_1$ and $p_2$ is shown in Equation (3).

$$sim(p_1, p_2) = \frac{\min(|p_1|, |p_2|)}{\sqrt{|p_1|, |p_2|}} \sum_{i=1}^{|c|} \sum_{j=1}^{|c|} \max_{c \in s(c_i, c_j)} \left(1 - \frac{\log(hy(c)+1}{\log(\max_{hn})}\right) \tag{3}$$

All the keywords in anchor text and topics can be replaced by *concepts* in HowNet. Hence, we can use the weighted mean of two group of *concepts* instead of the similarity between anchor text and topics, as shown in Equation (4).

$$sim(A,T) = \frac{\sum_{i=1}^{n} sim(p_{Ai}, p_{Ti})}{n} \tag{4}$$

Where $sim(A,T)$ represents the similarity between the anchor text and topic. $p_{Ai}$ is the $i^{\text{th}}$ concept in the anchor text, $p_{Ti}$ is the $i^{\text{th}}$ concept in the topic, and n is the total number of concepts.

Normally, the length of anchor text is limited, and the information the anchor text contains is also limited. The similarity computing method in Equation (4) can distinguish the semantic differences, and the results will be more accurate. Since most of the webpages' topics can be deduced by anchor text, the semantic similarity can sharply increase the performance of FWC.

### *2.2. Webpage Similarity*

In many cases, it is enough for FWC to use anchor text similarity to evaluate the similarity between target webpages and topics. However, there is still a large number of complex webpages on the Internet. Under these circumstances, it is necessary for FWC to take some measures into consideration. In this paper, we use the entire content of the target webpages for computing the similarity. This part is a supplement for FWC, which can increase the accuracy of FWC.

Webpages usually contain text, such as news websites, video websites, etc. The topics are also expressed by text, such as sports news, political news, comedy film, and so on. Cosine similarity [19] is a popular algorithm for computing text similarity. Cosine similarity is based on VSM (vector space model) [20]. In VSM, the text can be expressed as $T = \{< t_1, w_1 >, < t_2, w_2 >, \cdots, < t_i, w_i >, \cdots < t_n, w_n >\}$, where $t_i$ is the $i^{\text{th}}$ feature word in the text, $w_i$ is $t_i$'s weights, and *n* is the total number of feature words. The cosine similarity is defined in Equation (5).

$$\cos(T_1, T_2) = \frac{\sum_{i=1}^{n} w_{1i} \times w_{2i}}{\sqrt{\sum_{i=1}^{n} w_{1i}^2} \sqrt{\sum_{i=1}^{n} w_{2i}^2}} \tag{5}$$

Where $w_{1i}$ is the $i^{\text{th}}$ feature weight in text $T_1$ and $w_{2i}$ is the $i^{\text{th}}$ feature weight in text $T_2$.

The most important problem in Equation (5) is feature weights. TF-IDF [21] is a very popular feature weight computing method in information fields. The definition of TF-IDF is shown in Equation (6).

$$TF-IDF(t,T)=\frac{tf(t,T)}{1+\sum_{i=1}^{n}tf(t_i,T)}\times\log(\frac{N}{n_t}) \tag{6}$$

Where $tf(t,T)$ is the frequency of feature *t* in text *T*, *N* is the total amount of text in the corpus, and $n_t$ is the total amount of text that contains feature *t* in the corpus.

TF (term frequency) is the frequency of feature *t* in text *T*, i.e., $tf(t,T)$ . IDF (inverse document frequency) is the ratio that contains feature *t* in the corpus. In the theory of TF-IDF, IDF is a regulation factor that endows more important features with bigger weights. From the point view of information theory, TF-IDF is an effective feature weighting algorithm. It gives more weights to important features. However, in text classification fields, it is a three-level structure, i.e., *feature-text-category*. The traditional TF-IDF is not suitable for text classification feature weighting. The main shortcoming is that IDF cannot correctly regulate the importance of features. Therefore, this paper proposes an improved feature weighting algorithm that can endow appropriate weights based on the features' importance.

*2.2.1. Traditional Gini Index Algorithm*

The Gini index is an attribute impurity measurement method that is widely used in decision tree algorithms [22]. Decision tree algorithms use the Gini index to select the optimal attributes. The Gini index is a lightweight attribute importance computing method. It is faster than most algorithms based on entropy.

Suppose there are *K* categories in the corpus and the probability of feature *p* belonging to *k* is $p_k$ . Then, the definition of the Gini index is shown in Equation (7).

$$Gini(p)=\sum_{k=1}^{K}p_k(1-p_k)=1-\sum_{k=1}^{K}p_k^2 \tag{7}$$

At the field of text classification, suppose a training set *T* contains *D* texts, these texts belong to *K* categories, and the $k^{\text{th}}$ category contains $C_k$ texts. The training set's Gini index is shown in Equation (8).

$$Gini(T)=1-\sum_{k=1}^{K}(\frac{C_k}{D})^2 \tag{8}$$

The training set *T* can be split into two parts, $T_1$ and $T_2$, given the attribute *A* equals *a*, which is shown in Equation (9).

$$T_1=\{(x,y)\in T\mid A(x)=a\},\quad T_2=T-T_1 \tag{9}$$

Then, the training set's Gini index can be expressed as Equation (10).

$$Gini(T,A)=\frac{D_1}{D}Gini(T_1)+\frac{D_2}{D}Gini(T_2) \tag{10}$$

Where $D_1$ is the number of documents that $T_1$ contains and $D_2$ is the number of documents that $T_2$ contains.

As an analogy, if the training set *T* is split into *n* parts under the condition that the attribute *A* equals some values, the training set's Gini index can be expressed as Equation (11).

$$Gini(T,A)=\sum_{i=1}^{n}\frac{D_i}{D}Gini(T_i) \tag{11}$$

Where $D_i$ is the number of documents that $T_i$ contains and *D* is the total number of documents in the training set.

Equation (11) is the impure expression of the Gini index, which means that the smaller the Gini index, the better the

attribute. Therefore, the training set will split according to the best attribute.

*2.2.2. The Improved Gini Index Algorithm*

Suppose the training set contains *m* categories, and category $c_i$ $(1 \le i \le m)$ can be split into *k* subsets according to different features in it. The Gini index of $c_i$ is shown in Equation (12).

$$Gini(c_i) = 1 - \sum_{k=1}^{K} \frac{|c_{ik}|}{|c_i|} \tag{12}$$

Where $|c_{ik}|$ is the number of $k^{th}$ subset in $c_i$ and $|c_i|$ is the total number features of $c_i$.

Equation (12) is also the impure expression. However, in many cases, especially text classification, we prefer to use the pure expression of Gini index, which is shown in Equation (13).

$$Gini(c_i) = \sum_{k=1}^{K} \frac{|c_{ik}|}{|c_i|} \tag{13}$$

Depending on whether feature *w* appears in $c_i$, $c_i$ can be split into two subsets $t_1$ and $t_2$, i.e., including *w* or not including *w*.

$$Gini(c_i, w) = \frac{|t_1|}{|c_i|} Gini(t_1) + \frac{|t_2|}{|c_i|} Gini(t_2) \tag{14}$$

We always focus on the features that appear in the subset, because the appearing features contribute more information to the Gini index. Equation (14) can be simplified to Equation (15).

$$Gini(c_i, w) = \frac{|t_1|}{|c_i|} Gini(t_1) = P(w|c_i) Ginit(t_1) \tag{15}$$

Where $P(w|c_i)$ is the conditional probability of feature *w* in $c_i$.

Equation (15) is the pure expression of the Gini index; the greater the Gini index, the higher the importance of *w* to $c_i$. The value of the improved Gini index reflects the importance of feature *w* to category $c_i$.

*2.2.3. The Improved TF-Gini Algorithm*

As for the shortcomings of the traditional TF-IDF algorithm, this paper improves the algorithm from two aspects: TF and IDF. The traditional TF algorithm is the frequency of feature in one document. It is obviously suitable for information theory. However, in text classification, our main focus is on the contribution of features to categories. Thus, the improvement of the traditional TF algorithm is shown in Equation (16).

$$TF(w, c_i) = \frac{\sum_{c_i} tf(w)}{1 + \sum_{\bar{c}_i} tf(w)} \tag{16}$$

Where $\sum_{c_i} tf(w)$ is the feature frequency of *w* in $c_i$ and $\sum_{\bar{c}_i} tf(w)$ is the feature frequency of *w* outside $c_i$. Equation (16) reflects the intensive ubiquitous of *w* in $c_i$. If TF is bigger, there are more times of *w* appearing in $c_i$. This reflects the importance of *w* to $c_i$. Therefore, this improved algorithm is more suitable for text classification than the traditional TF algorithm.

In this paper, we use the improved Gini index to replace the IDF part. By analysing the improve Gini index algorithm, it reflects the importance of feature $w$ to $c_i$. The algorithm can overcome the shortcomings of the traditional IDF algorithm. Hence, this paper constructs a new feature weighting algorithm, called the TF-Gini feature weighting algorithm, which is shown in Equation (17).

$$TF-Gini(w,c_i)=\frac{\sum_{c_i} tf(w)}{1+\sum_{\overline{c_i}} tf(w)}\times P(w\mid c_i)Ginit(t_1) \tag{17}$$

*2.3. The Influence of Website's Structure*

A webpage is a kind of special text that contains a specific structure, i.e., HTML label [19]. The location of the key words also represents their importance. For example, the key word appearing in the title has more importance than that in the main body. The locations are marked by the HTML label, such as the *<html> </html>*, *<body> </body>*, and so on. Key words surrounded by different HTML labels should be endowed different weights. The weights assigned in this paper are shown in Table 1.

Table 1 The key words weights surrounded in HTML label

| HTML Label | Weights | HTML Label | Weights |
|---|---|---|---|
| <title> | 5 | <b> | 3 |
| <h1> | 4 | <i> | 2 |
| <h2> | 3 | <u> | 2 |
| <h3> | 2 | <sup> | 1 |
| <h4> ~ <h6> | 2 | <sub> | 1 |
| <font size = 7> | 4 | <big> | 2 |
| <font size = 6> | 3 | <small> | 1 |
| <font size = 5> | 2 | <font size = 3> ~ <font size = 1> | 0.5 |
| <font size = 4> | 1 | | |

Therefore, the cosine similarity should be improved according to the key words' weights, which is shown in Equation (18).

$$\cos(T_1,T_2)=\frac{\sum_{i=1}^{n} w'_{1i}\times w'_{2i}}{\sqrt{\sum_{i=1}^{n}(w'_{1i})^2}\sqrt{\sum_{i=1}^{n}(w'_{2i})^2}} \tag{18}$$

Where $w'=w\times\mu$, $w$ is the original weight that can be calculated by the TF-Gini algorithm, and $\mu$ is the key words' location weights that are shown in Table 1.

*2.4. The Focused Web Crawler based on the Hybrid Similarity*

The process of the improved FWC is shown in Figure 1. The steps of the improved FWC are as flows.

**Step 1** Obtain the hyperlink's anchor text, and calculate the anchor text similarity. If the similarity is bigger than the threshold, download the target webpage.

**Step 2** Sometimes, the similarity cannot be estimated through the anchor text. The FWC need the whole content of the target webpage to calculate the similarity. In this step, the cosine similarity algorithm is a frequently-used method, and TF-Gini is an effective feature weighting algorithm. In addition, the key words' location also has more influence on the cosine similarity algorithm.

According to the experimental results, more than 90% of target webpages can be recognized by anchor text similarity.

This will greatly improve the efficiency of the improved FWC.

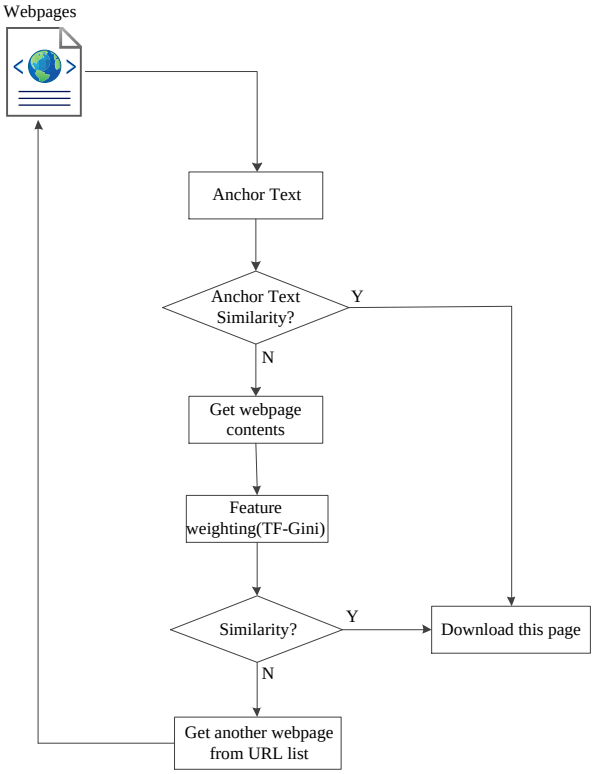


Figure 1. The process of improved FWC

## 3. Experiments and Analysis

### *3.1. Evaluation Criteria for Algorithms*

Precision and Recall are the most popular methods to evaluate an algorithms' performance [23]. Confusion matrix [24], also known as an error matrix, is a specific table layout that allows for visualization of the performance of an algorithm. Each row of the matrix represents the instances in a predicted category, and each column represents the instances in an actual category. Table 2 shows the confusion matrix.

Table 2. Confusion matrix

| | | Actual category | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted category | Positive | True Positives (TP) | False Positives (FP) |
| | Negative | False Negatives (FN) | True Negatives (TN) |

True positive (*TP*) is the proportion of positive instances that were correctly identified. False positive (*FP*) is the proportion of negatives instances that were incorrectly classified as positive. False negative (*FN*) is the proportion of positives instances that were incorrectly classified as negative. True negative (*TN*) is the proportion of negative instances that were classified correctly. Hence, the precision and recall are defined in Equations (19) and (20).

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{19}$$

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{20}$$

High precision indicates that the instances' category as positive is indeed positive, i.e., small number of FP. High recall indicates that more instances' categories are correctly recognized, i.e., small number of *FN*. High recall and low precision

means that most of the positive instances are correctly recognized, but there are many false positives. Low recall and high precision means that many positive instances are missed, i.e., high *FN*, but those predicted as positive are indeed positive. Precision and recall are contradictory. The *F*1-measure represents both of them. It is a balanced evaluation criterion that uses harmonic mean in place of arithmetic Mean, as it punishes the extreme more. The definition of F1-measure is shown in Equation (21).

$$F1-measure=\frac{2\times Precision\times Recall}{Precision+Recall} \tag{21}$$

Confusion matrix is suitable for binary classification. As for multi-classification, we user micro-precision (MicroP), macro-precision (McaroP), micro-recall (MicroR), macro-recall (MacroR), Micro-F1 measure (MicroF1), and Macro-F1 measure (MacroF1) to evaluate the algorithms' performance. They are shown as follows:

$$MacroP=\frac{\sum_{i=1}^{m}Precision_i}{m} \tag{22}$$

$$MicroP=\frac{\sum_{i=1}^{m}TP_i}{\sum_{i=1}^{m}TP_i+\sum_{i=1}^{m}FP_i} \tag{23}$$

$$MacroR=\frac{\sum_{i=1}^{m}Recall_i}{m} \tag{24}$$

$$MicroR=\frac{\sum_{i=1}^{m}TP_i}{\sum_{i=1}^{m}TP_i+\sum_{i=1}^{m}FN_i} \tag{25}$$

$$MacroF1=\frac{2\times MacroP\times MacroR}{MacroP+MacroR} \tag{26}$$

$$MicroF1=\frac{2\times MicroP\times MicroR}{MicroP+MicroR} \tag{27}$$

Where *m* is the total number of samples in the training set.

### *3.2. Evaluation Criteria for Focused Web Crawler*

The performance of FWC is also measured by precision and recall. The computation method is defined in Equations (28) and (29).

$$Recall=\frac{webpages\ that\ downloaded\ by\ FCW}{total\ number\ webpages\ in\ websites}\times 100\% \tag{28}$$

$$Precision=\frac{webpages\ that\ relate\ to\ the\ topics}{the\ total\ number\ of\ downloaded\ webpages}\times 100\% \tag{29}$$

In fact, it is difficult to trace all webpages in the web, that is to say, it is almost impossible to calculate the denominator of recall. Therefore, precision is the only indicator to evaluate the FWC. The higher the precision, the better the performance.

### 3.3. Experimental Results and Analysis

In this paper, we mainly test two experiments. One is to evaluate the performance of the TF-Gini algorithm, and the other is to check the performance of the improved FWC algorithm proposed in this paper.

#### 3.3.1. Feature Weighting Experimental Results

In this experiment, the training set is downloaded from the Internet, and all data is split into three parts. Two parts of the data are randomly selected as the training set, and the remaining part is selected as the test set. After repeating this experiment three times, the mean value of the three times is taken as the final result. We use the TF-IDF algorithm as the classical algorithm and the TF-Gini algorithm as the improved algorithm. The experimental results are shown in Figure 2.

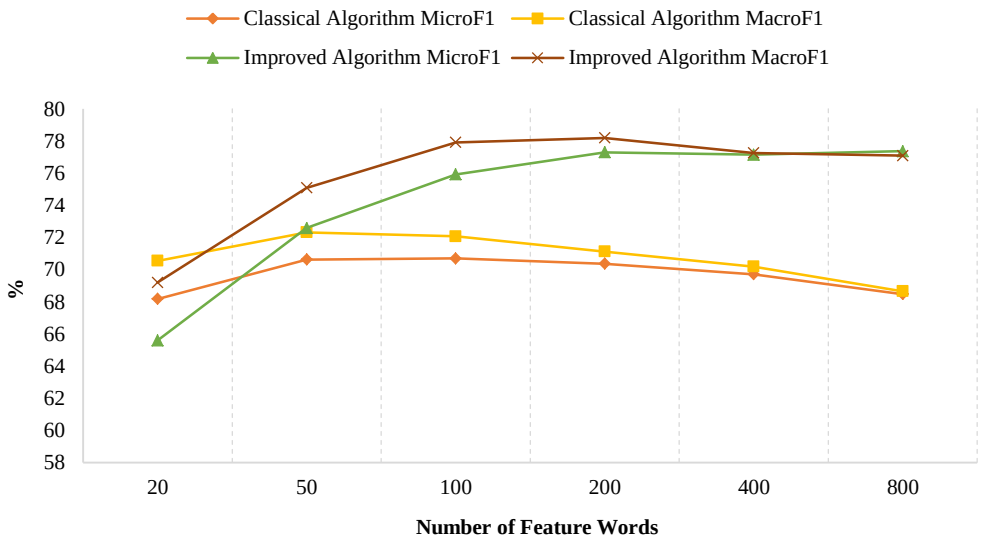


Figure 2. Experimental results of feature weighting

The purpose of feature weighting is to select the most important words for text classification, which can sharply decrease the dimension of feature space. A better feature weighting algorithm greatly improves the performance of classification algorithms. The improved feature weighting algorithm proposed in this paper, named TF-Gini, is better than the traditional TF-IDF algorithm. From Figure 2, as the number of feature words increases, the Micro F1 and Macro F1 of TF-Gini is 77%, while the performance of TF-IDF is 68%.

When the number of feature words is small (less than 50), the performances of TF-Gini and TF-IDF are similar. This means that both algorithms can select the most important features. However, as the number of features increases, the improved algorithm's performance is better than that of the traditional algorithm. This is because the traditional algorithm is not suitable for text classification, as it imports more interference features that reduce the algorithm's precision.

The experimental results proved that TF-Gini is an appropriate feature weighting algorithm for text classification, and it endows feature words appropriate weights according to its importance.

#### 3.3.2. Webpage Crawling Experimental Results

In this experiment, we test the performance of hybrid similarity algorithms in FWC. We use three similarity computing models in this experiment, i.e., the traditional algorithm, anchor text similarity algorithm, and hybrid similarity. Five themes are selected for FWC algorithm testing. We choose 150 feature words for describing each theme. Three news websites are chosen for FWC to download the target webpages, such as *Sohu news*, *Sina news*, and *Tecent news*.

In these experiments, we use three different similarity computing algorithms for FWC, i.e., classical similarity, anchor text similarity, and hybrid similarity proposed in this paper. Three news websites are selected for the FWC crawling test, and the process of FWC is shown in Figure 1.

From the experimental results, shown in Figures 3-5, it can be seen that:

1) The performance of the hybrid similarity algorithm proposed in this paper is the best, because the improved algorithm improves the accuracy of FWC.

2) When the FWC is working, most of the webpages' topics can be recognized by anchor text similarity. This can sharply increase the performance of FWC.

3) Some webpages could not be recognized by anchor text similarity. However, these webpages can be recognized by the hybrid similarity algorithm proposed in this paper. This can increase the accuracy of the FWC.

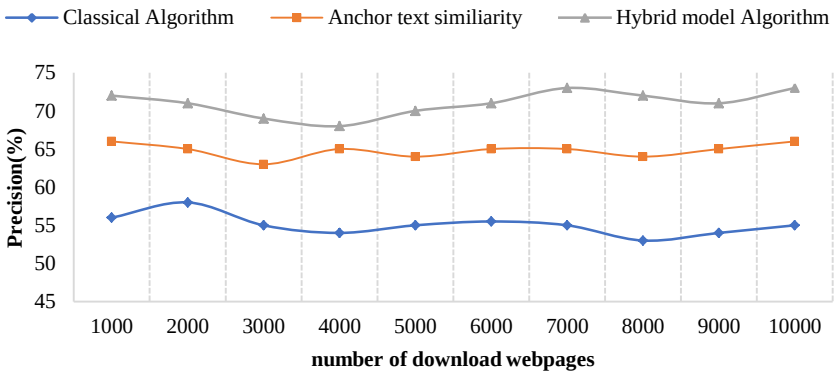


Figure 3. The precision of FWC on website 1

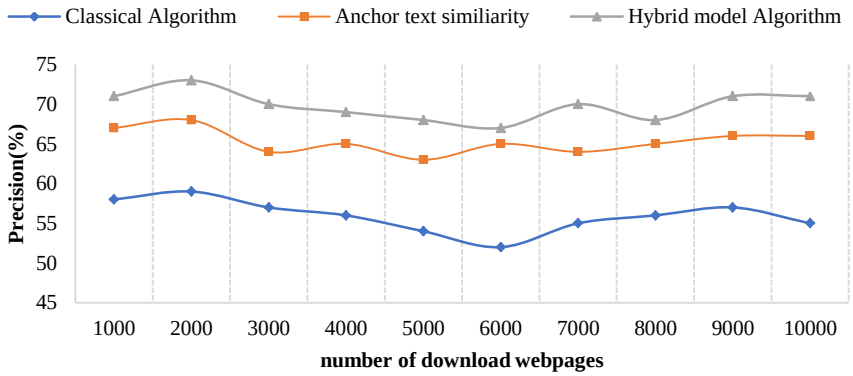


Figure 4. The precision of FWC on website 2

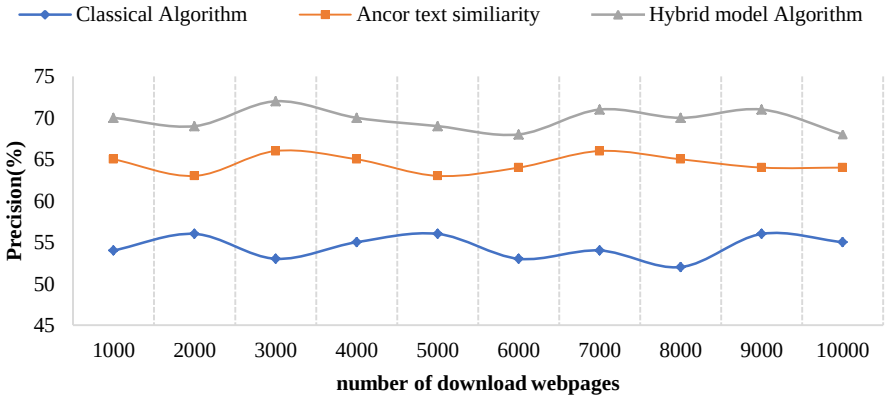


Figure 5. The precision of FWC on website 3

Therefore, the experiment results proved the effectiveness of the hybrid similarity. The hybrid similarity computation method contains anchor text, webpage contents, and webpage structure. Its performance is the best, with an increase of nearly 10% compared with the traditional algorithm.

## 4. Conclusions

With the development of the Internet, FWC is becoming the most convenient method to obtain online information and is widely used in search engines, personalization recommendations, and so on. Under this circumstance, this paper proposed an improved FWC based on hybrid similarity. In order to compute the similarity between the target webpages and topics, the improved algorithm mixed together the anchor text similarity, text similarity, and webpage structure. This paper improved the semantic similarity in text similarity, TF-Gini index feature weighting in text similarity, and cosine similarity by considering the webpage's structure. The experimental results have proven that the improved FWC algorithm's performance is better than that of the traditional web crawler algorithm, and the performance increased by nearly 10%.

## Acknowledgements

This work is supported by the Research Fund for the Doctoral Program of Zhengzhou University of Light Industry (No. 2017BSJJ046 and 2018BSJJ039), Second Education Fund for Industry and Education Project "Digital Science and Technology, Wisdom for the Future" (No. 2018A01094), and Henan Province Educational Committee (No. 17A520064).

## References

1. CNNIC, "The 43th China Internet Network Development Report," (http://www.cnnic.net.cn/hlwfzyj/hlwxzbg/hlwtjbg/, accessed 2019)
2. J. Yue and Q. Liu. "Survey on Topic-Focused Crawlers," *Computer Engineer and Science*, Vol. 37, No. 2, pp. 231-237, 2015
3. H. Dong and F. K. Hussain, "Self-Adaptive Semantic Focused Crawler for Mining Services Information Discovery," *IEEE Transactions on Industrial Informatics*, Vol. 10, No. 2, pp. 1616-1626, 2014
4. T. J. Kim and H. J. Kim, "Machine Learning-based Topical Web Crawler: An Ensemble Approach Incorporating Me-ta-Features," *Journal of Engineering and Applied Sciences*, Vol. 12, No. 8, pp. 4651-4656, 2017
5. S. Batsakis, E. Petrakis, and E. Milios, "Improving the Performance of Focused Web Crawler," *Data and Knowledge Engineering*, Vol. 68, No. 10, pp. 1001-1013, 2009
6. D. Bra and R. Post, "Information Retrieval in the World-Wide Web: Making Client-based Searching Feasible," *Computer Networks and Isdn System*, Vol. 27, No. 2, pp. 183-192, 1994
7. M. Hersovici, M. Jacovi, Y. Maarek, D. Pelleg, and M. Shtalhaim, "The Shark-Search Algorithm. An Application: Tailored Web Site Mapping," in *Proceeding of the International Conference on World Wide Web*, Vol. 30, No. 1-7, pp. 317-326, 1998
8. L. Zhang, Y. Qi, and H. Jiang, "Application of Improved Shark-Search Algorithm in Web Crawler," *Computer Technology and Development*, Vol. 27, No. 8, pp. 192-194+199, 2017
9. B. Wang, J. Wang, X. Sun, and N. Wang, "An Improved Shark-Search Algorithm for Agriculture Web Search Engine," *Chemical Engineering Transactions*, Vol. 51, pp. 709-804, 2016
10. Q. Liu, B. Xiang, N. Yuan, E. Chen, H. Xiong, Y. Zheng, et al., "An Influence Propagation View of PageRank," *ACM Transactions on Knowledge Discovery from Data*, Vol. 11, No. 3, pp. 1-30, 2017
11. Y. Zhou, X. Chen, and W. Wen, "Research on Focused Crawler based on Bayes Classifier," *Application Research of Computers*, Vol. 26, No. 9, pp. 3415-3420+3439, 2009
12. I. S. Altingovde and O. Ulusoy, "Exploiting Interclass Rules for Focused Crawling," *IEEE Intelligent System*, Vol. 29, No. 6, pp. 66-73, 2004
13. G. Pant and P. Srinivasan, "Link Context in Classifier-Guided Topical Crawlers," *IEEE Transaction on Knowledge and Data Engineering*, Vol. 18, No. 1, pp. 107-122, 2006
14. J. Li and K. Yamaguchi, "Focused Crawling by Exploiting Anchor Text using Decision Tree," in *Proceeding of the 14th International World Wide Web*, pp. 1190-1191, 2005
15. G. Pant and P. Srinivasan, "Learning to Crawl: Comparing Classification Schemes," *ACM Transactions on International Systems*, Vol. 23, No. 4, pp. 430-462, 2005
16. Q. Zhang and D. Haglin, "Semantic Similarity Between Ontologies at Different Scales," *IEEE/CAA Journal of Automatic SINICA*, Vol. 3, No. 2, pp. 132-140, 2016
17. H. Yan, F. Li, and Y. Zhou, "Calculation of Sentence Similarity based on HowNet," *Computer Technology and Development*, Vol. 25, No. 11, pp. 53-57, 2015
18. P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," in *Proceeding of the 14th International Joint Conference on Artificial Intelligence*, pp. 39-41, 1995
19. S. Sohangir and D. Wang, "Improved Sqrt-Cosine Similarity Measurement," *Journal of Big Data*, Vol. 4, No. 25, pp. 1-13, 2017
20. S. A. Fawaz and A. Dia, "Toward an Enhanced Arabic Text Classification using Cosine Similarity and Latent Semantic," *Journal of King Saud University*: *Computer and in-Formation Sciences*, Vol. 29, No. 2, pp. 189-195, 2017
21. K. Chen, Z. Zhang, J. Long, and H. Zhang, "Turning from TF-IDF to TF-IGM for Term weighting in Text Classification,"

*Expert Systems with Applications*, Vol. 66, pp. 245-260, 2016
22. L. Rutkowski, L. Pietruczuk, P. Duda, and M. Jaworski, "Decision Trees for Mining Data Streams based on the McDiarmid's Bound," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 6, pp. 1272-1279, 2013
23. C. J. Rijsbergen, "Information Retrieval," Butterworths, London, 1979
24. M. Ohasaki, P. Wang, K. Matsuda, S. Katagiri, H Watanabe, and A. Ralescu, "Confusion-Matrix based Kernel Logistic Regression for Imbalanced Data Classification," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 29, No. 9, pp. 1806-1819, 2017