BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **21**, No 2
Sofia • 2021 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2021-0022

An Enhanced Semantic Focused Web Crawler Based on Hybrid String Matching Algorithm

K. S. Sakunthala Prabha[1], *C. Mahesh*[1], *S. P. Raja*[2]

[1]*Department of Information Technology, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Avadi, Chennai, Tamil Nadu, India*
[2] *School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India*
E-mails: prabha1414@gmail.com chimahesh@gmail.com avemariaraja@gmail.com

***Abstract*:** *Topic precise crawler is a special purpose web crawler, which downloads appropriate web pages analogous to a particular topic by measuring cosine similarity or semantic similarity score. The cosine based similarity measure displays inaccurate relevance score, if topic term does not directly occur in the web page. The semantic-based similarity measure provides the precise relevance score, even if the synonyms of the given topic occur in the web page. The unavailability of the topic in the ontology produces inaccurate relevance score by the semantic focused crawlers. This paper overcomes these glitches with a hybrid string-matching algorithm by combining the semantic similarity-based measure with the probabilistic similarity-based measure. The experimental results revealed that this algorithm increased the efficiency of the focused web crawlers and achieved better Harvest Rate* (*HR*), *Precision* (*P*) *and Irrelevance Ratio* (*IR*) *than the existing web focused crawlers achieve.*

***Keywords*:** *Probabilistic model, hybrid semantic similarity, web focused crawler, string matching.*

1. Introduction

Preceding survey reveals around 20 million web pages being scrolled over for various purposes while currently it shows a rapid increase of more than 1.7 billion web pages [1]. Due to the rampant growth of web pages, the search engines seemed to work laboriously to index the web pages. The web crawler, which is supposed to be the key unit of the search engine is a programmed bot and fetches web pages starting from pre-defined Uniform Resource Locators (URLs). The classic web crawler fetches substantial amount of web pages for a search inclusive of irrelevant web pages and it also demands enormous storage capacity along with increased download time. This disadvantage gives rise to topic-driven crawler, which downloads only the applicable web pages from the web for the given theme.

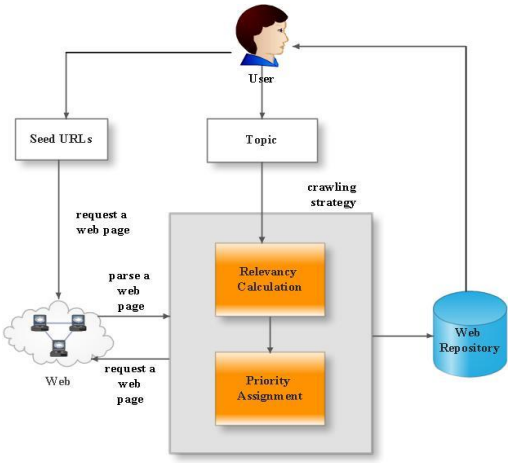

Fig. 1. Workflow diagram of focused web crawler

As shown in Fig. 1, the initial URLs are assigned by the user for an identified topic to the focused crawler. The focused web crawler begins visiting web pages from the pre-identified URLs and evaluates the similarity score of the unvisited web pages. Based on the similarity score, priority is allotted and then cached in the web page repository.

This paper is organized as follows: the existing methodologies are surveyed in Section 2. The newly constructed crawler framework is concentrated in Section 3.5, the various performance evaluation metrics are conferred in Section 4 and the experimental analysis is explained in Section 5. Finally, the outcomes and the future study are conferred in Section 6.

2. Related work

Most of the focused web crawlers, [2-4] have used only full page text to estimate the similarity score of the web page, [5-8] used both full page text and the anchor text for estimating the relevance score. The authors [9-11] used cosine similarity metric to estimate the similarity score of the unvisited web pages. The cosine similarity value can be assessed by finding the Term Frequency and Inverse Document Frequency (TF-IDF). If the web page contains the common phrase of the given theme, then the similarity value will be assessed or else it will be set to zero. It omits the semantically relevant web pages. These flaws are overcome by Hliaoutakis et al. [2] and they proposed Semantic Similarity Retrieval Model (SSRM) which calculates relevance score of the web page by combining term frequency and semantic similarity. Du et al. [12] proposed Semantic Similarity Vector Space Model (SSVSM) which calculates relevance score by combining cosine based similarity metric and semantic similarity based metric.

Dong and Hussain [13] proposed a self-adaptive crawler by integrating semantic similarity score ($f_{\mathrm{ic}}(t,p)$) with statistics similarity score ($f_{\mathrm{stsm}}(t,p)$) to compute the relevance score of the web page by using full page text and the anchor text. The relevance score ($f_{\mathrm{rs}}(\mathrm{url})$) of this crawler [13] is computed by the next equation:

(1) $$f_{\mathrm{rs}}(\mathrm{url}) = \max\left(f_{\mathrm{ic}}(t,p), f_{\mathrm{stsm}}(t,p)\right).$$

Liu and Du [9] designed a Cell-like Membrane Computing Optimization (CMCFC) algorithm for focused web crawler. The cosine similarity of full page text ($f_{\mathrm{rs}}(t,p)$), anchor text ($f_{\mathrm{rs}}(t,a)$), title text ($f_{\mathrm{rs}}(t,\mathrm{title})$) and the surrounding paragraph text ($f_{\mathrm{rs}}(t,\mathrm{st})$) with respect to the topic is computed to find the relevance score of the web page. The relevance score ($f_{rs}(\mathrm{url})$) of CMCFC crawler [9] is computed by the next equation:

(2) $$f_{\mathrm{rs}}(\mathrm{url}) = f_{\mathrm{rs}}(t,p) + f_{\mathrm{rs}}(t,a) + f_{\mathrm{rs}}(t,\mathrm{title}) + f_{\mathrm{rs}}(t,\mathrm{st}).$$

Zheng, Kang and Kim [14] designed a semantic focused crawler using Artificial Neural Network (ANN). The synonyms of the given topic term is computed from the ontology. Then the term frequency of these synonyms are calculated from the web page and given as an input to the ANN to predict the topical relevance of the web page. The major drawback here is that it may not work as required in uncontrolled web environment.

These surveyed focused crawlers downloaded only relevant web pages with certain shortcomings as:

1) Most of them consider the features like the full-page text and anchor text related to the specified topic only, for calculating relevance score. Hence, these focused crawlers cannot obtain very precise hyperlink priorities.

2) The weighted values for Vector Space Model (VSM) based crawlers are calculated based on personal experience, which produces inaccurate results.

3) SSRM combinea the term frequency with semantic similarity and SSVSM combines the cosine similarity with semantic similarity to find the relevance score based on ontology. Since only the path length based semantic similarity is considered these crawlers incorrectly gather the relevance score for some topic and web page pair.

This paper proposes a new hybrid string-matching algorithm to overcome the challenges studied by combining semantic-based similarity with probabilistic-based similarity to determine the relevance score between the topic and the web page based on the following assumptions.

The similarity between the topic and the web page when assessed by

1. Path length (l) is the similarity is high when the path length between them is less.
2. Depth (d) is the similarity is high when the depth between two terms is less.
3. Commonality is the more commonality they have, the more similar they are.
4. Difference is the less difference they have, the more similar they are.
5. Inverse document frequency, term frequency and document length normalization.

The contributions of the proposed work are given as follows:

1. The proposed work combines the semantic similarity model with probabilistic model to find the relevance score of the web page.

2. Five focused crawlers Breadth First Search, Vector Space Model (VSM), SSRM, SSVSM and our proposed hybrid crawlers are implemented and evaluated.

3. Methodologies

3.1. Breadth First Search (BFS) Crawler

BFS crawler [15] starts fetching web pages from seed URLs based on the Breadth-First-Search Algorithm. The crawler ranks the web page once it has reached the maximum depth it intended to visit, or if there are no further pages to be visited.

The page rank can be formulated by using the next equation:

$$f_{\mathrm{pr}}(\mathrm{page}) = \propto + (1 - a) \sum_{i=1}^{n1} f_{\mathrm{io}}(\mathrm{page}_i) \tag{3}$$

where $f_{\mathrm{pr}}(\mathrm{page})$ is the page rank function, $\propto$ is the ratio of tuning parameter and the total web pages, a is the tuning parameter, $f_{\mathrm{io}}(\mathrm{page}_i)$ is the function of in-link out-link ratio of page_i. Fig. 2 shows the working of the BFS crawler.

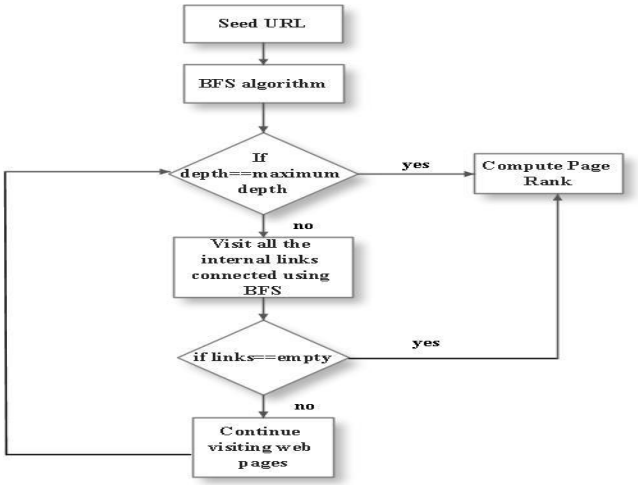

Fig. 2. Breadth-First-Search crawler

3.2. Vector Space Model (VSM) crawler

In the VSM crawler [16] the pre-defined URLs are assigned by the user for a given theme. The VSM web crawler starts visiting web pages from the pre-defined URLs and finds the similarity score of unexplored web pages by using cosine similarity metric. The VSM crawler computes the similarity between the topic and the full-page text and also between the topic and the anchor text. Then the results of the two similarity measures are integrated to calculate the relevance score. On the Basis of the relevance score priority is assigned and stored in a web page repository. The weight values are calculated by using TF-IDF. The cosine similarity of the VSM crawler can be formulated by the next equation:

$$f_{\mathrm{rs}}(\mathrm{url}) = \frac{w_{\mathrm{tp}} * f_{\mathrm{cs}}(t,\ p) + w_{\mathrm{ta}} * f_{\mathrm{cs}}(t,\ a)}{2}, \tag{4}$$

where $f_{\mathrm{rs}}(\mathrm{url})$ is the relevance score function of the web page, $f_{\mathrm{cs}}(t, p)$ is the cosine similarity function of the topic and the page text, $f_{\mathrm{cs}}(t, a)$ is the cosine similarity function of the topic and the anchor text, w_{tp} is the weight value between the topic

and page text, w_{ta} is the weight value between the topic and the anchor text. Fig. 3 shows the working of the VSM crawler.

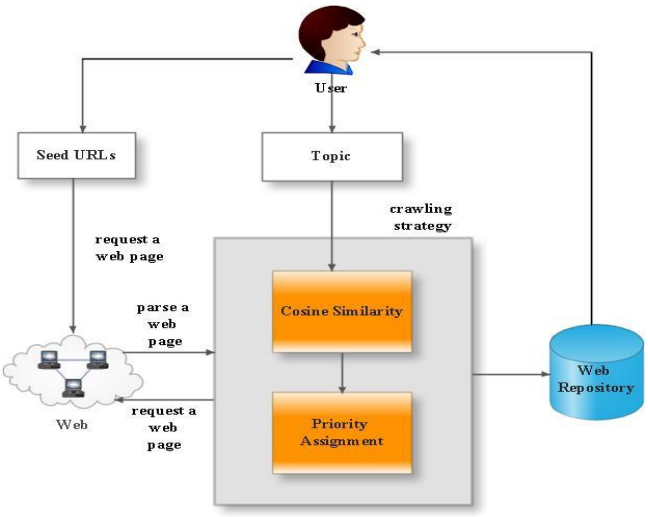

Fig. 3. Vector space model crawler

3.3. Semantic Similarity Retrieval Model (SSRM) crawler

H l i a o u t a k i s et al. [2] have proposed a semantic focused crawler called SSRM by integrating TF-IDF and the path length based semantic similarity algorithm. The relevance score has been calculated for two features: full page text and the anchor text for the given topic. The priority score of the un-visited web pages are computed by combining relevance score of two features. The relevance score ($f_{\mathrm{rs}}(\mathrm{url})$) of SSRM can be computed by the next equation

$$f_{\mathrm{rs}}(\mathrm{url}) = \left(f_{\mathrm{fp}}(t,p) * f_{\mathrm{tfidf}}(t,p)\right) + \left(f_{\mathrm{at}}(t,a) * f_{\mathrm{tfidf}}(t,a)\right), \tag{5}$$

where $f_{\mathrm{fp}}(t,p)$ and $f_{\mathrm{fp}}(t,a)$ are the semantic similarity score of the full page term and the anchor term with repect to the theme, $f_{\mathrm{tfidf}}(t,p)$ and $f_{\mathrm{tfidf}}(t,a)$ are the TF-IDF score of the full page term and the anchor term.

3.4. Semantic Similarity Vector Space Model (SSVSM) crawler

D u Y. et al. [12] proposed a semantic focused crawler by combining the cosine similarity with the path length based semantic similarity. The relevance score is calculated for two features namely full page text and the anchor text for the given topic and then it combines their relevance score to assess the precedence of the unvisited web pages. The relevance score ($f_{\mathrm{rs}}(\mathrm{url})$) of SSVSM can be calculated by the next equation:

$$f_{\mathrm{rs}}(\mathrm{url}) = \left(f_{\mathrm{fp}}(t,p) * f_{\mathrm{c}}(t,p)\right) + \left(f_{\mathrm{at}}(t,a) * f_{\mathrm{c}}(t,a)\right), \tag{6}$$

where $f_{\mathrm{fp}}(t,p)$ and $f_{\mathrm{fp}}(t,a)$ is the semantic similarity score of the full page term and the anchor term with respect to the given theme, $f_{\mathrm{c}}(t,p)$ and $f_{\mathrm{c}}(t,a)$ is the cosine score of the full page term and the anchor term with respect to the given theme.

3.5. Proposed methodology

The proposed framework shown in the Fig. 4 consists of five important components. They are (i) web page fetcher, (ii) policy centre, (iii) crawler repository, (iv) web page parser, and (v) relevance computation. Web page fetcher is used to fetch the URL of unvisited web pages and downloads relevant web pages. Policy centre verifies whether the downloadable web page is based on the predefined strategies or not. The crawler repository stores the downloaded web pages. Web page parser parses the downloaded web pages, Relevance computation is used to compute relevance score between Topic Description (TD_i) terms and Feature Description (FD_j) using hybrid string matching algorithm.

Here the topic is initialized by the user. The seed URLs for the initialized topics are retrieved from the top ten results of Google. Then, the first step is to create TD_i values by finding the root word and synonyms with the help of WordNet 2.1 [17] Ontology-base by using NLTK [18] library function. The next step is to send all top ten seed URLs one by one to the web page fetcher to download. Web page fetcher retrieves all the URLs from the downloaded web pages and sends them to the policy centre. The Policy centre identifies whether the web page is to be fetched with the help of predefined policies and sends the information back to the web page fetcher. Then the web page fetcher begins to download all the web pages, which are accepted for downloading by the policy centre. The crawler repository stores these downloaded web pages. The web page parser receives these stored web pages containing the full web page with HTML tags and then returns the parsed web page in plain text to the crawler repository by removing the HTML tags. Then from the parsed web pages, the features such as title term, heading term, bold text, anchor text and full-page terms of the web page are extracted. FD_j values are extracted by feature processing techniques such as word tokenization, Parts-of-Speech tagging, Non-sense word filtering and stemming on all the features. These can be done by using NLTK library function.

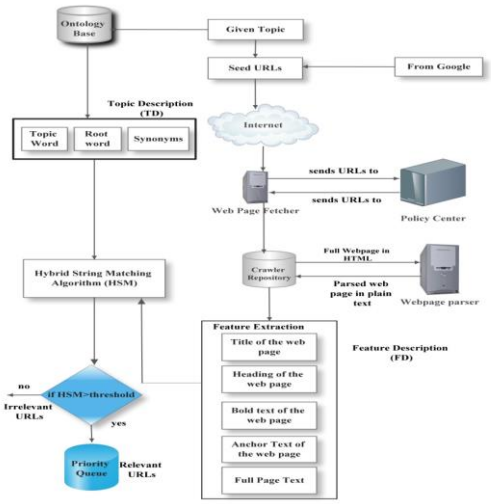

Fig. 4. Proposed focused crawler architecture

The similarity between the TD_i terms and the FD_j terms are calculated by using hybrid string matching algorithm. If similarity value between extracted TD_i terms and FD_j terms are above the threshold then the website shall be deemed to be related to the website. Otherwise, it would be considered as irrelevant web site. Then depending on the relevance score of the web page, priority is assigned for every URL available in it, and then stored in the priority queue.

3.5.1. Proposed Hybrid semantic string matching algorithm

The five assumptionss mentioned in Section 2 are considered to find similarity between TD_i terms and FD_j terms. In certain word pairs of TD_i and FD_j, for instance "furnace" and "stove", similarity measure based on information content generates the relevance score of 0.18 while measure based on edge counting generates the relevance score of 0.58. For some other word pairs of TD_i and FD_j, for instance "coast" and "hill" similarity measure based on information content generates the relevance score of 0.58 while measure based on edge counting generates the relevance score of 0.366. To solve these drawbacks a hybrid string matching algorithm is proposed by combining the edge counting based methods proposed by Li, Bandar and McLean [19], Information content based methods proposed by Lin [20] and the probabilistic based model Best Match 25 (BM25) proposed by Robertson [21].

3.5.1.1. Semantic string matching algorithm

3.5.1.1.1. Path Length and Depth based similarity

This paper satisfies the first two assumptions mentioned in section 2 by using Li, Bandar and McLean [19] path length-depth based semantic similarity measure.

The path length (l) between two words TD_i and FD_j can be determined based on the following assumptions:

1) If TD_i and FD_j are in the same concept the value of l is set to 0.

2) If TD_i and FD_j are not in the same concept but concept of TD_i and FD_j contain one or more common words, the value of l is set to 1.

3) Otherwise, set the value of $l = \text{count of path length}$.

The depth (d) between two words TD_i and FD_j can be obtained by the counting of the subsumer levels to top of syntactic structure. The similarity between the concepts at the upper layers is low and lower layers are high for polysemous words. To avoid this problem the function is set as a monotonically increasing function.

The similarity between TD_i terms and FD_j terms is

$$(7)\quad f_{\mathrm{ld}}(\mathrm{TD}_i, \mathrm{FD}_j) = \max_{C \in l(\mathrm{TD}_i, \mathrm{FD}_j),\ d(\mathrm{TD}_i, \mathrm{FD}_j)} \left(e^{-k_1 l(\mathrm{TD}_i, \mathrm{FD}_j)} \cdot \frac{e^{k_2 d(\mathrm{TD}_i,\ \mathrm{FD}_j)} - e^{-k_2 d(\mathrm{TD}_i,\ \mathrm{FD}_j)}}{e^{k_2 d(\mathrm{TD}_i,\ \mathrm{FD}_j)} + e^{-k_2 d(\mathrm{TD}_i,\ \mathrm{FD}_j)}} \right),$$

where $k_1 \geq 0$ and $k_2 > 0$ are scaling parameters, c is the count of path length, $\delta(\mathrm{FD}_j)$ is concept of FD_j in WordNet, C is a concept in WordNet, $l(\mathrm{TD}_i, \mathrm{FD}_j)$ is length between the two concepts TD_i, FD_j in WordNet, $d(\mathrm{TD}_i, \mathrm{FD}_j)$ is depth value between the two concepts TD_i, FD_j in WordNet.

The path length between TD_i and FD_j is

$$l(\mathrm{TD}_i, \mathrm{FD}_j) = \begin{cases} 0 \text{ if } \mathrm{TD}_i = \mathrm{FD}_j, \\ 1 \text{ if } \mathrm{TD}_i \in \delta(\mathrm{FD}_j), \\ c \quad \text{otherwise}, \end{cases} \tag{8}$$

where c is count of path length, $\delta(\mathrm{FD}_j)$ is concept of FD_j in WordNet.

3.5.1.1.2. Information content based similarity

This paper satisfies the third and fourth assumptions mentioned in Section 2 by Lin [20] similarity measure. The Information Content (IC) between TD_i and FD_j can be derived by the commonality and the difference between two terms. The relevance score function between TD_i and FD_j is depicted in

$$f_{\mathrm{ic}}(\mathrm{TD}_i, \mathrm{FD}_j) = \max_{C \in l(\mathrm{TD}_i,\ \mathrm{FD}_j),\ d(\mathrm{TD}_i,\ \mathrm{FD}_j)} \left(\frac{2 * \mathrm{IC}(\mathrm{LCS}(\mathrm{TD}_i,\ \mathrm{FD}_j))}{\mathrm{IC}(\mathrm{TD}_i) + \mathrm{IC}(\mathrm{FD}_j)} \right). \tag{9}$$

The Information Content of TD_i is formulated by

$$\mathrm{IC}(\mathrm{TD}_i) = -\log(P(\mathrm{TD}_i)). \tag{10}$$

The Information Content of FD_j is formulated by

$$\mathrm{IC}(\mathrm{FD}_j) = -\log\left(P(\mathrm{FD}_j)\right). \tag{11}$$

The Least Common Subsumer (LCS) between TD_i and FD_j is shown in

$$\mathrm{LCS}(\mathrm{TD}_i, \mathrm{FD}_j) = -\log\left(P(\mathrm{TD}_i \vee \mathrm{FD}_j)\right), \tag{12}$$

where $\mathrm{LCS}(\mathrm{TD}_i, \mathrm{FD}_j)$ is Least Common Subsumer of TD_i and FD_j, $\mathrm{IC}(\mathrm{TD}_i)$ is Information Content of TD_i and $\mathrm{IC}(\mathrm{FD}_j)$ is Information Content of FD_j.

Here, the TD_i and FD_j are the two concepts in WordNet. The $P(\mathrm{TD}_i)$ and $P(\mathrm{FD}_j)$ can be formulated using next equations

$$P(\mathrm{TD}_i) = \frac{\text{Total Number of concepts subsumed by } \mathrm{TD}_i}{\text{Total Number of concept in WordNet}}, \tag{13}$$

$$P(\mathrm{FD}_j) = \frac{\text{Total Number of concepts subsumed by } \mathrm{FD}_j}{\text{Total Number of concept in WordNet}}. \tag{14}$$

3.5.1.2. Probabilistic based similarity algorithm

This paper satisfies the fourth assumption mentioned in section 2 by BestMatch25 (BM25) model. BestMatch25 (BM25) is a probabilistic model based on Inverse Document Frequency (IDF), Term Frequency (TF) and Document Length Normalization (DocLen). The BM25 probabilistic function can be computed by

$$f_{\mathrm{bm}}(\mathrm{TD}_i, \mathrm{FD}_j) = \max_{\mathrm{TD}_i,\ \mathrm{FD}_j} \left(\frac{\left(\sum \propto_{\mathrm{TD}_i,\ \mathrm{FD}_j} * \log\left(\frac{N - n_i + 0.5}{n_i + 0.5} \right) \right)}{k_{\max}} \right). \tag{15}$$

The $\propto_{\mathrm{TD}_i, FD_j}$ can be calculated by the next equation

$$\propto_{\mathrm{TD}_i,\ \mathrm{FD}_j} = \frac{(k_1 + 1) * f_{\mathrm{TD}_i,\ \mathrm{FD}_j}}{k_1 \left((1 - b) + b * \frac{\mathrm{len}(\mathrm{FD}_j)}{\mathrm{avg\_doclen}} \right) + f_{\mathrm{TD}_i,\ \mathrm{FD}_j}}, \tag{16}$$

where: $k_{\max}$, k_1 are constant which are set experimentally; N is the total number of document FD_j; n_i is the number of documents that contain TD_i; $f_{\mathrm{TD}_i,\ \mathrm{FD}_j}$ is the term

frequency; $\text{len}(\text{FD}_j)$ is the length of the document FD_j; avg_doclen is average document length.

3.5.1.3. Proposed Hybrid String Matching Algorithm

In order to find maximum similarity between TD_i and FD_j, this paper proposes a new hybrid string algorithm by combining the semantic string matching with the probabilistic string matching to satisfy the assumptions mentioned in Section 2. The hybrid string-matching algorithm can be formulated by using the equation

(17) $$f_{\text{rs}}(\text{TD}_i, \text{FD}_j) = \max\left(f_{\text{ld}}(\text{TD}_i, \text{FD}_j), f_{\text{ic}}(\text{TD}_i, \text{FD}_j), f_{\text{bm}}(\text{TD}_i, \text{FD}_j)\right).$$

The priority for unvisited web pages are calculated by relevance score of full page text, bold text, anchor text, title text and heading text. Then priority of the web page can be calculated by using the next equation

(18) $$f_{\text{p}}(\text{URL}) = \text{average}\left(f_{\text{fp}}(\text{TD}_i, \text{FD}_j), f_{\text{at}}(\text{TD}_i, \text{FD}_j), f_{\text{b}}(\text{TD}_i, \text{FD}_j), f_{\text{t}}(\text{TD}_i, \text{FD}_j), f_{\text{h}}(\text{TD}_i, \text{FD}_j)\right),$$

where $f_{\text{p}}(\text{URL})$ is the priority score of unvisited URL, $f_{\text{fp}}(\text{TD}_i, \text{FD}_j)$ is relevance score of full page text, $f_{\text{at}}(\text{TD}_i, \text{FD}_j)$ is relevance score of anchor text, $f_{\text{b}}(\text{TD}_i, \text{FD}_j)$ is relevance score of bold text, $f_{\text{t}}(\text{TD}_i, \text{FD}_j)$ is relevance score of title text, $f_{\text{h}}(\text{TD}_i, \text{FD}_j)$ is relevance score of heading text.

3.5.1.4. Algorithm of the proposed method

Input: Topic *T*
Output: $f_{\text{rs}}(\text{TD}_i, \text{FD}_j)$
For *i* = 1 to *n*
  Retrieve the Topic terms and store it in TD_i
  For *j* = 1 to *m*
    Retrieve the Feature set values and store it in FD_j
    Then calculate
  $f_{\text{rs}}(\text{TD}_i, \text{FD}_j) = \max\left(f_{\text{ld}}(\text{TD}_i, \text{FD}_j), f_{\text{ic}}(\text{TD}_i, \text{FD}_j), f_{\text{bm}}(\text{TD}_i, \text{FD}_j)\right)$
    If $f_{\text{rs}}(\text{TD}_i, \text{FD}_j)$ > threshold
      Download the web page and store the URLs present it in priority queue
    End If
  End For
End For

4. Experimental approach and analysis

The prototype of BFS, VSM, SSRM, SSVSM and the proposed HSM crawler are developed in Python3 comprised in the platform of Spyder3.6. For SSRM, SSVSM and the proposed HSM-focused crawler, WordNet 2.1 ontology is used to find semantic similarity. Ten sets of topics and their respective seed URLs are provided as input from the famous Google search engine in the experimental configuration as shown in Table 1.

Table 1. Initial URLs for the given topic

S.No	Topic	Seed URL
1	Football	• https://www.bbc.com/sport/football • https://www.theguardian.com/football
2	Knowledge Mapping	• https://www.mindmeister.com/blog/build-knowledge-map/ • https://openknowledgemaps.org/
3	Robot Army	• https://mods.factorio.com/mod/robotarmy • https://robotarmy.com.au/
4	Smart Phone	• https://www.smartphone.nl/ • http://store.smart.com.ph/phones/
5	Cloud Computing	• https://aws.amazon.com/what-is-cloud-computing/ • https://cloudcomputing-news.net/
6	wildfires	• https://www.who.int/health-topics/wildfires • https://www.ready.gov/wildfires
7	Shahrukh khan	• https://starsunfolded.com/shah-rukh-khan/ • https://screenrant.com/shah-rukh-khan-best-movies-imdb/
8	computer	• https://www.expert.nl/computers • https://www.alternate.nl/
9	Apple	• https://www.apple.com/ • https://www.forbes.com/companies/apple/
10	Movie	• https://www.netflix.com/browse/genre/34399 • https://moview.nl/

5. Performance evaluation

The proposed crawler is compared with four other crawlers such as Breadth-First-Search Crawler (BFS), Vector Space Model Crawler (VSM), Semantic Similarity Retrieval Model Crawler (SSRM), Semantic Similarity Vector Space Model Crawler (SSVSM). Ten sets of common topics and each with ten different seed URLs are retrieved from the popular search engine Google, and serves as input for all crawlers. The given topic and their corresponding seed URLs are indicated in Table 1. The experimental results of all the five focused crawlers are demonstrated on the basis of the metrics given in Section 5.1.

5.1. Performance metrics

5.1.1. Harvest rate

Harvest Rate (HR) [22] is the ratio of Number of related web pages retrieved (R_{wp}) by $Total\ number\ of\ web\ pages\ retrieved$ (N_{wp}). The HR can be calculated by the next equation:

(19) $$\mathrm{HR} = \frac{R_{\mathrm{wp}}}{N_{\mathrm{wp}}}.$$

5.1.2. Precision

Precision (P) is the ratio of intersection of relevant web pages downloaded for topic (r_i) and total web pages downloaded for topic (n_i) by total web pages downloaded for topic (n_i). The precision of crawler can be calculated by the next equation

(20) $$P = \frac{r_i \cap n_i}{n_i}.$$

5.1.3. Irrelevance Ratio

Irrelevance Ratio (IR) [23] is ratio of Number of irrelevant web pages retrieved $N_{\text{wp}} - R_{\text{wp}}$ by $Total\ number\ of\ web\ pages\ retrieved\ N_{\text{wp}}$. The Irrelevance Ratio (IR) can be calculated by the equation

(21) $$\text{IR} = \frac{N_{\text{wp}} - R_{\text{wp}}}{N_{\text{wp}}}.$$

5.2. Analysis

The experimental results have been analyzed for all five focused crawlers namely BFS, VSM, SSRM, SSVSM and the proposed HSM crawler. Threshold value for the experiment is set to 0.7. The k_{mean} value is set to 2. The experimental results of all the five focused crawlers are given away in the Table 2 to Table 4. The aim of the analysis phase is to prove that the proposed crawler performs better than the existing focused crawlers, considering metrics as mentioned in Section 5.1.

5.2.1. Performance evaluation using harvest rate

The comparison graph of the BFS, VSM, SSRM, SSVSM and the proposed HSM crawler with respect to the average Harvest Rate (HR) is given in Fig. 5. The BFS crawler downloads all the web pages that it visits without considering the relevance score. Because of this, BFS crawler produced poor results and generated an average HR of 0.184 after 5000 web page crawls. The VSM crawler computes the lexical similarity between the topic and the web page and omits the semantic similarity. This leads to low harvest rate and produces an average HR of 0.21 after 5000 web page crawls. The SSRM crawler multiplies the semantic similarity score with the TF-IDF score to generate the relevance score of the web page. If anyone score among the TF-IDF and the semantic similarity is very less then it leads to poor results. After 5000 web page crawls, SSRM crawler produced an average HR of 0.27. The SSVSM crawler combines the advantage of both the cosine similarity and the semantic similarity. It is computationally costlier to compute the SSVSM score. The SSVSM crawler produced an average HR of 0.312 after 5000 web page crawls. The proposed HSM crawler produced an average HR of 0.389. The result clearly proves that the proposed crawler produced better harvest rate and also downloaded more relevant web pages than the other focused crawlers.

Table 2. Average HR of all topics

Number of Web pages	BFS Crawler	VSM Crawler	SSRM crawler	SSVSM crawler	Hybrid crawler
100	0.310	0.36	0.39	0.431	0.514
200	0.291	0.34	0.37	0.417	0.526
300	0.289	0.34	0.37	0.397	0.567
400	0.276	0.34	0.37	0.426	0.542
500	0.276	0.33	0.37	0.436	0.497
600	0.273	0.33	0.36	0.391	0.483
700	0.273	0.33	0.34	0.372	0.471
800	0.271	0.32	0.34	0.381	0.462
900	0.269	0.32	0.34	0.363	0.431
1000	0.266	0.31	0.33	0.347	0.447
2000	0.203	0.29	0.32	0.337	0.431
3000	0.231	0.27	0.31	0.321	0.414
4000	0.197	0.24	0.29	0.317	0.407
5000	0.184	0.21	0.27	0.312	0.389

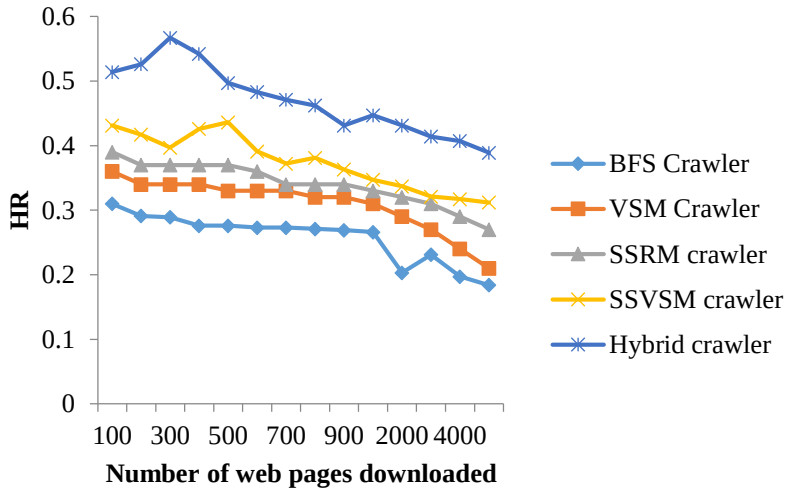

Fig. 5. Result comparison of harvest rate for the five focused crawlers

5.2.2. Performance evaluation by precision

The comparison graph for the BFS crawler, VSM crawler, SSRM crawler, SSVSM crawler and the proposed HSM crawler with respect to the average precision rate is indicated in Fig. 6. After 5000 web page crawls, the BFS crawler, VSM crawler, SSRM crawler, SSVSM crawler and the proposed HSM crawler produced an average precision of 16%, 17%, 23%, 29% and 36%, respectively. The proposed HSM crawler produced an average precision of 36%, which is a better precision rate than the other focused crawlers, and hence proved that the proposed crawler could be able to work well in the dynamic crawling environment.

Table 3. Average precision of all topics

Number of Web pages	BFS Crawler	VSM Crawler	SSRM crawler	SSVSM crawler	Hybrid crawler
100	32	37	32	43	46
200	34	34	34	44	41
300	31	34	36	53	42
400	29	33	37	43	42
500	27	32	37	42	41
600	27	31	34	41	41
700	27	31	32	42	40
800	27	27	31	39	41
900	23	24	31	37	40
1000	20	23	30	36	39
2000	17	24	31	37	40
3000	19	21	33	39	41
4000	16	19	27	33	38
5000	16	17	23	29	36

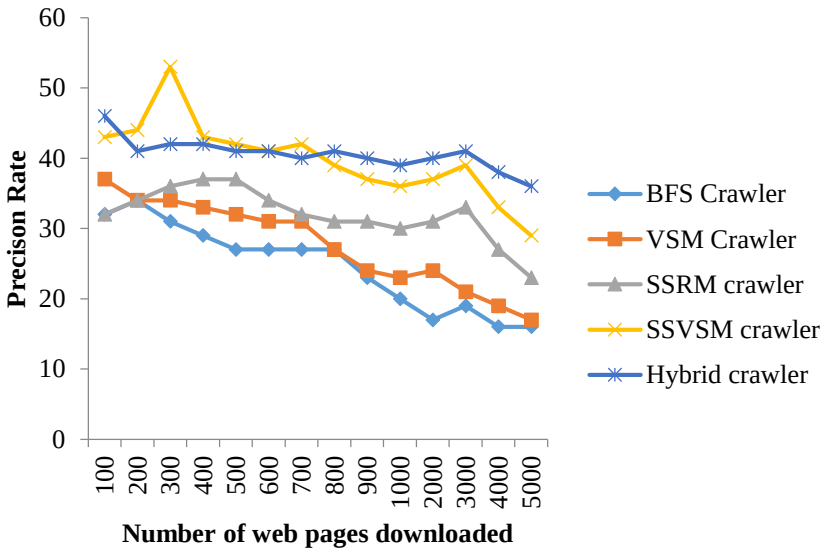

Fig. 6. Result comparison of precision rate for the five focused crawlers

5.2.3. Performance evaluation by irrelevance ratio

The comparison graph for the BFS crawler, VSM crawler, SSRM crawler, SSVSM crawler and the proposed HSM crawler with respect to the average irrelevance ratio is indicated in Fig. 7. After 5000 web page crawls, the BFS crawler, the VSM crawler, the SSRM crawler, the SSVSM crawler, and the proposed HSM crawler produced an average IR of 0.816, 0.79, 0.73, 0.688 and 0.611, respectively. The proposed HSM crawler has an average IR of 0.611, which is a better irrelevance ratio than other focused crawlers, and hence proved that the proposed crawler could be able to filter out more irrelevant web pages.

Table 4. Average irrelevance ratio of all topics

Number of Web pages	BFS Crawler	VSM Crawler	SSRM crawler	SSVSM crawler	Hybrid crawler
100	0.69	0.64	0.61	0.569	0.486
200	0.709	0.66	0.63	0.583	0.474
300	0.711	0.66	0.63	0.603	0.433
400	0.724	0.66	0.63	0.574	0.458
500	0.724	0.67	0.63	0.564	0.503
600	0.727	0.67	0.64	0.609	0.517
700	0.727	0.67	0.66	0.628	0.529
800	0.729	0.68	0.66	0.619	0.538
900	0.731	0.68	0.66	0.637	0.569
1000	0.734	0.69	0.67	0.653	0.553
2000	0.797	0.71	0.68	0.663	0.569
3000	0.769	0.73	0.69	0.679	0.586
4000	0.803	0.76	0.71	0.683	0.593
5000	0.816	0.79	0.73	0.688	0.611

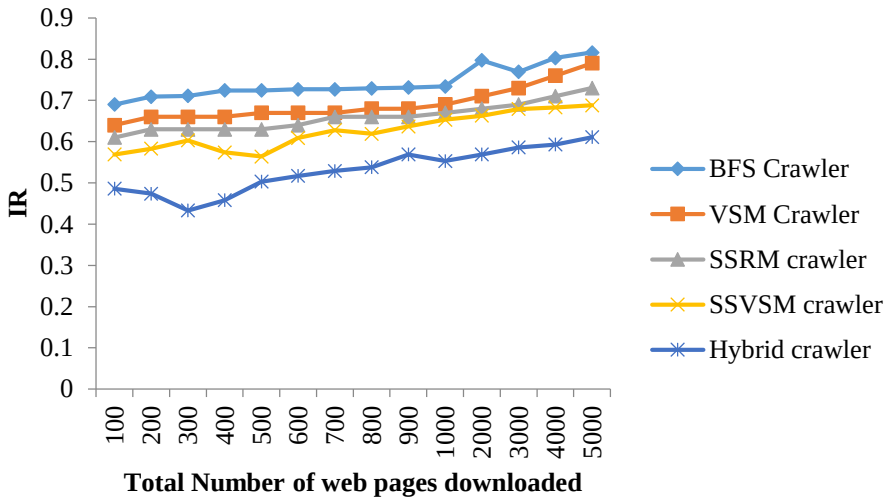

Fig. 7. Result comparison of irrelevance ratio for the five focused crawlers

6. Conclusion

This paper has proposed the Hybrid focused crawler that could enhance the performance of the focused web crawlers. This framework integrates semantic similarity based approach with the probabilistic similarity model and it takes title text, full page text, bold text, anchor text and heading text as its feature documents. This framework first extracts the Topic Description (TD_i) terms by applying stemming and synonym search. Then it extracts the Feature Description (FD_j) terms such as title text, full page text, bold text, anchor text and heading text from the unvisited web pages. Finally it computes the similarity value between TD_i terms and

FD_j terms by Hybrid String Matching Algorithm. Based on relevance score, web pages are prioritized and stored in a priority queue. The experimental results demonstrate that this proposed algorithm outperforms the Breadth-First-Crawler (BFS), Vector Space Model Crawler (VSM), Semantic Similarity Retrieval Model (SSRM), Semantic Similarity Vector Space Model Crawler (SSVSM) in terms of harvest rate, precision rate and irrelevance ratio. The string-matching algorithm with statistics based machine learning techniques can be enhanced in future.

References

1. Internet Live Stats. 2020. **https://www.internetlivestats.com/total-number-of-websites/**
2. Hliaoutakis, A., G. Varelas, E. Voutsakis, E. G. M. Petrakis, E. Milios. Information Retrieval by Semantic Similarity. – Int. J. Semant. Web Inf. Syst., Vol. **2**, 2011, No 3, pp. 55-73.
3. Geng, Z., D. Shang, Q. Zhu, Q. Wu, Y. Han. Research on Improved Focused Crawler and Its Application in Food Safety Public Opinion Analysis. – In: Proc. of 2017 Chinese Autom. Congr., 2017, pp. 2847-2852.
4. Liu, Z., Y. Du, Y. Zhao. Focused Crawler Based on Domain Ontology and FCA. – J. Inf. Comput. Sci., Vol. **8**, 2011, No 10, pp. 1909-1917.
5. Chakrabarti, S., M. van den Berg, B. Dom. Focused Crawling: A New Approach to Top-Specific Web Source Discovery. – Comput. Networks, Vol. **31**, 1999, No 11-16, pp. 1623-1640.
6. Menczer, F. Complementing Search Engines with Online Web Mining Agents. – Decis. Support Syst., Vol. **35**, 2003, No 2, pp. 195-212.
7. Park, J. R., C. Yang, Y. Tosaka, Q. Ping, H. el Mimouni. Developing an Automatic Crawling System for Populating a Digital Repository of Professional Development Resources: A Pilot Study. – J. Electron. Resour. Librariansh., Vol. **28**, 2016, No 2, pp. 63-72.
8. Agre, G. H., N. V. Mahajan. Keyword Focused Web Crawler. – In: Proc. of 2nd Int. Conf. Electron. Commun. Syst. ICECS'15, 2015, pp. 1089-1092.
9. Liu, W. J., Y. J. Du. A Novel Focused Crawler Based on Cell-Like Membrane Computing Optimization Algorithm. – Neurocomputing, Vol. **123**, 2014, pp. 266-280.
10. Farag, M. M. G., S. Lee, E. A. Fox. Focused Crawler for Events. – Int. J. Digit. Libr., Vol. **19**, 2018, No 1, pp. 3-19.
11. Chen, Z., J. Ma, J. Lei, B. Yuan, L. Lian, L. Song. A Cross-Language Focused Crawling Algorithm Based on Multiple Relevance Prediction Strategies. – Comput. Math. with Appl., Vol. **57**, 2009, No 6, pp. 1057-1072.
12. Du, Y., W. Liu, X. Lv, G. Peng. An Improved Focused Crawler Based on Semantic Similarity Vector Space Model. – Appl. Soft Comput. J., Vol. **36**, 2015, pp. 392-407.
13. Dong, H., F. K. Hussain. Self-Adaptive Semantic Focused Crawler for Mining Services Information Discovery. – IEEE Trans. Ind. Informatics, Vol. **10**, 2014, No 2, pp. 1616-1626.
14. Zheng, H. T., B. Y. Kang, H. G. Kim. An Ontology-Based Approach to Learnable Focused Crawling. – Inf. Sci. (Ny)., Vol. **178**, 2008, No 23, pp. 4512-4522.
15. Najork, M., J. L. Wiener. Breadth-First Search Crawling Yields High-Quality Pages. – In: Proc. of 10th Int. Conf. World Wide Web, WWW'01, 2001, pp. 114-118.
16. Salton, G., A. Wong, C. Yang. Information Retrieval and Language Processing: A Vector Space Model for Automatic Indexing. – Commun. ACM, Vol. **18**, 1975, No 11, pp. 613-620.
17. Princeton University. About WordNet. WordNet, Princeton University, 2010.
18. Bird, E. L., E. K. Bird, Steven. Natural Language Processing with Python. O'Reilly Media Inc, 2009.
19. Li, Y., Z. A. Bandar, D. McLean. An Approach for Measuring Semantic Similarity between Words Using Multiple Information Sources. – IEEE Trans. Knowl. Data Eng., Vol. **15**, 2003, No 4, pp. 871-882.

20. L i n, D. Definition of Similarity in Informaiton Theory.Pdf, 1989.
21. R o b e r t s o n, S. The Probabilistic Relevance Framework: BM25 and Beyond. – Foundation and Trend K in Retrievel, Vol. **3**, 2010, No 4.
22. D h a n i t h, P. R. J., B. S u r e n d i r a n. An Ontology Learning Based Approach for Focused Web Crawling Using Combined Normalized Pointwise Mutual Information and Resnik Algorithm. – Int. J. Comput. Appl., Vol. **0**, 2019, No 0, pp. 1-7.
23. D h a n i t h, P. R. J., B. S u r e n d i r a n, S. P. R a j a. A Word Embedding Based Approach for Focused Web Crawling Using the Recurrent Neural Network. – International Journal of Interactive Multimedia and Artificial Intelligence, 2020, pp. 1-11.

Received: 30.12.2020; Second Version: 26.03.2021; Accepted: 14.04.2021