International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2607 - 2614
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0359

Survey on Focused Web Crawling Approaches for Ayurvedic Plant Information Retrieval

Sakshi Powale[1], Asavari Desai[2], Riya Desai[3], Swati Sonavane[4], Dr. Divya Tamma[5]
[1,2,3,4]UG Scholar, Department of AI&DS, Rajiv Gandhi Institute of Tech, Mumbai, Maharashtra, India.
[5]Assistant Professor, Department of AI&DS, Rajiv Gandhi Institute of Tech, Mumbai, Maharashtra, India.
Email ID: *strategic.sakshi@gmail.com[1], asavaridesai325@gmail.com[2], riadesai3110@gmail.com[3], truptisonavane847@gmail.com[4], divya.tamma@mctrgit.ac.in[5]*

Abstract

The Ayurvedic medical system is heavily reliant on medicinal plants, demanding correct information retrieval from the web. This study examines specific web crawling strategies for collecting useful information about Ayurvedic botanicals, with a focus on deep learning methodologies. Crawlers optimized with machine learning models retrieve domain-specific content while filtering out unnecessary data. The paper looks at approaches such as the TRES framework, which is a reinforcement learning-based crawler that discretizes vast state and action spaces in order to effectively choose ideal URLs. Furthermore, convolutional neural networks (CNN) and natural language processing (NLP) have been used in crawlers to improve categorization, as demonstrated by successful Turkish language processing applications. The paper "Learning to Crawl: Comparing Classification Schemes" conducts a comparative comparison of old rule-based approaches and newer deep learning classifiers, demonstrating the latter's superiority. In addition, a Naive Bayes classifier is employed in an Ayurvedic plant-focused crawler, which employs query expansion via a carefully curated thesaurus to improve relevancy in retrieved web pages. This poll emphasizes the need for more efficient, adaptive, and focused crawlers powered by deep learning to progress Ayurvedic research.

Keywords: *Focused Web Crawling, Deep Learning, TRES Framework, Reinforcement Learning, Convolutional Neural Networks (CNN), Natural Language Processing (NLP)*

1. Introduction

The Ayurvedic medical system, which has historical roots in India, promotes a holistic approach to health through the use of therapeutic herbs. Recently, there has been a spike in interest in Ayurvedic treatments, spurred by a global need for natural alternatives, emphasizing the importance of accurate information on these plants. However, the sheer volume of internet content makes it difficult to identify important information. Traditional search engines frequently fall short of domain-specific queries, prompting the development of targeted web crawling approaches that improve information retrieval efficiency and accuracy. Advances in machine learning and deep learning have greatly enhanced web crawling capabilities. Unlike traditional methods, newer systems use techniques like reinforcement learning, as demonstrated by the TRES framework, to optimize URL selection. Content classification has been improved by the combination of natural language processing (NLP) with convolutional artificial neural networks (CNNs). This research explores targeted web crawling algorithms for Ayurvedic plant knowledge, with an emphasis on deep learning applications that enable access to high-quality, domain-relevant information.

2. Literature Survey

Nisha Pawar and Dr. K. Rajeswari's study [1], focuses on creating a targeted web crawler that will recover web pages about medicinal plants and the illnesses they cure. Motivating the research is the abundance of unstructured online material about medicinal plants, especially in relation to the plant-based medicines used in the Indian Ayurvedic system. Conventional search engines, however, frequently have trouble effectively filtering and retrieving pertinent content in this field. To address these issues, the authors suggest a tailored web crawler that produces more concentrated search results by navigating the medicinal plants domain. A manually constructed medicinal plant thesaurus and the Naive Bayes classification algorithm form the basis of the suggested crawler's methodology. The Naive Bayes classification method is used to assess each web page's relevance based on four essential lexical features: title text, meta-description, anchor text, and URL tokens. The system starts with a set of seed URLs and grows by following links. Cosine similarity is used to process these features and rank web pages according to how relevant they are to the user's query. Moreover, query expansion greatly increases the precision and accuracy of the search results by connecting user queries to relevant keywords in the thesaurus, which aids the crawler in conducting more efficient searches. The experimental findings show how successful this system is. The crawler used the Naive Bayes classification algorithm to retrieve relevant web pages with great precision, achieving a classification accuracy of 90%. When compared to other algorithms, such as decision trees and multilayer perceptron's, Naive Bayes performed better. Furthermore, using a manual thesaurus to expand queries increased retrieval accuracy by 20% when compared to general-purpose crawlers. The paper emphasizes the potential for further development, suggesting the incorporation of more advanced machine learning algorithms and the expansion of the thesaurus to cover a broader range of medicinal plants and diseases, laying the groundwork for future research in specialized information retrieval systems. The study [2] describes TRES (Tree Reinforcement Spider), an advanced system for optimizing focused web crawling. Focused crawlers seek out appropriate web pages on a certain domain while avoiding non-relevant ones. The crawling environment is modelled as a Markov Decision Process (MDP) by the authors, who employ Reinforcement Learning (RL) to improve the crawler's decision-making strategies. Tree-Frontier, an innovative decision-tree-based method, is employed in the framework to efficiently manage the enormous state and action spaces associated with focused crawling. Tree-Frontier's discretization of the state-action space allows the RL agent to evaluate fewer actions at each step, considerably lowering computing complexity while maintaining a high harvest rate (the proportion of relevant pages retrieved). The TRES framework incorporates several innovative components, including a keyword expansion strategy and a deep learning classifier, KwBiLSTM, for page relevance evaluation. Starting with a small set of seed keywords, the framework uses word embeddings to expand this set, improving its ability to guide the crawler. The KwBiLSTM classifier predicts the relevance of web pages by leveraging semantic features and keyword appearance, serving as the reward function in the RL setup. This combination of RL, keyword expansion, and a deep classifier allows TRES to prioritize URLs effectively and achieve better results than traditional methods that rely solely on classifiers or brute-force evaluations. Experimental results demonstrate that TRES outperforms state-of-the-art crawlers like ACHE and Seed Finder in terms of both harvest rate and computational efficiency. Tested across domains such as Sports, Food, and Hardware, TRES achieved superior performance with fewer computational resources by using Tree-Frontier to reduce the number of evaluated actions per step. Its ability to adapt to different environments, coupled with the effectiveness of its keyword expansion and classification components, positions TRES as a highly efficient solution for focused web crawling, setting a new standard in the field. The research [3] describes a novel focused web crawler that uses deep convolutional networks (CNN) to parse Turkish language accurately. The architecture is organized into numerous parallel components, including the control, crawler, link extractor, link sorter, and natural language processing (NLP) units. This framework facilitates the efficient crawling and processing of large amounts of Turkish web information. The natural language processing unit, driven by CNN, can accurately classify Turkish online pages, outperforming seven cutting-edge CNN models for text categorization. Notably, the web crawler can extract and sort out links, hence enhancing the classification accuracy of Turkish text by incorporating the most appropriate word embeddings, such as Bidirectional Encoder Representations from Transformer. In terms of

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2607 - 2614
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0359

technique, the crawler was tested with three datasets: one having 50,000 Turkish web pages retrieved by the crawler and two publicly accessible datasets with 28,567 and 22,431 pages, respectively. To improve classification, the paper investigates numerous sophisticated embedding approaches, with BERT proving the most effective for Turkish text. This finding emphasizes the need of proper word representation for web page classification. The CNN-based technique was tested against other models such as VdCNN, CharCNN, and TurkishCNN, demonstrating the proposed system's resilience, particularly its ability to manage noise and the dynamic nature of Turkish web material. The research emphasizes that preprocessing methods like as stemming, lemmatization, and normalization helped to significantly improve performance. The contribution of this study lies in filling a crucial gap in Turkish web processing, providing a focused crawler specifically tailored for the language. This system could enhance Turkish search engines and contribute to the creation of Turkish-specific datasets. Furthermore, the application of parallel processing methods to improve scalability and processing speed makes it an adaptable tool for crawling huge Turkish web data. The report also identifies potential areas for further research, such as processing dynamic information and supporting AJAX, which could enhance the system's capabilities. In order to guide topical web crawlers, Gautam Pant and Padmini Srinivasan's paper [4] compares three distinct classification schemes: Neural Networks (NN), Support Vector Machines (SVM), and Naive Bayes (NB). These crawlers seek out relevant web pages based on a certain topic, using classification algorithms to prioritize which URLs to follow. The paper tackles the absence of systematic comparisons of various systems by running experiments on over 100 topics to evaluate their performance in a dynamic web context. The research focuses on three classifiers: Naive Bayes, Support Vector Machines, and Neural Networks. For Naive Bayes, two versions were tested using normal density estimation and kernel density estimation. For SVM, three polynomial kernels (1st, 2nd, and 3rd-degree) were evaluated, while different architectures of Neural Networks were tested using varying numbers of hidden nodes and learning rates. Additionally, the study examines the effectiveness of these classifiers in situations with limited training data (seed URLs) and whether adaptation during crawling, by adding pseudo-examples, improves performance. The major findings show that Naive Bayes performs poorly when compared to SVM and Neural Networks due to skewed posterior probabilities, limiting its capacity to efficiently rank URLs. In contrast, SVM and Neural Networks outperformed Naive Bayes, with SVM having a modest edge in training efficiency. Although SVM and Neural Networks performed similarly in terms of precision and recall, they explored different regions of the web and returned different groups of relevant pages. When crawlers were initialized with fewer seed URLs, their performance fell, but SVM and Neural Networks surpassed Naive Bayes, exhibiting improved generalization with limited training dataset. The researchers also investigated adaptive crawling by retraining classifiers with pseudo-examples, however this method did not significantly increase performance. The findings indicate that the pseudo-examples may not have been sufficiently informative, or that the classifiers did not benefit from retraining throughout the crawl.

3. Methodology

3.1. Methods for Focused Crawling

Naive Bayes Classification: The naive bayes classifier is a probabilistic model that relies on Bayes theorem; it implies that the features such as words or keywords describing a web page are conditionally distinct given the class label. Despite this simplification, the model frequently performs well in practical scenarios, especially for tasks involving text classification.

$$\mathbf{P(C \mid F_1, F_2, \ldots, Fn)} = \frac{\mathbf{P(C)\prod_{i=1}^{n} P(Fi \mid C)}}{\boldsymbol{P(F_1, F_2, \ldots, Fn)}}$$

How it Works: Bayes' Theorem relates the conditional probability of a class given the features to the prior probability of the class and the conditional probabilities of the features given the class:

- **P ($C|F_1, F_2, \ldots, Fn$)**: The posterior probability of class C given the features F1, F2, Fn. This is what we want to compute to classify the page.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 02 Issue: 11 November 2024
Page No: 2607 - 2614
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2024.0359

- **P(C)**: The prior probability of class C, which is the frequency of that class in the training data.
- **P(Fi|C)**: The conditional probability of feature Fi given class C, computed based on the occurrences of Fi in labeled training data.
- **P (F_1, F_2, ..., Fn)**: The total probability of the features, which acts as a normalizing constant.

This classifier is trained on labeled data in the context of focused web crawling, with each web page being classified as relevant or irrelevant based on its article content. When a new web page is seen, the Classifier of Naive Bayes evaluates the page's posterior probabilities of belonging to the relevant or irrelevant class by multiplying the conditional probabilities of the page's characteristics (keywords). The page is then classified to the class with the highest posterior probability.

Support Vector Machines (SVM): SVM is a supervised learning algorithm that maximizes the margin between two classes (relevant and irrelevant) by identifying the best hyperplane. The margin is a distance between each class's closest data points, also known as support vectors, and the hyperplane. SVM is perfect for text classification issues like focused web crawling, where each page may be presented as a high-dimensional vector of features, because it works well in high-dimensional domains.

Mathematical Formulation: SVM seeks to discover a hyperplane w·x+b=0 that partitions data points from two classes, where w is the weight vector normal to the hyperplane and b is the bias factor. The issue of optimization can be stated as follows:

$$\mathbf{minimize}(\mathbf{w},\mathbf{b})\,\frac{1}{2}\,||\boldsymbol{w}||^{2}$$

Subject to limitations:

$$y_i(w\cdot x_i+b) \geq 1 \text{ for all } i$$

Here, xi is the feature vector of the i-th page, and yi is the class label (+1 for relevant, −1 for irrelevant). the purpose is to maximize the margin 1/||w|| between the two classes while guaranteeing that all of the points have been correctly classified thus satisfying the constraints

How it Works in Focused Crawling: To use SVM for focused crawling, each web page is represented as a high-dimensional vector of features extracted from its content. The training set consists of labelled web pages, and SVM finds the decision boundary that separates relevant pages from irrelevant ones. When a new web page is encountered, the trained SVM predicts its class (relevant or irrelevant) by determining on which side of the hyperplane it lies.

Decision Trees: Decision trees are hierarchical, tree-like structures which utilized to classify data. The tree's interior nodes reflect decisions based on particular data attributes, while the branches indicate the potential outcomes of those decisions. The tree's leaf nodes portray its ultimate classification (relevant or irrelevant). [8]

How it Works: The decision tree technique divides the data at each node recursively according to the characteristic that optimizes gain of information or minimizes entropy (for classification tasks). The goal is to choose the feature that best separates the data at each step. Mathematically, information gain IG(X,f) for a feature f is calculated as:

$$IG(X,f)=H(X)-H(X|f)$$

Where: H(X) is the entropy of the dataset before the split (the uncertainty or disorder). H(X|f) is the entropy after the split based on feature f, which is the weighted average entropy of the resulting subsets.

How it Works in Focused Crawling: To apply decision trees in web crawling, a labelled training set of web pages is used to build the tree. Features like keyword presence, HTML structure, and metadata are used to split the dataset. Depending on the features of the new web page the crawler moves from the root to a leaf node in the decision tree, classifying the page as relevant or irrelevant based on the majority class in the leaf node. [7] K-Nearest Neighbors (KNN): An instance-based learning technique called NN uses a majority class of its K nearest neighbors in the feature space to identify the class of a new instance, or web page. The "neighbors" are calculated utilizing a distance metric, usually Euclidean distance, however additional distance metrics, such as cosine similarity, can also be utilized for text-based applications. [6]

How it Works: The knn technique calculates the distance between a new web pages feature vector and those of all previously labeled pages in the dataset once the k nearest neighbours have been discovered the new pages class label is allocated based on the majority of these neighbours the mathematical distance d between two feature vectors x and x is commonly determined as follows:

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (xi - xi')^2}$$

Where xi and xi′ are the features of the two vectors.

How it Works in Focused Crawling: To apply KNN, the crawler treats each web page as a vector of features, such as word frequency vectors or metadata attributes. During the crawling process, when a new page is encountered, the KNN algorithm calculates the distance between this new page and the labeled pages in the database. The class of the majority of the K nearest pages is then used to classify the new page as relevant or irrelevant.

Convolutional Neural Networks (CNNs): Convolutional Neural Networks (CNNs) have been adapted for text classification and web page classification tasks by applying convolutional operations to feature maps that represent a webpage's content. Traditionally used in image processing, CNNs have found applications in natural language processing (NLP), especially when considering the local spatial structure of text or HTML, where word proximity and co-occurrence are essential for identifying relevance.

Mathematical Foundations: In CNNs, the convolution operation applies a filter (kernel) W to the input data (in this case, a matrix representing the content of a web page) to extract local features. The convolution operation for a 2D matrix X with a filter W is given by:

$$(X * W)(i, j) = \sum_{m} \sum_{n} X(i + m, j + n) W(m, n).$$

Here, X(i,j) represents the input data at position (i,j), and W(m,n) represents the filter applied to the corresponding region. This procedure is applied to the whole input matrix, yielding a feature map that highlights relevant sections (such as keywords or phrases).

Deep Reinforcement Learning (DRL): Deep Reinforcement Learning (DRL) blends deep and reinforcement learning to train agents to make decisions in dynamic situations. In focused web crawling, DRL can be used to model the crawler as an agent that makes decisions (which page to crawl) depending on the current state of the environment (the web) and the rewards it receives for its actions.

Mathematical Foundations: In DRL, the agent learns through interactions with an environment. The objective is to maximize cumulative benefits over time. The agent's behavior is represented by a policy π(a|s), Which links states s to actions a. The Q function Q(s,a) is the expected payoff for doing action a in states:

$$Q(s,a)=E[R_t \mid s,a]$$

Where Rt represents the expected cumulative reward over time t. DRL models, such as Deep Q-Networks (DQN), employ neural networks for approximating the Q-function. The agent updates its policy based on the rewards it receives after each action.

Text Tokenization: Text Tokenization is the foundational step in most NLP pipelines. It reduces unstructured text to smaller units—tokens—like words, sub words, or characters. Tokenization is necessary because most NLP models require discrete units of text as input. Tokenization can vary from word-level to sub word-level tokenization, depending on the task. [9]

Mathematical Explanation: At its core, tokenization divides a string of text into tokens. For instance, the sentence "The cat sat on the mat" would be split into the following tokens:

Tokens=["The","cat","sat","on","the","mat"]

Part-of-Speech (POS) Tagging: Part-of-Speech (POS) Tagging is a technique that labels each token in a sentence with its grammatical role, such as noun, verb, adjective, etc. It is useful for extracting the syntactic structure of text and identifying important content like keywords or subject-verb-object relationships.

Mathematical Explanation: POS tagging makes use of sequence labeling models such as Hidden Markov Models (HMMs) and Conditional Random Fields, or deep learning-based approaches. Given a sequence of tokens {w1, w2, wn}, the goal is to assign a labe yi ∈ Y to each token such that:

$$yi = POS(wi)$$

For a sentence like "Deep learning is powerful," POS tagging assigns labels like:

{"NNP" (noun),"NN" (noun),"VBZ" (verb),"JJ" (adjective)} Here, NNP refers to a proper noun ("Deep learning"), and VBZ refers to a verb in the present tense ("is").

Text Classification: Text Classification assigns a label to a document or a section of text based on its content. In focused web crawling, It is used to determine which web page is relevant to the target topic or not. Text classification algorithms typically use supervised learning techniques. [5]

Mathematical Explanation: Text classification models typically use a feature extraction technique (such as TF-IDF or word embeddings) to transform the text into a vector representation, and then a classifier (like Support Vector Machines (SVMs), Logistic Regression, or Neural Networks) to make predictions. The transformation of text into a feature vector can be represented as:

$$x = [\text{TF-IDF}(w1), \text{TF-IDF}(w2), \text{TF-IDF}(wn)]$$

where wi are the text's tokens and x are the feature vector. Once the feature vector is obtained, the classifier predicts the relevance y∈ {relevant, irrelevant} using the learned parameters w: y=classifier(w·x)

4. Results & Discussion

The paper [1] introduces a focused web crawler aimed at retrieving information about medicinal plants, utilizing a Naive Bayes classifier and a manually built thesaurus for query expansion and classification. Key contributions include its domain-specific focus, improved data relevance through a specialized thesaurus, and enhanced classification accuracy using domain-specific keywords. However, it faces limitations, such as the static thesaurus, which restricts adaptability to new terms. Additionally, reliance on Naive Bayes limits scalability and efficiency across diverse domains. The absence of modern deep learning techniques hampers the crawler's contextual understanding and automatic learning capabilities. The primary research gap identified is the need for dynamic learning mechanisms, such as Reinforcement Learning or deep learning, to enhance adaptability and classification in evolving domains.The paper [2] discusses significant advancements in focused web crawling through the implementation of Reinforcement Learning (RL) to dynamically learn optimal crawling strategies, enhancing URL selection based on the relevance of previously retrieved pages. It presents the Tree-Frontier algorithm, which simplifies large action spaces, enabling the crawler to evaluate fewer URLs while maintaining high harvest rates. Experimental results indicate that the TRES system surpasses traditional methods in retrieving relevant pages. However, the computational complexity of RL and tree-based approaches demands substantial resources, limiting usability in resource-constrained environments. Additionally, the paper does not thoroughly explore the method's generalizability across topics, relies on customized domain-specific reward functions, and initially depends on human-provided keywords. Key research gaps include the need for cross-domain applicability and addressing multilingual content challenges in global web crawling systems. The paper [3] proposes a CNN-based focused web crawler specifically for Turkish-language content, utilizing deep learning methods like CNNs for classifying web pages. A notable contribution is its effective classification of unstructured and noisy data, along with parallel processing for managing large datasets. It also employs BERT and other word embedding techniques, making it one of the first crawlers tailored for Turkish content. However, its exclusive focus on Turkish limits generalizability to other languages and multilingual contexts, while language-specific embeddings may hinder performance in mixed environments. Additionally, the deep learning models require significant computational resources, creating challenges for real-time use, with a noted gap in broader language applicability and adaptive learning mechanisms. The study[4] identifies several limitations and research gaps, notably the absence of feature selection, which could enhance classifier performance, particularly for Naive Bayes, which struggles with high-dimensional data compared to SVM or Neural Networks. The lack of feature selection may lead to irrelevant features, reducing efficiency. Additionally, Naive Bayes shows skewed performance in crawling tasks, producing extreme scores that limit effective URL prioritization. The use of static classifier models trained only at the beginning of the crawl fails to adapt to new data in real-time, which is essential for handling dynamic web content. Furthermore, the reliance on the Open Directory Project (ODP) for seed URLs may restrict the generalizability of findings, as it may not capture the full diversity of web content. The study's implementation of shallow, feed-forward Neural Networks with limited hidden layers also presents an opportunity for improvement. Future study could look into adaptive classifiers and advanced deep learning approaches like deep neural networks or CNNs to improve crawler performance by better capturing relationship, shown in Table 1.

Table 1 Literature Survey Detail

SR. NO	YEAR	AUTHOR (PUBLICATION)	RESEARCH GAP
1.	2016	Nisha Pawar, Dr. K. Rajeswari, Dr.Aniruddha Joshi(IEEE)[1]	The paper suggests improved algorithms for web crawling specific to medicinal plants, citing a manually curated thesaurus for query expansion and lack of advanced machine learning techniques.
2.	2022	Andreas Kontogiannis, Dimitrios Kelesis, Vasilis Pollatos, Georgios Paliouras, George Giannakopoulos (IEEE)[2]	The paper emphasizes optimizing crawling strategies over improving content classification, potentially reducing effectiveness if not state-of-the-art. Reinforcement learning is computationally expensive and may not scale well with large datasets.
3.	2023	Saed ALqaraleh, Hatice Meltem Nergız Sırın (The International Arab Journal of Information Technology, Vol. 20, No. 3)[3]	Limited focus on processing dynamic webpages or supporting asynchronous JavaScript and XML (AJAX), which are common in modern web pages. Further, it mainly focuses on Turkish language processing and does not explore other agglutinative languages.
4.	2005	Gautam Pant, Padmini Srinivasan (ACM Transactions on Information Systems, Vol. 23, No. 4)[4]	Further research could explore the low overlap in crawled pages between different classifiers, analyze decision-making processes, develop new metrics, and explore hybrid approaches that combine multiple classifier strengths.

Conclusion

This paper surveys focused web crawling techniques for extracting information on Ayurvedic medicinal plants, highlighting the role of deep learning. Traditional search engines often fail to meet the needs of researchers in alternative medicine, but focused crawlers enhanced by machine learning offer effective data retrieval. The review includes innovative strategies like the TRES framework, which uses reinforcement learning, and Naive Bayes classifiers with query expansion. Despite advancements, gaps remain in adaptability and scalability, especially regarding dynamic content and multilingual contexts. Future research should focus on adaptive, cross-domain crawling systems utilizing deep learning for improved access to quality information.

References

[1]. N. Pawar, K. Rajeswari and A. Joshi, "Implementation of an efficient web crawler to search medicinal plants and relevant diseases," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 2016, pp. 1-4, doi: 10.1109/ ICCUBEA.2016. 7860006.
[2]. ALkaraleh, Saed, and Hatice Meltem Nergiz Şirin. "A topic-specific web crawler using deep convolutional networks." The International Arab Journal of Information Technology, Vol. 20, No. 3, May 2023.
[3]. Kontogiannis, Andreas, et al. "Tree-based focused web crawling with reinforcement learning." arXiv preprint arXiv:2112.07620 (2022).
[4]. Gautam Pant and Padmini Srinivasan. 2005. Learning to crawl: Comparing classification

schemes. ACM Trans. Inf. Syst. 23, 4 (October 2005), 430–462. https://doi.org/10.1145/1095872.1095875

[5]. Elaraby, M. & Mohamed, Sherihan & Moftah, Hossam & Rashad, Magdi. (2019). A new architecture for improving focused crawling using deep neural network. Journal of Intelligent & Fuzzy Systems. 37. 1-13. 10.3233/JIFS-182683.

[6]. Dhanith, P. & B, Surendiran & Raja, S. (2020). A Word Embedding Based Approach for Focused Web Crawling Using the Recurrent Neural Network. International Journal of Interactive Multimedia and Artificial Intelligence. In Press. 1. 10.9781/ijimai.2020.09.003.

[7]. S. Patel, D. Agarwal, and A. Verma, "A Comparative Study of Focused Web Crawling Techniques Using Deep Learning," 2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA), vol. 6, pp. 563-570, 2018.

[8]. B. Yu, H. He, and J. Xu, "An Adaptive Focused Crawler Based on Topic-Related Graph Model," 2019 IEEE International Conference on Knowledge Engineering and Applications (ICKEA), vol. 2, pp. 287-292, 2019.

[9]. L. Sun and J. Han, "A Novel Approach to Focused Web Crawling using Supervised Learning," 2017 IEEE International Conference on Artificial Intelligence and Web Mining (AIWM), vol. 1, pp. 178-183, 2017.