Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Improved Web Page Categorization with Semantic-Aware Focused Crawling Using GloVe and TF-IDF

Li Zhang[1,] ✉

[1] *School of Artificial Intelligence, Zhejiang College of Security Technology, Wenzhou, Zhejiang, 325016, China*

ABSTRACT

Focused crawlers are targeted to search the internet for web pages on specific topics. Its main task is to collect preprocessed and topic related web pages and ignore irrelevant web pages. Traditional focused crawlers have limited success in achieving multi-text categorization of web pages. Due to the large amount of unstructured data present in web pages, the correct classification of web pages based on a given topic is the main practical challenge for focused crawlers. The main objective of this work is to design an improved focused crawling approach using web page classification.In this paper, a text classification model based on the combination of GloVe word vector model and $TF-IDF$ weighting technique is proposed to improve the accuracy of web page classification. The GloVe based text classification model is further utilized to guide focused crawlers to classify web pages. The proposed GloVe and $TF-IDF$ text categorization models are validated on 10 different datasets and the results are compared with traditional machine learning algorithms as well as different methods based on Naive Bayes, Bag-of-Words and Word2Vec. According to the experimental results, the proposed text classification model is 7-12% better than traditional machine learning algorithms.

Keywords: focused crawler, GloVe, $TF-IDF$, Text Classification

1. Introduction

In today's era of information explosion, the application of the Internet has penetrated into every aspect of our lives. As one of the basic elements of the Internet, web pages play an extremely important role. However, there are so many different web pages that it is important to categorize them in order to better organize and manage the information. By categorizing web pages, it helps to sort out and integrate the information, and users can understand and grasp the information of specific topics or domains more clearly, so as to improve the search and browsing experience of users; at the same time, it helps enterprises and individuals to carry out online marketing and promotion, and improve brand awareness and influence [1-3]. In addition, by classifying and organizing web pages, search engines can more accurately provide users with search results and

✉ Corresponding author.
E-mail address: Lily_acp@126.com (L. Zhang).
Received 25 March 2024; Revised 05 June 2024; Accepted 15 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-364

improve the accuracy and speed of search [4]. Web pages are categorized according to their content, function, and usage, and the common ones are informational web pages, e-commerce web pages, social networking web pages, media web pages, blog web pages, forum web pages, search engine web pages, and educational web pages [5-7].

Word vectors are a representation that converts words in natural language into vector form, and it has achieved remarkable success in natural language processing tasks [8]. GloVe is a well-known word vector model that combines the global statistical information of co-occurrence matrices with the local features of word vector models [9]. Co-occurrence matrix is the number of times two words appear in the same context. Its core principle is based on the linear relationships represented by word vectors, and it captures the complex semantic relationships between words by analyzing the co-occurrence statistical information of the words in order to facilitate natural language processing related tasks, such as text categorization, sentiment analysis, semantic analysis, information retrieval, etc [10-13].

TF-IDF algorithm is a commonly used algorithm for information retrieval and text mining, which can measure the importance of a word in a text [14]. TF refers to the frequency of a word in a text, the higher the frequency, the higher the importance of the word for the text [15]. However, it is inaccurate to measure the importance only by the frequency of the word, because some common words appear in most of the text but do not have actual importance. This requires the use of IDF to solve this problem. IDF refers to the inverse document frequency of a word in a collection of text, i.e., the importance of a word in the entire collection of text, the larger the value of IDF, the more important the word is [16]. The TF-IDF algorithm is to multiply the TF and the IDF to get the weight value of a word, which can be used to measure the importance of a word in a text The higher the weight value, the higher the importance of the word for the text [17]. The research based on TF-IDF algorithm mainly focuses on several aspects of text classification, keyword extraction, and text similarity calculation. Among them, the TF-IDF algorithm classifies each word by calculating the weight values of each word and then inputting these weight values as feature vectors into the classification model [18]. With the TF-IDF algorithm, the importance of each word to the text can be accurately measured, thus improving the accuracy of classification. In addition, the TF-IDF algorithm can obtain a list of keywords for a word by calculating the weight value of each word and then arranging them in descending order of the weight value [19]. These keywords can accurately reflect the topic and content of the text. And the TF-TDF algorithm can get the similarity value between two texts by calculating the sum of the weights of the words that are common to both texts [20]. The applications based on TF-IDF algorithm are very wide, such as search engines, natural language processing, content recommendation, text mining and other fields have applications [21-24].

At present, there are still some problems worth further research for multi-label text classification focused crawlers. These problems are described as follows:

A. Robustness against anti-crawling mechanisms: Some of the potential harms of malicious web crawlers, it can cause various harms and pose threats to websites and internet users. To mitigate these risks, website owners employ various techniques such as robots.txt files, CAPTCHA challenges, IP blocking, and behavior-based detection systems to identify and block malicious web crawlers.

B. Duplication of content: Crawlers may inadvertently create duplicate copies of web content if not designed to handle URL normalization or if the same content is present on multiple URLs. This duplication can lead to indexing issues, confusion in search engine results, and waste of storage resources.

C. Data incompleteness and freshness: Web content is dynamic, and crawlers can have difficulties keeping up-to-date with changes on websites. There may also be limitations in capturing content from web pages that require user logins or interactions. This can lead to incomplete or outdated data in the crawled dataset.

To address the limitations of traditional web crawling algorithms and tackle the challenges they pose, we propose a deep learning-based specialized focused crawler algorithm. This algorithm first employs $TF-IDF$ vectorization to represent link texts and then employs a logistic regression model to predict the most relevant links. The focus is achieved by selecting the link with the highest score predicted by the model. Our approach comprises two key modules: a feature selection module based on deep reinforcement learning and a classification module composed of deep neural networks. The feature selection module optimizes the filtering process based on the output of the classification module, while the classification module detects crawlers using the feature set provided by the feature selection module.

This paper makes the following contributions:

A. We introduce a Text Feature Representation Method that addresses the diversity of crawlers, and the chosen feature set exhibits greater applicability.

B. We propose a deep learning-based web crawler detection method that exhibits high accuracy and precision.

The content of this paper is organized as follows: In the second part, we introduce the recent work in the field of crawler detection. The third part introduces the Multi-Label Text classification method for the new types of crawlers based on deep learning. We give the experimental results in the fourth and the fifth part. The sixth part is the summary of this article.

2. Overview

There are few scholars who have studied web page classification, in terms of accuracy, literature [25] uses recurrent neural network to classify web pages by title, description, and keywords, the success rate of this method is at 85%, and this success rate does not change significantly by training with migration learning. Incidentally, it is mentioned that literature [26] used migration learning with training model to learn the context, modeled with deep Inception with residual connectivity, and classified web pages with an integrated approach of parallel multiscale semantic fine-tuning of the target task, which outperformed other migration learning classification methods. Literature [27] preprocesses the long text of web pages to visualize high-frequency words in the text to form a document with word cloud images, and then using convolutional neural networks, these high-frequency words are identified to achieve classification with a success rate higher than 85%. In a later study, this author utilized web browsing records, and after cleaning and preprocessing these data to form a word cloud image, the same convolutional neural network was then used to construct a classification model [28]. However, this method only classified whether the browsing belonged to games or online video streams, and both were based on word cloud image classification of high-frequency words, which has some limitations. When some common high-frequency words appear, it will interfere with this classification result. Literature [29] fused web page text heterogeneous features with the tree structure features of HTML tags in vectors and then combined them with multiple classifiers to classify web pages, with an accuracy of up to 95%, and this classification is better than the classification based on text features only. Literature [30] combines four feature methods, Jaccard similarity, information gain, gain ratio, and chi-square analysis with four classification algorithms, namely, support vector machine, k-nearest neighbor, plain Bayes, and J48, on the basis of word similarity to classify crime web pages, and the combination of Jaccard similarity+support vector machine has the best result in terms of accuracy, which is as high as 98%.

In terms of text, vocabulary, and image features, literature [31] used HTML tags as features to classify Internet web pages by a parliamentary optimization algorithm, which was effective but not significant. Literature [32] classified the words in semantic networks such as Wikipedia into entities, then used the weight estimation algorithm to calculate the class weights of the entity words, and combined with the entity similarity to perform the final classification of the page, which is effective for the classification of massive web pages. Literature [33] designed a SMGCN web page classification

method that consists of a combination of multiview construction for obtaining the optimal graph structure and view training i.e., semi-supervised multiview convolutional representation learning for attention schemes. Literature [34] performs web page classification supported by multilayer domain ontology classification of internet mining techniques which involves setting up an invisible distiller to eliminate pages from relevant domains and discarding outliers using graph based classification techniques. Literature [35] uses TF-IDF algorithm for web page classification in combination with machine learning based on natural language processing and text mining techniques which emphasizes on preprocessing of text. Literature [36] uses Glove model to obtain semantic features of web pages and extracts their contextual knowledge with stacked bi-directional long and short term memory algorithm for web page classification, which outperforms machine learning and deep learning algorithms.

3. Related work

3.1. The Realization Theory of Web Crawler

The primary function of a web crawler system is to fetch web page data, serving as a crucial data source for search engine systems. Many prominent web search engine systems of considerable scale are recognized as web data-driven search engines, such as Google and Baidu.

In this paper, focus crawler is proposed with the aim of acquiring more topic-specific information. It operates by adhering to a pre-defined crawling topic. Given an initial set of seed URLs, an analysis algorithm tailored to the topic is employed to conduct relevant content analysis on the web pages slated for crawling. This process involves the filtering out of pages that do not align with the designated topic, while retaining links pertinent to the topic within the queue for subsequent crawling iterations. This iterative process continues until specific conditions are met.

It's important to note that the URL seed set utilized by the focus crawler must consist of pre-defined pages highly correlated with the chosen topic. The focus crawler exclusively concentrates on page links that bear relevance to the designated topic, actively seeking to identify and retrieve as many topic-related pages as possible during the crawling operation. This focused approach serves to minimize the download of unrelated or extraneous web pages.

Focused Crawler can selectively fetch pages related to pre-defined topics. Compared with using the web crawler, it greatly saves hardware and network resources, and the number of saved pages is small enough to be updated quickly, so it can satisfy the need for information in a specific field.

This paper divides focused crawlers into three categories: Content Evaluation & Hyperlink Analysis Focused Crawler(CEHAFC), Augmented Learning Focused Crawler(ALFC) and In Context Graphs Focused Crawler(CGFC). These three categories of focused crawlers are described in the following.

3.2. Content Evaluation & Hyperlink Analysis Focused Crawler(CEHAFC)

$$sim(q,p)=\frac{\sum_{k\in q\cap p} W_{kq}\times W_{kp}}{\sqrt{\left(\sum_{k\in p} W_{kp}^2\times\left(\sum_{k\in q} W_{kq}^2\right)\right)}} \tag{1}$$

When crawling a new web page, the following formula (1) is usually used to calculate the relevance, and the relevance is used to determine whether it is relevant to the topic:

Where q represents the topic, p represents the web page document, Wkp represents the weight of term k in document p, the weight of term is generally calculated by $TF-IDF$ method, Wkq represents the main The feature word weight of the question. The result is the relevance of the page to the topic.

Set a relevancy threshold, if the new page is less relevant to the topic, this page is not relevant, then discard this page. Otherwise, the web page is judged to be related Page to extract all URL links contained in the page. And then judge each of these chains one by one Whether the connection already exists in URL queues. If so, the link is deleted and its The residual link is saved in queue as a link to be crawled. This mechanism avoids duplication Crawl web pages with the same link, saving network bandwidth and hardware resources.

If the web page is related to the topic, all URL links contained in the web page are extracted and they have the same priority. The priority is usually calculated using the following formula(2).

$$priority\left(U_{pi}\right)=\frac{sim\left(q,p\right)}{N_p} \tag{2}$$

q denotes the topic, p denotes the web document, sim(q,p) denotes the relevance of web page p to the topic q, Np denotes the number of URL links contained in web page p, Upi is the ith link of web page p, and the result of the computation is the prioritization value of link Upi.

3.3. Augmented Learning Focused Crawler(ALFC)

Using a Bayesian classifier, hyperlinks are categorized based on the entire web page text and link text, and the weight of each link is calculated to determine the order in which the links are accessed. Its basic principle is: in the case of known a priori probability of the class and the conditional concept of all the values of each attribute in the class, can calculate the conditional probability of belonging to a certain class of samples to be categorized, and select the largest conditional probability of the sample as the classification.

3.4. Context Graphs Focused Crawler(CGFC)

Context Graphs Focused Crawler can solve the problem of low efficiency of traditional topic crawlers in searching topic web pages, which is a crawling strategy based on web topology modeling. The Context Graphs method belongs to the heuristics based on the structure of web hyperlinks, which not only takes into account the citation relationship between web pages but also abstracts the hierarchical model of URLs, making the crawling guidance strategy more reasonable. It not only considers the citation relationship between web pages but also abstracts a hierarchical model for URLs, which makes the crawling guidance strategy more reasonable.

The Context Graphs method builds a hierarchical model of the topic based on a collection of seeds. A typical Context Graphs model is shown in Figure 1. Other web pages on the path to the target topic are categorized into different levels of the model according to their distance from the target topic: the seed web page is at level 0, web pages that can link directly to the seed are categorized at level 1, web pages that link directly to level 1 (web pages with a distance of 2 from the seed web page) are placed at level 2, and so on. The more steps it takes to link to the target topic page, the closer the page is to the center level of the model, the more relevant or connected it is to the topic, while the pages in the outer levels of the model are likely to be general directory pages or lead pages.

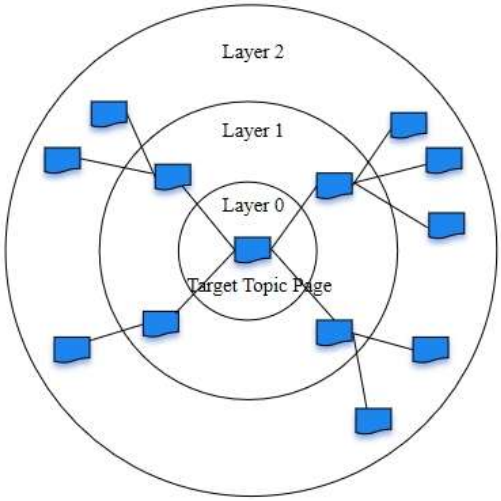

Fig. 1. An example of Context Graphs Model

4. Focused crawler based on GloVe and TF-IDF

The dynamic nature of the web is a challenge in related computation for focused web crawlers. we propose a new crawler detection method based on deep leaning. In order to determine the relevance of URLs both syntactically and semantically, we employ a word embedding approach for computing the relevance of web pages. We calculate the cosine similarity between the global vector representations of a topic and the contents of the web page. The computed cosine similarity is utilized as input to the trained random forest classifier, which predicts the relevance of the web page.

Our model consists of two modules: a semantic focused crawler based on Global Vectors Model (GloVe) and Term Frequency Inverse Document Frequency Algorithm ($TF-IDF$). This focused crawler divides the text contents of unvisited URLs into two parts including feature selection and session classification. The $TF-IDF$ constructs text semantic vectors of the above two texts and the topic semantic vector, and utilizes the cosine similarity between the text semantic vector and the topic semantic vector as the topic similarity of the text (see Figure 2).

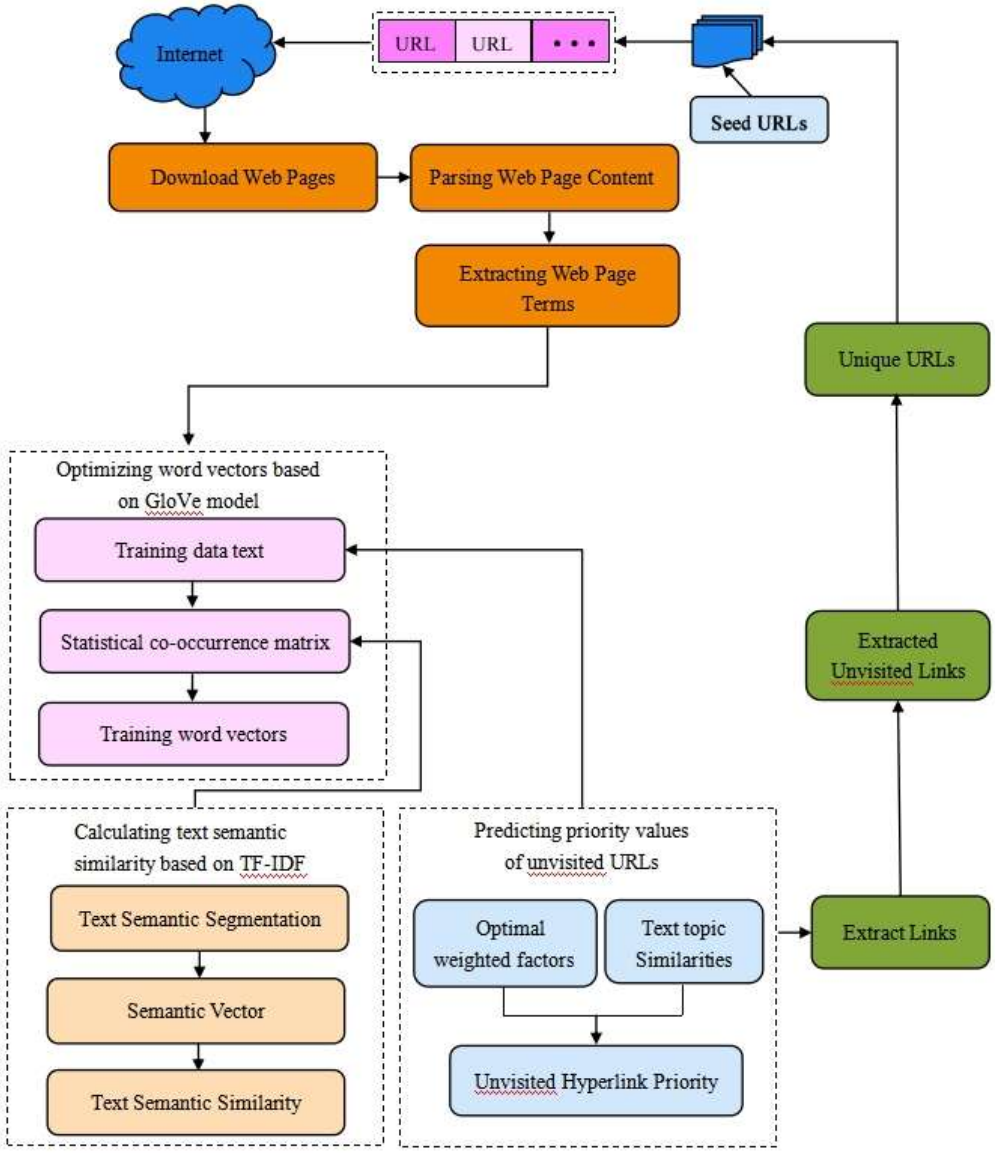

Fig. 2. Proposed Framework of Focused Crawler

4.1. Gobal Vectors Model

To extract the cosine-based feature, this work uses GloVe word embedding. The co-occurrence matrix M will be a V×V matrix comprising the i th row and j th column of M for a given corpus of V words. M_{ij} specifies the count the word i has co-occurred with the word j, and M_{ji} denotes the count the word j has co-occurred with the word i and represented in Equation (3).

$$vec_i^T \times vec_j = logP(j \mid i) \tag{3}$$

where vec_i and vec_j form the embedding vector of word i and j respectively.

$$P(j \mid i) = \frac{M_{ij}}{\sum M_{ij}} = \frac{M_{ij}}{M_i} \tag{4}$$

$$P(i \mid j) = \frac{M_{ji}}{\sum M_{ji}} = \frac{M_{ji}}{M_j} \tag{5}$$

The value of Equation (4) is substituted in Equation (3), which is derived as follows in Equations (6) and (7):

$$vec_i^T \times vec_j = log\left(\frac{M_{ij}}{M_i}\right) \tag{6}$$

$$vec_i^T \times vec_j = log\left(M_{ij}\right) - log\left(M_i\right) \tag{7}$$

$$vec_j^T \times vec_i = log\left(\frac{M_{ji}}{M_j}\right) \tag{8}$$

It is to be noted that $log\left(M_i\right)$ and $log\left(M_i\right)$ depend only on the words i and j. Equation (8) can be written as scalar biases b_i and b_j, respectively, in Equation (9).

$$vec_i^T \times vec_j = log\left(M_{ij}\right) - b_i - b_j \tag{9}$$

The optimization problem can be formulated as in Equation(10) :

$$\min \sum_{i,j} f\left(M_{ij}\right)(vec_i^T \times vec_j + b_i + b_j - \log\left(1 + M_{ij}\right))^2 \tag{10}$$

Algorithm 1 Combining TF-IDF and GloVe feature vectors

```
Input: train_texts, train_labels
Output: train_features
# Constructing TF-IDF feature vectors
tfidf_vectorizer = TfidfVectorizer()
train_tfidf = tfidf_vectorizer.fit_transform(train_texts)
# Constructing GloVe feature vectors
dim = 300    # Dimension of GloVe word vectors
train_glove = np.zeros((len(train_texts), dim))
for i, text in enumerate(train_texts):
tokens = text.split()
for token in tokens:
if token in glove_model:
train_glove[i] += glove_model[token]
# Merging TF-IDF and GloVe feature vectors
train_features = np.concatenate((train_tfidf.toarray(), train_glove), axis=1)
```

4.2. Word embedding

When entering the text data into the model for training or prediction, we need to pre-process the text.In this works, Use the natural language processing kit NLTK (Natural Language Toolkit) to divide the text, remove stop words and stem reduction, and get the cleaned text $x = \{w_1, w_2, \ldots, w_N\}$ composed of N words. In order to map the original text onto the vector space for easy computer recognition and computation, We use the GloVe word embedding method to transform each word w_i in the text into its corresponding word vector $e_n \in R^{D_1}$ to obtain the word embedding representation $\{e_1, e_2, \ldots, e_N\}$ of the input text sequence, where D_1 is the dimension of the word vector.

The GloVe method is based on the idea of matrix decomposition, which decomposes the lexical co-occurrence matrix into two low-dimensional matrices, one of which represents the relationship between words and the other represents the vector representation of words. A global word

co-occurrence matrix is first built, where the matrix elements represent the number of times two words appear in a text window at the same time, and then the word vector representations are obtained by performing singular value decomposition (SVD) on this matrix. Compared with other word embedding methods, the GloVe method considers the global co-occurrence relationship between words, rather than just limiting to the word vector representation. In contrast to other word embedding methods, the GloVe method takes into account the global co-occurrence relationship between words, rather than limiting itself to the occurrence of each word in the context. As a result, it is able to better capture the semantic relationships between words, such as the relationship between synonyms and antonyms.

4.3. Label embedding methods

All the texts and labels in the training set are first input into the Gobal Vectors Model, and then the label-word probability distribution $\beta \in R^{L \times V}$ is generated by the Gobal Vectors Model training, where V denotes the size of the whole vocabulary. According to the current input text $x = \{w_1, w_2, ..., w_N\}$ of Gobal Vectors Model to generate the corresponding label embedding vector matrix P ,label embedding vector matrix $P = \{p_1, p_2, ..., p_L\}, P \in R^{L \times N}$ can represent the importance of each word of the current input under different labels, which is introduced into the Gobal Vectors Model and can effectively guide the model to map the word vectors to the correct labeling direction. In order to understand the initialization process of the label embedding vector more intuitively.

4.4. Feature Extraction Layers

In this paper, we used $TF - IDF$ for text feature extraction. $TF - IDF$ is used to evaluate the importance of a word for one of the documents in a document set or a corpus for extracting features of the text.

The $TF - IDF$ is calculated as follows:

$$TF - IDF = TF * IDF \tag{11}$$

TF is Term Frequency, IDF is Inverse Document Frequency.

TF is the word frequency, i.e., the frequency of occurrence of a word in a document, assuming that a word occurs times in the whole document, and the whole document has N words, then the value of TF is $\frac{1}{N}$.

IDF for the reverse document frequency, assuming that the entire document has n articles, and a word appears in k articles, then the value of IDF is

$$IDF = log2(n / k) \tag{12}$$

Assuming that the whole document has D articles, the $TF - IDF$ value of word i in the j th article is:

$$weight_{i,j} = frequency_{i,j} * \log_2 (D / document\_freq_i) \tag{13}$$

In order to do feature extraction, we have used a textual paragraph related to the crawling topic. The textual data is then converted into tokens which are used as features. The Algorithm2 describes the feature extraction process.

Algorithm 2 Constructing feature vectors for test data

Input: test_texts, test_labels
Output: test_features
Reading test data

```
test_texts = ['test1', 'test2', 'test3', ...]
# Constructing TF-IDF feature vectors for test data
test_tfidf = tfidf_vectorizer.transform(test_texts)
# Constructing GloVe feature vectors for test data
test_glove = np.zeros((len(test_texts), dim))
for i, text in enumerate(test_texts):
    tokens = text.split()
    for token in tokens:
        if token in glove_model:
            test_glove[i] += glove_model[token]
# Merging TF-IDF and GloVe feature vectors of test data
test_features = np.concatenate((test_tfidf.toarray(), test_glove), axis=1)
# Multi-label categorization of predictive test data
predictions = classifier.predict(test_features)
```

4.5. Lables and Words Interaction Layer

In this paper, we use the Gobal Vectors Model to train the multilabeled text dataset to generate the label-word probability distribution $\beta \in R^{L\times V}$, where V denotes the size of the whole vocabulary. Then, the label embedding vector matrix $P=\{p_1, p_2, ..., p_L\}$ is obtained by β initializing the label embedding vectors of the current input $\{w_1, w_2, ..., w_N\}$, where is trainable, and this process is similar to the process of mapping words to word embedding vectors. Finally, the label embedding vector matrix P is pointwise multiplied with the word feature vector matrix H computation, which makes the label interact with the word, The formula is as follows:

$$U = PH \tag{14}$$

It can be rewritten in matrix multiplication mathematical form:

$$u_j = \sum_{n=1}^{N} p_{jn} h_n \tag{15}$$

$$U = \{U_1, U_2, ..., U_L\} \tag{16}$$

The label-word probability distribution can be viewed as the importance of each word for a particular label, and the principle behind the dot product is to compute u_i. by using p_{jn} as h_n weights in a weighted summation. where p_{jn} denotes the probability of the nth word appearing in the jth label, h_n denotes the feature vector of the nth word, and u_i denotes the textual representation for the jth label. The step is repeated and finally the text feature vector matrix $U \in R^{L\times D_2}$ is computed for a particular tag.

4.6. Output Layer

After obtaining label-specific text features U in the label interaction layer, we use a multilayer perceptron with two fully-connected layers to construct a multilabel text classifier that predicts the probability that the input text belongs to each label. The formula is as follows:

$$\hat{y} = sigmoid\left(W_1 f\left(W_2 U^T\right)\right) \tag{17}$$

$W_1 \in R^b, W_2 \in R^{b\times D_2}$ are trainable parameters, f is an nonlinear activation function, and the sigmoid function is used to map the output onto the interval (0,1).

5. Experimental design

Table 1 shows the experimental tools and environment. The focused crawlers used in this work are deep learning, GloVe, $TF-IDF$, semi-supervised. The prototype of the crawlers was developed in Python3.10 using the pycharm platform. All the focused web crawlers uses 10 topics, including Car, Education, Animals, Health, Arts, Finance, Tourism, Sports, History & Society, Games & Quizzes, along with their seed URLs as input. A set of 10 topics and their respective seed URLs are shown in Table 2.

Table 1. Experimental tools and environment

Name	Configurations
Operating System	Windows 11
CPU	2.60GHz 2.59 GHz
GPU	TCC/WDDM
CUDA	12.1
Python	3.10

Table 2. Topic and seed URLs

S.No	Topic	Seed URL
1	Car	https://www.caranddriver.com/news/
		https://www.edmunds.com/
2	Education	https://www.statista.com/topics/2090/education-in-china/#topicOverview
		https://www.unesco.org/en/education
3	Animals	https://kids.nationalgeographic.com/animals/
		https://www.britannica.com/animal/animal
4	Health	https://www.who.int/health-topics/
		https://www.britannica.com/topic/health
5	Arts	https://www.olymp-arts.org.cn/
		https://www.britannica.com/Arts-Culture
6	Finance	http://www.chinadaily.com.cn/business/money
		https://www.britannica.com/money/topic/finance
7	Tourism	https://www.britannica.com/topic/tourism
		https://ourworldindata.org/tourism
8	Sports	https://www.britannica.com/sports/sports
		http://www.chinadaily.com.cn/sports/
9	History & Society	https://www.britannica.com/History-Society
10	Games & Quizzes	https://www.britannica.com/quiz/browse

6. Performance evaluation

6.1. Performance metrics

The four metrics in this work that evaluates the performance of the training phase for the various machine-learning algorithms are given in Table 3.

Table 3. Metrics for performance of training phase

Metric	Symbols	Objective	Formula
Accuracy	accu	To compute the overall accuracy of the model	$accu = \frac{}{tp+tn+fp+fn}$
Precision	pre	To compute the preciseness of the model	$pre = \frac{}{N}\sum_{i=1}^{} \frac{}{\lvert a_i \rvert}$
Recall	rec	To find the efficiency of the model	$\frac{}{\lvert D \rvert}$
F1-score	f1	To compute the aggregate performance of the model	$f1 = \frac{}{pre+rec}$
Harvest rate	hr	To compute the downloading capability of the focused crawler	$hr = \frac{1}{N}\sum_{i=1}^{N} n_i$
Irrelevance ratio	ir	To compute the inaccuracy of the focused crawler	$ir = \frac{}{N}\sum_{i=1}^{} \frac{}{\lvert D' \rvert}$

N_i is the number of relevant web pages downloaded, N is the total number of web pages downloaded, D is the set of relevant web pages in the target set, and D' is the set of irrelevant web pages in the target set.

6.2. Analysis

In this paper, we experimentally determine the best classifier, which efficiently computes the relevance of web pages. We collected a datasets of 600,000 topic-page pairs across 10 topics. Each topic consists of 48,0000 training data and 12,0000 test data. Word embedding, label embedding, feature extraction, label and word interaction, and output are performed on the topic and web page data. After preprocessing the training data, word embedding-based cosine similarity is extracted from the loaded GloVe model as features for each topic web page pair. The extracted features are used to train Word2Vec, Bag-of-Words, Naive Bayes classifiers. After training the classifiers, a test datasets of topic web pages is used to test the performance of the classifiers. The training phase is evaluated using four metrics: accuracy, recall, f1 score and precision. Table 4 shows the comparative results of the four classifiers on 48,000 training datasets. The accuracy of Word2Vec, Bag-of-Words, Naive Bayes and GloVe is 0.9312, 0.8260, 0.8321 and 0.9538 respectively. The comparative results clearly show that the crawling method based on the GloVe model and $TF-IDF$ multi-label text classification outperforms the other methods based on Naive Bayes, Bag-of-Words and Word2Vec.

Table 4. Precision, recall, F1-score and accuracy with 480,000 training samples

Algorithm	Precision		Recall		F1-score		Accuracy
	Class1	Class0	Class1	Class0	Class1	Class0	
Naive Bayes	0.72	0.73	0.75	0.72	0.72	0.71	0.8321
Bag-of-Words	0.80	0.81	0.83	0.81	0.80	0.82	0.8260
Word2Vec	0.83	0.82	0.82	0.81	0.80	0.83	0.9312
GloVe+TF-IDF	0.91	0.93	0.92	0.91	0.90	0.92	0.9538

6.3. Results summary

We conducted 10 different topic categories of experiments to determine the crawling performance of the four crawlers in terms of seed URLs being directly and indirectly related to the topic. The category wise accuracy comparison 10group datasets on ten different topics is shown in Figure 3, The results show that the GloVe model and $TF-IDF$ crawlers outperform the existing Naive Bayes, Bag-of-Words and Word2Vec crawlers.

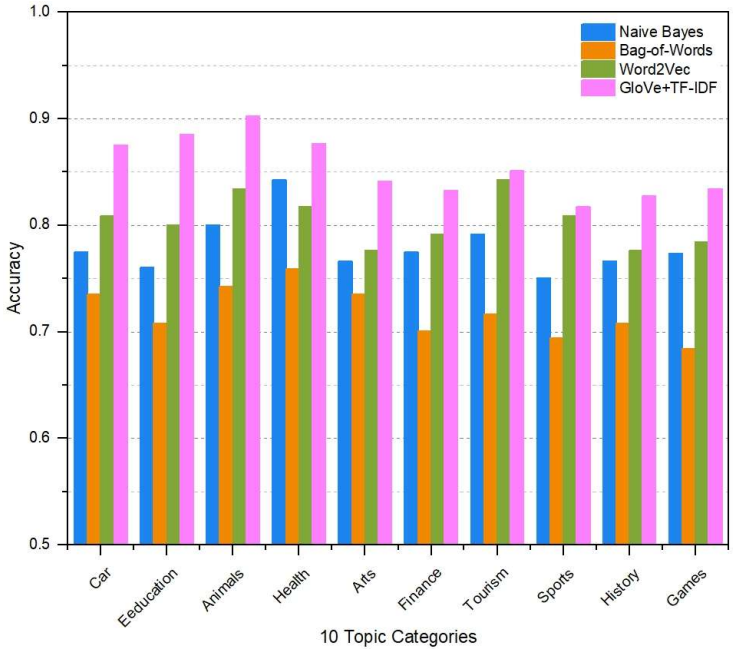

Fig. 3. Category wise accuracy comparison 10 group datasets on ten different topics

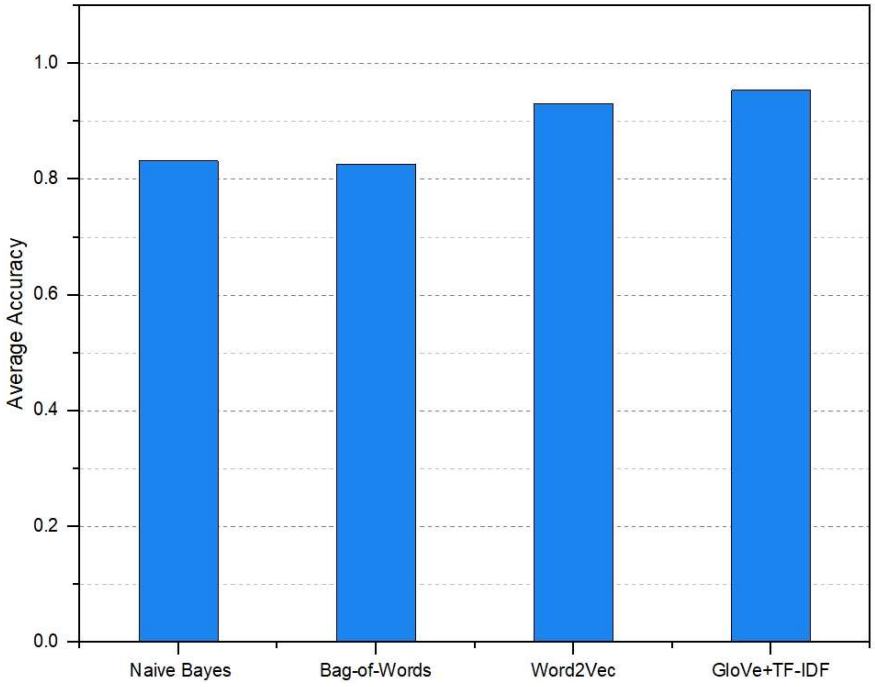

Fig. 4. Average accuracy comparison of different algorithms over 10 topic datasets

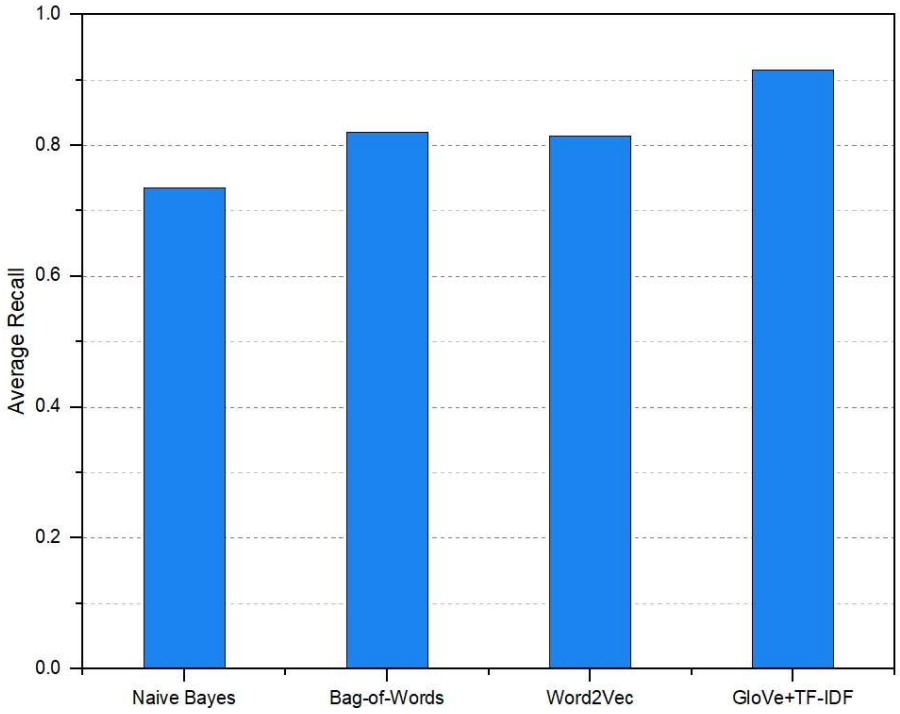

Fig. 5. Average target recall comparison for 10 different topics

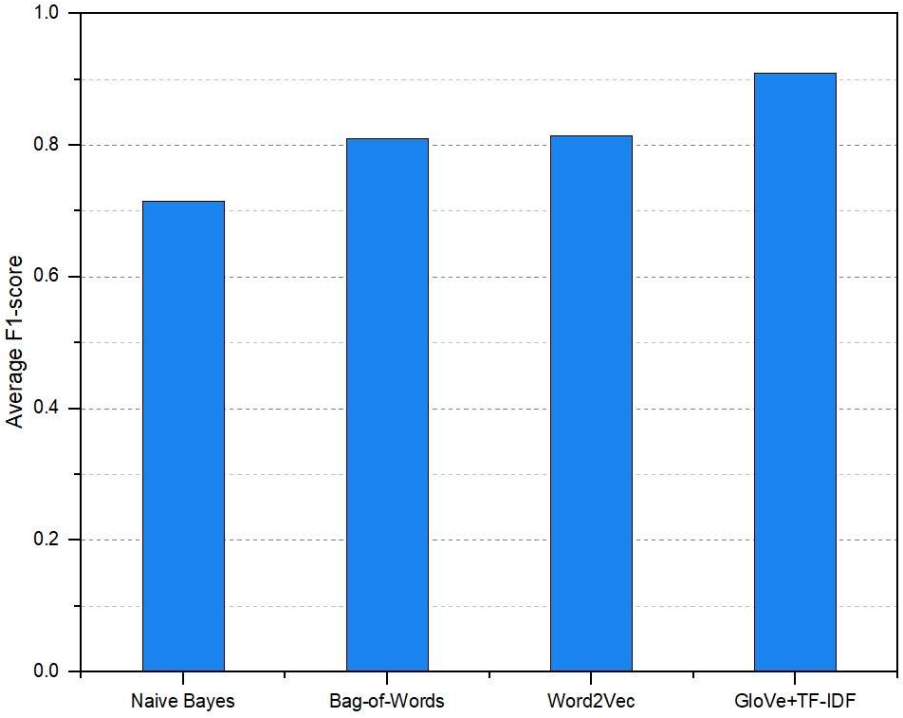

Fig. 6. Average target F1-score comparison for 10 different topics

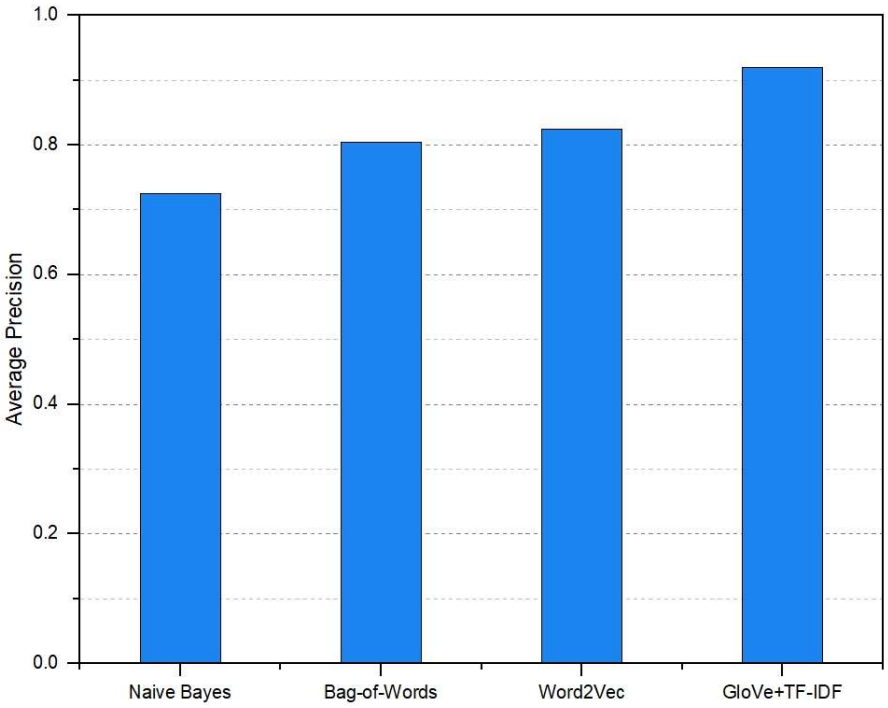

Fig. 7. Average target Precision comparison for 10 different topics

Figure 3 is illustrating Category wise accuracy comparison 10 group datasets on ten different topics. The Glove+ $TF-IDF$ approximately 91.6% which is 1.35% better than Word2Vec, 2.76% better than Naive Bayes, and 4.56% better than Bag-of-Words.

Figure 4 is illustrating average accuracy comparison of different algorithms over 10 topic datasets for the GloVe + $TF-IDF$, Naive Bayes, Bag-of-Words and Word2Vec is95.3%, 83.2%, 82.6% and 93.1% respectively.

Figure 5 is illustrating Average target recall comparison for 10 different topics for the GloVe + $TF-IDF$, Naive Bayes, Bag-of-Words and Word2Vec is91.5%, 73.5%, 82% and 81.5% respectively.

Figure 6 is illustrating Average target F1-score comparison for 10 different topics for the GloVe + $TF-IDF$, Naive Bayes, Bag-of-Words and Word2Vec is91%, 71.5%, 81% and 81.5% respectively.

Figure 7 is illustrating Average target precision comparison for 10 different topics for the GloVe + $TF-IDF$, Naive Bayes, Bag-of-Words and Word2Vec is92%, 72.5%, 82.5% and 80.5% respectively.

7. Conclusion

This study proposes an improved focused crawling method using web page categorization to accomplish all the specified goals. The key objective of the proposed work is to implement an effective Focused Crawler approach. In this regard, the first requirement is to design an effective text classifier to work with Focused Crawler. For this purpose, a web page based on the GloVe model and $TF-IDF$ multi-label text classification model is designed to classify web pages on specific topics. Experimental results show that the text classification model based on the GloVe model and $TF-IDF$ proposed in this paper outperforms the classical machine learning methods such as Naive Bayes, Bag-of-Words and Word2Vec in terms of precision, recall, f1-score and accuracy.

Competing Interests

The authors declare that they have no competing interests related to this research work.

Author's Contribution Statement

The sole author of this study, conceived the research idea, designed the study, conducted the experiments, analyzed the data, and wrote the manuscript.

Ethical Considerations and Informed Consent for Data Usage

The research conducted in this study adhered to ethical principles and obtained informed consent from all participants involved in the data collection process, ensuring compliance with ethical standards throughout the research.

Data Availability and Access

The data supporting the findings of this study are available upon reasonable request. Interested parties may contact the corresponding author for access to the data.

References

[1] Osanyin, A., Oladipupo, O., & Afolabi, I. (2018). A review on web page classification. Covenant Journal of Informatics and Communication Technology.

[2] Sunita, Singh, G., & Rana, V. (2021). A method for webpage classification based on URL using clustering. Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science), 14(2), 442-447.

[3] Hashemi, M. (2020). Web page classification: a survey of perspectives, gaps, and future directions. Multimedia Tools and Applications, 79(17), 11921-11945.

[4] Matošević, G., Dobša, J., & Mladenić, D. (2021). Using machine learning for web page classification in search engine optimization. Future Internet, 13(1), 9.

[5] Espinosa-Leal, L., Akusok, A., Lendasse, A., & Björk, K. M. (2021). Website classification from webpage renders. In Proceedings of ELM2019 9 (pp. 41-50). Springer International Publishing.

[6] Ali, F., Khan, P., Riaz, K., Kwak, D., Abuhmed, T., Park, D., & Kwak, K. S. (2017). A fuzzy ontology and SVM–based Web content classification system. IEEE Access, 5, 25781-25797.

[7] Nainwani, P. V., & Prajapati, P. (2018). Comparative study of web page classification approaches. Int J Comput Appl, 179, 6-9.

[8] Camacho-Collados, J., & Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. Journal of Artificial Intelligence Research, 63, 743-788.

[9] Xue, C., Fei, R., & Yao, Q. (2024, April). Text Representation Based on WT-GloVe Word Vector Weighting Model. In 2024 International Conference on Intelligent Computing and Robotics (ICICR) (pp. 131-136). IEEE.

[10] Sari, W. K., Rini, D. P., & Malik, R. F. (2019). Text classification using long short-term memory with glove. Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI), 5(2), 85-100.

[11] Pimpalkar, A. (2022). MBiLSTMGloVe: Embedding GloVe knowledge into the corpus using multi-layer BiLSTM deep learning model for social media sentiment analysis. Expert Systems with Applications, 203, 117581.

[12] Mohammed, S. M., Jacksi, K., & Zeebaree, S. (2021). A state-of-the-art survey on semantic similarity for document clustering using GloVe and density-based algorithms. Indonesian Journal of Electrical Engineering and Computer Science, 22(1), 552-562.

[13] Raffly, F. P. M. R., & Setiawan, E. B. S. (2024). Feature Expansion with GloVe and Particle Swarm Optimization for Detecting the Credibility of Information on Social Media X with Long Short-Term Memory (LSTM). Scientific Journal of Informatics, 11(3), 621-634.

[14] Zhang, L. (2021). Research on case reasoning method based on TF-IDF. International Journal of System Assurance Engineering and Management, 12(3), 608-615.

[15] Dogan, T., & Uysal, A. K. (2019). On term frequency factor in supervised term weighting schemes for text classification. Arabian Journal for Science and Engineering, 44, 9545-9560.

[16] Beel, J., Breitinger, C., & Langer, S. (2017). Evaluating the CC-IDF citation-weighting scheme: how effectively can 'Inverse Document Frequency'(IDF) be applied to references. Proceedings of the 12th iConference.

[17] Zhu, Z., Liang, J., Li, D., Yu, H., & Liu, G. (2019). Hot topic detection based on a refined TF-IDF algorithm. IEEE access, 7, 26996-27007.

[18] Xiang, L. (2022). Application of an Improved TF ‐ IDF Method in Literary Text Classification. Advances in Multimedia, 2022(1), 9285324.

[19] Zhuohao, W. A. N. G., Dong, W. A. N. G., & Qing, L. I. (2021). Keyword Extraction from Scientific Research Projects Based on SRP ‐ TF ‐ IDF. Chinese Journal of Electronics, 30(4), 652-657.

[20] Lan, F. (2022). Research on Text Similarity Measurement Hybrid Algorithm with Term Semantic Information and TF ‐ IDF Method. Advances in Multimedia, 2022(1), 7923262.

[21] Xu, R. (2014, July). POS weighted TF-IDF algorithm and its application for an MOOC search engine. In 2014 International Conference on Audio, Language and Image Processing (pp. 868-873). IEEE.

[22] Bounabi, M., Elmoutaouakil, K., & Satori, K. (2021). A new neutrosophic TF-IDF term weighting for text mining tasks: text classification use case. International Journal of Web Information Systems, 17(3), 229-249.

[23] Zhu, Z. (2024). Personalized new media marketing recommendation system based on TF-IDF algorithm optimizing LSTM-TC model. Service Oriented Computing and Applications, 1-13.

[24] Yu, N. (2018, August). A visualized pattern discovery model for text mining based on TF-IDF weight method. In 2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC) (Vol. 2, pp. 183-186). IEEE.

[25] Buber, E., & Diri, B. (2019). Web page classification using RNN. Procedia Computer Science, 154, 62-72.

[26] Gupta, A., & Bhatia, R. (2021). Ensemble approach for web page classification. Multimedia Tools and Applications, 80(16), 25219-25240.

[27] Apandi, S. H., Sallim, J., & Mohamed, R. (2023). A Convolutional Neural Network (CNN) Classification Model for Web Page: A Tool for Improving Web Page Category Detection Accuracy. JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 4(3), 110-121.

[28] Apandi, S. H., Sallim, J., Mohamed, R., & Ahmad, N. (2024). Data Pre-processing of Website Browsing Records: To Prepare Quality Dataset for Web Page Classification. JOIV: International Journal on Informatics Visualization, 8(1), 239-246.

[29] Deng, L., Du, X., & Shen, J. Z. (2020). Web page classification based on heterogeneous features and a combination of multiple classifiers. Frontiers of Information Technology & Electronic Engineering, 21(7), 995-1004.

[30] Markkandeyan, S., Selvam, L., Tamizharasu, K., & Aandi, S. (2024). Experimental Analysis of the Machine Learning Algorithms for Crime Web Page Classification. IETE Journal of Research, 70(5), 4890-4902.

[31] Kiziloluk, S., & Ozer, A. B. (2017). Web pages classification with parliamentary optimization algorithm. International Journal of Software Engineering and Knowledge Engineering, 27(03), 499-513.

[32] Li, H., Xu, Z., Li, T., Sun, G., & Choo, K. K. R. (2017). An optimized approach for massive web page classification using entity similarity based on semantic network. Future Generation Computer Systems, 76, 510-518.

[33] Wu, F., Jing, X. Y., Wei, P., Lan, C., Ji, Y., Jiang, G. P., & Huang, Q. (2022). Semi-supervised multi-view graph convolutional networks with application to webpage classification. Information Sciences, 591, 142-154.

[34] Saleh, A. I., Al Rahmawy, M. F., & Abulwafa, A. E. (2017). A semantic based Web page classification strategy using multi-layered domain ontology. World Wide Web, 20, 939-993.

[35] Babapour, S. M., & Roostaee, M. (2017, December). Web pages classification: An effective approach based on text mining techniques. In 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI) (pp. 0320-0323). IEEE.

[36] Nandanwar, A. K., & Choudhary, J. (2021). Semantic features with contextual knowledge-based web page categorization using the GloVe model and stacked BiLSTM. Symmetry, 13(10), 1772.