DOI: 10.24996/ijs.2024.65.12.35

ISSN: 0067-2904

Efficient Streamlined Online Arabic Web Page Classification Using Artificial Bee Colony Optimization

Hamza Dabjan*, **Mohamad-Bassam Kurdy**

Syrian Virtual University, Homs, Syria

Received: 14/6/2023 Accepted: 1/12/2023 Published: 30/12/2024

Abstract

In the realm of online information retrieval, Arabic web content presents unique challenges due to the complexity of the Arabic language and the varying quality of available materials. These complexities often confound search engines, hindering precise web page classification. This research addresses these challenges with a streamlined approach to effective online Arabic web page classification. Our strategy involves two key components: First, we employ a comprehensive text mining approach to extract valuable insights from a corpus of documents. This begins with text extraction from the online web page after providing the URL and the removal of stop words to enhance data quality. Additionally, we leverage Natural Language Processing (NLP), specifically lemmatization, to normalize text and reduce linguistic variations, ensuring consistent and meaningful representation. To complete the text mining process, we use the Bag of Words (BoW) model to transform preprocessed text data into a numerical format, capturing word frequencies. Second, we harness the power of the Artificial Bee Colony (ABC) optimization algorithm, inspired by bees' foraging behavior, as a pivotal element in our decision-making process. This algorithm provides a robust framework for optimizing classification tasks. Empirical results affirm the effectiveness of our approach, achieving an impressive 95.349% accuracy rate. This advancement bridges the gap between the intricacies of the Arabic language and efficient web content organization, promising a more informative Arabic web landscape.

Keywords: Text Mining, NLP, Arabic Language, Web Page Classification, Artificial Bee Colony, Artificial Intelligence.

1 Introduction

The internet's rapid growth has made web page classification crucial for managing, retrieving, and integrating information, as well as indexing, topic-specific crawling, extraction, advertisement removal, filtering, and parental control systems [1].

The Arabic language itself complicates the classification process because of its intricate orthography and rich morphology [2]. Highlight feature vectors' span increases, making it crucial to choose highlights strategically to avoid insignificant data and different meanings. [3]. Many benchmarking corpora can classify Latin, Japanese, and Chinese texts, unlike Arabic texts. [4] [5]. Arabic vowels (waw, yaa, and alf) and consonant letters require a unique morphology and grammar system, while other letters are consonants [6]. Arabic's vast vocabulary and concepts set it apart. [7]. The Arabic language has 28 letters, in addition to

*Email: hamza_95260@svuonline.org

the Arabic hamza (ء), which is considered a letter by some Arabic linguists, and it is written from right to left. It has two genders: feminine and masculine, and has numerical numbers in singular, dual, and plural forms. Grammatically, there are three cases: nominative, accusative, and genitive. Nouns have three linguistic cases: nominative when subject, accusative when verb object, and genitive when preposition object [8], [9]. Arabic uses diacritics to represent small vowel letters, without upper or lower case, and includes (fatha, kasra, damma, sukun, shadda, and tanween) [7].

So it is an inflectional and highly derived language. For example, the word (سياج) has two roots: (ساج) and (سيج). Moreover, some words have different connotations depending on their context in the text. The word (messenger) may refer to the prophet of God, messenger, envoy, or others. There is also a problem that sometimes occurs in determining the origin of the word, whether it is a verb or a noun, so the word (يسير) (is walking) may refer to the present tense of the verb (walk) or it may be an adjective indicating (easy) [10].

In light of these linguistic intricacies, this research presents a novel approach to online Arabic web page classification. It introduces a streamlined workflow that capitalizes on text mining techniques and leverages the Artificial Bee Colony (ABC) optimization algorithm to navigate the challenges inherent in Arabic language processing. Our contribution lies in providing an efficient and effective model for online web page classification, which is considered an NLP task [11] tailored specifically to the Arabic language. In Section 3, we talk about the ABC optimization algorithm, detail our model's workflow, dataset preparation, training phase, and the role of the ABC algorithm. In Section 4, we will conduct a comprehensive result analysis and compare our findings with existing research in the field. In Section 5, we will suggest some prospects for developing our system.

2 Related Works

Boukil and others [12] introduce an innovative method for Arabic text classification using Arabic stemming, the term frequency-inverse document frequency technique, and convolutional neural networks. They extract, select, and reduce features using Arabic stemming; they use the term frequency-inverse document frequency technique as a feature weighting technique; and they use deep learning algorithms like convolutional neural networks for classification. This combination, combined with hyperparameter tuning, achieves excellent results on multiple benchmarks.

To help Arabic language researchers choose the right dataset for their studies, Ababneh [13] used seven Arabic datasets and well-known and accurate learning models, such as naive Bayes, random forest, K-nearest neighbor, support vector machines, and logistic regression. The datasets were the Single-Label Arabic News Articles Dataset (SANAD), Khaleej, Arabiya, Akhbarona, KALIMAT, Waten2004, and Khaleej2004. The analysis of the relevance and time scores shows that training the support vector machine model on Khaleej and Arabiya obtained the most significant results in the shortest amount of time.

Another study [14] proposes a feature selection method based on a combination of chi-square and Artificial Bee Colony (ABC). Chi-square, a filter method that is computationally fast, simple, and can deal with a large-dimensional feature, is used as the first level of the feature selection process. After that, the wrapper method, the Artificial Bee Colony algorithm, is used as the second level, where Naive Base is used as a fitness function. The results showed that a reduced number of features outperformed classification accuracy using the original feature set. Furthermore, the proposed method had better performance compared with the chi-square method and the ABC algorithm as a feature selection method.

In another paper [15], the authors evaluate Arabic short text classification using three standard Naïve Bayes classifiers: multinomial Naïve Bayes, completed Naïve Bayes, and Gaussian Naïve Bayes. In their method, they classify the theses and dissertations using their titles to perform the classification process. Their method classifies the document based on its titles and places it in the desired specialization. Several preprocessing techniques have been applied, such as punctuation removal, stop word removal, and space vectorization. For feature extraction, they adopt the TF-IDF method. The study results showed that the completed Naïve Bayes classifier proposed the best performance.

3 Materials and Methods

3.1 Artificial Bee Colony (ABC) Optimization algorithm

The ABC algorithm, which is inspired by the behaviors of honey bee colonies, is a swarm intelligence algorithm that allows you to find optimal solutions for various problems [16]. The search space of the ABC algorithm is represented as a honey bee colony's environment, and each point in the search space matches a potential food source. Each food source contains a different amount of nectar, and those nectars represent the fitness value of the food source. The ABC algorithm works with three kinds of bees: onlookers, employed bees, and scout bees [16], [17]. In this algorithm, a food source is only designed for an employed bee. Employed bees calculate the amount of nectar. The colony has the same number of employed and onlooker artificial bees. Employed bees are responsible for gathering the required quality information and transferring it to onlooker bees. Therefore, the onlooker bees will be directed to rich food sources in terms of the amount of nectar. These procedures proceed until the limit value is reached [18]. In the literature, many ABC algorithms have been developed for various applications [19]. To solve global optimization problems, many improved ABC variants have been developed [20].

Figure 1 shows the general flow chart of the ABC algorithm:

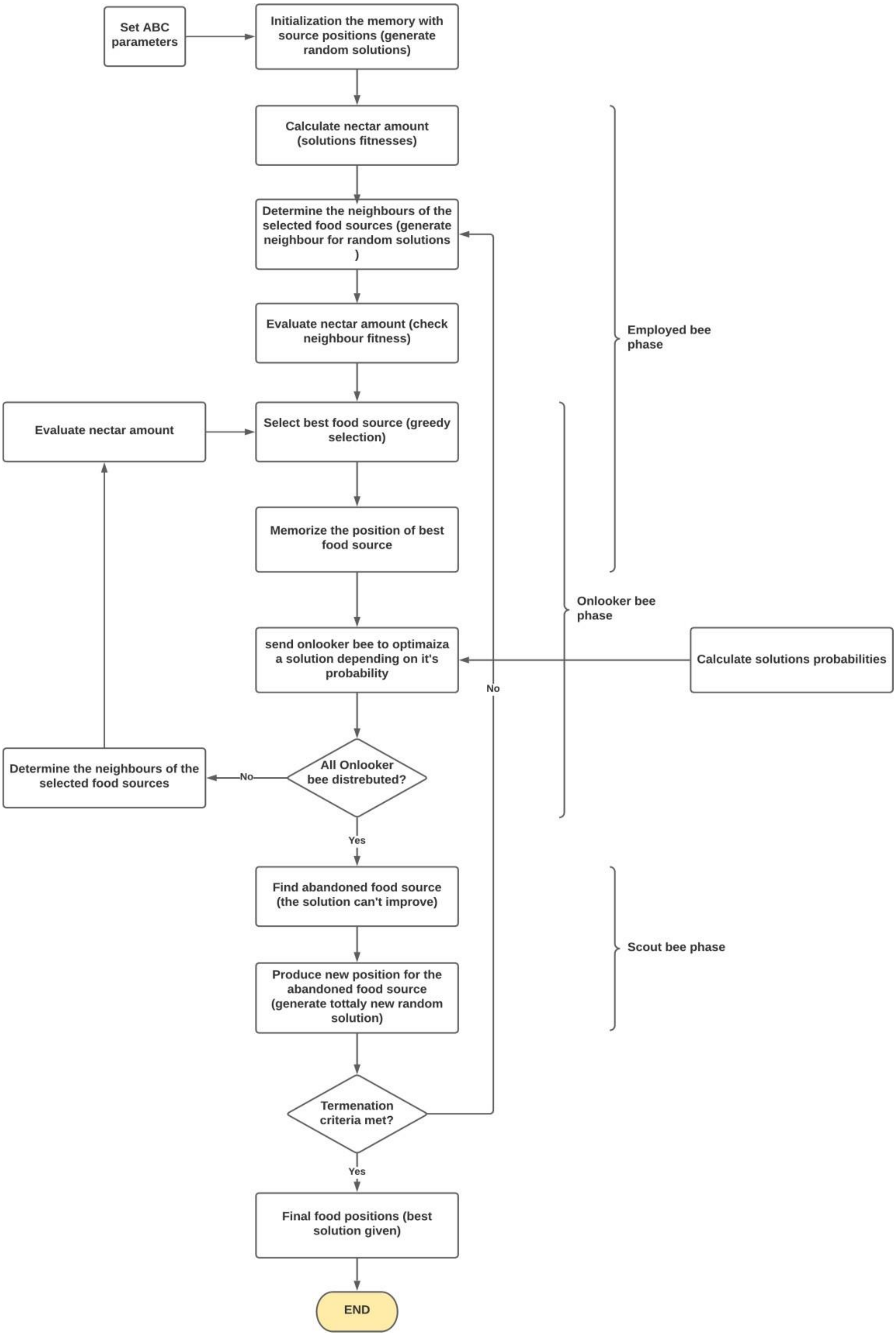

Set ABC parameters
Initialization the memory with source positions (generate random solutions)
Calculate nectar amount (solutions fitnesses)
Determine the neighbours of the selected food sources (generate neighbour for random solutions)
Evaluate nectar amount (check neighbour fitness)
Employed bee phase
Evaluate nectar amount
Select best food source (greedy selection)
Memorize the position of best food source
Onlooker bee phase
send onlooker bee to optimaiza a solution depending on it's probability
Calculate solutions probabilities
No
Determine the neighbours of the selected food sources
No
All Onlooker bee distrebuted?
Yes
Find abandoned food source (the solution can't improve)
Produce new position for the abandoned food source (generate tottaly new random solution)
Scout bee phase
Termenation criteria met?
Yes
Final food positions (best solution given)
END

Figure 1: General flow chart of the ABC algorithm

3.2 Proposed model

The work plan is to find a presorted dataset, text-process it, and then use this set to train the system. The Artificial Bee Algorithm is then put to the test by being applied to a random online link to see the outcome, followed by an unclassified dataset to measure the algorithm's accuracy.

3.3 Proposed model work details:

The software that implements our system will be programmed using Java with NetBeans IDE 8.0.1.

3.3.1 Dataset Processing:

A suitable dataset has been adopted and has received a positive evaluation by researchers, a dataset called SANAD: Single-Label Arabic News Articles Dataset for Automatic Text Categorization [21].

This dataset is available in the form of independent text files distributed in seven volumes, each bearing the name of a specific classification, and each file contains a news article whose classification is its parent folder name. Moreover, the number of these files is more than 75,000, divided unequally into 7 categories: culture, business, politics, sports, health, technology, and local, and this does not fit the process of entering and exiting the algorithm to read each file separately, which will consume significant time and resources for a computer. All these files had to be combined into one XLSX file so that each text file would be a record. Therefore, 1000 files were randomly selected from each of the following five categories: business, politics, sports, health, and technology, i.e., 5000 files in total; the other categories were dissolved since we found it very difficult to deal with the entire dataset due to the computer's limited capabilities. We make an XLSX file from them to use as a training dataset file.

3.3.2 Linguistic Processing of the Dataset:

The datasets are in an unsuitable format for training the system because they contain a large number of insignificant terms in the classification process, or they are in synonymous forms or different time formulae, which allow the algorithm to be misled later in determining the proper classification. So to preprocess and represent the dataset effectively, we will employ several techniques, including lemmatization, stop word removal, and feature extraction.

a) Lemmatization

For the feature extraction method, we perform the classification task based on the number of times the words are repeated in the text, so it is necessary that the words that have the same meaning be unified into one word. For example, to clarify the idea, the following words (لغتنا, لغتهم, لغوي, اللغة, اللغات) have the same origin, which is (لغة), so we cannot consider them as separate words when calculating their frequency because they have the same meaning and the same origin.

This method is known as word origin or lemmatization. It is the process of assembling the inflected parts of a word such that they can be recognized as a single element, called the word's lemma or its vocabulary form. It is an important component in any natural language processing (NLP) system. It serves as a fundamental preprocessing step for the majority of applications dependent on the comprehension of natural language [22]. This process is the same as stemming, but it adds meaning to particular words. In simple words, it connects text with similar meanings to a single word [23]. So it is the process of returning words to their origin based on the context of the text rather than in an abstract manner, as the process of stemming does, which is ineffective for our job.

An Arabic word can be made up of clitics (proclitics and enclitics), lemma, suffixes, and prefixes, as seen in Figure 2. Arabic words can be reduced to a variety of forms, including:

- Root: it is composed of three or four letters, and it cannot support any other decomposition.
- Stem: it is that part of a word to which grammatical prefixes and suffixes are added.
- Light stem: it is a light version of the stem in which more suffixes and prefixes are eliminated.
- Lemma: it represents the minimal form of a word that bears its principal meaning, and it represents dictionary entries [24].

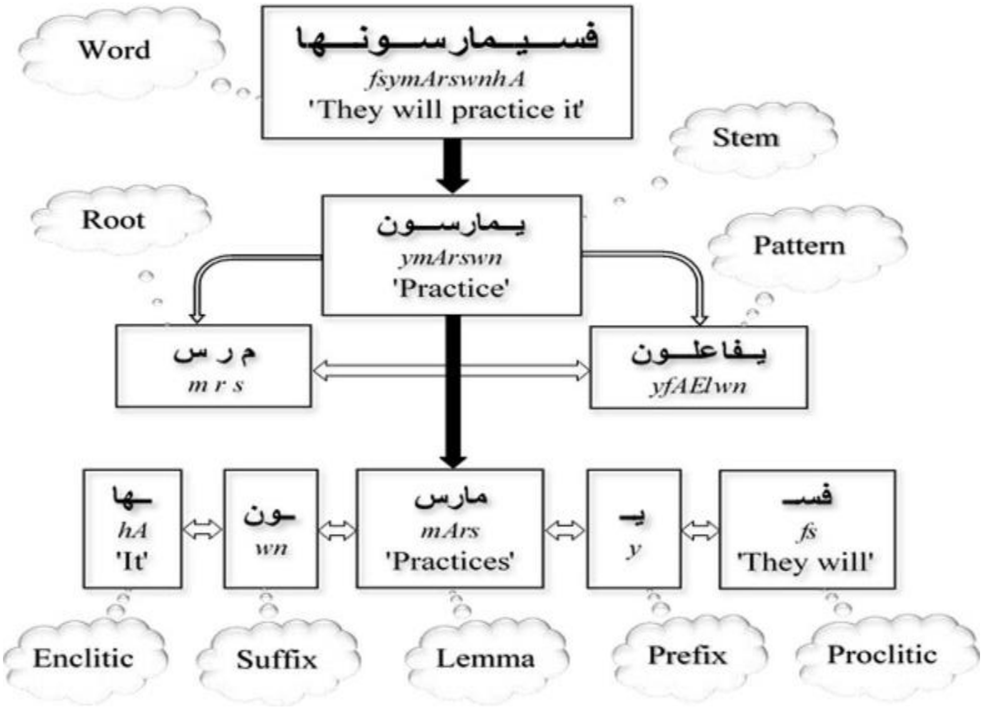


Figure 2: Example of an Arabic word morphology analyzed [25]

Let’s take the previous word (فسيمارسونها), which means (they will practice it), and the word (فستمارسونها), which means (you—refers to plural—will practice it). These two words have two steams (تمارسون، يمارسون) according to Figure 2, which will produce two features with (weight = 1) for each if we use steaming in the feature extraction process, but when we use lemmatization, these two words will produce one word (مارس) with (weight = 2), making lemmatization more efficient for extraction keywords for classification purposes. We will give an example of the lemmatization process in Arabic [26]:

Original text:

(يُشار إلى أن اللغة العربية يتحدثها أكثر من 422 مليون نسمة ويتوزع متحدثوها في المنطقة المعروفة باسم الوطن العربي بالإضافة إلى العديد من المناطق الأخرى المجاورة مثل الأهواز وتركيا وتشاد والسنغال وإريتريا وغيرها. وهي اللغة الرابعة من لغات منظمة الأمم المتحدة الرسمية الست).

The text after the lemmatization procedure is:

(أشار إلى أن لغة عربي تحدث أكثر من 422 مليون نسمة توزع متحدثوها في منطقة معروف اسم وطن عربي إضافة إلى عديد من منطقة آخر مجاور مثل أهواز تركيا تشاد سنغال أريتريا غير . هي لغة رابع من لغة منظمة أمة متحد رسمي ست).

To do lemmatization on our dataset, we’ll use the FARASA library [27], which was specifically developed to cope with the Arabic language and is being used for the first time to classify web pages.

b) Stop Word Removal

The language contains regularly repeated terms that, while necessary for understanding the text, do not determine the basic identity of the text or direct its classification. These words in Arabic include prepositions (من, إلى, عن, etc.), structure words (سوف, ممكن, لو, لكن, etc.), time and place adverbs, and many other words that are not considered keywords. These are known as “stop words” [28].

Stop words play a pivotal role in Natural Language Processing (NLP) techniques for information retrieval. A common preprocessing task when working with text data involves eliminating these stop words [11].

It is vital to eliminate stop words from the text before deciding to categorize it since they don’t contribute to any meaningful interpretation and their frequency is also high, which may affect the computation time [29].

Furthermore, we will delete punctuation marks because they interfere with proper categorization.

To do this, the text's words will be compared against a pre-prepared file [30] containing the Arabic language's famed stop words, and if they are found, they will be eliminated.

3.3.3 Training: Feature Extraction phase

The system will be trained on the dataset during this phase. Following the completion of the linguistic processing operations on the dataset, all articles in one category will be read, and the number of times each word is repeated will be recorded separately. As an output, each category has its own XLSX file.

So in this phase, we will extract features from the data. Feature selection is a data preprocessing procedure that opts for a subset of input variables while discarding those features that offer minimal or no predictive information [31]. This procedure is considered a text mining technique, taking into consideration that text mining delves into unstructured or partially structured text documents to identify meaningful patterns and rules, revealing trends and important characteristics related to specific subjects [31].

The POI library [32] was utilized to deal with XLSX files at this phase, which is a robust library that allows users to easily read, write, and change those files.

So, initially, the training dataset file will be retrieved, and the categories contained within it will be extracted and added to the categories list.

The file will then be read again, and all cells in the first column (articles) that correspond to the first item in the category mentioned in the adjacent cell will be read.

As a result, we extract all articles whose classification is the first item, for example, sports. Now we have a string that combines all of the texts in the sports category.

Then, each word in this string is read and counted, and the word and number of occurrences are stored in Map<String, Integer>, which represents a bag of words, and this bag of words is then unloaded into a classification XLSX file. The process described above will be performed for each item on the category list. That is, we will receive many files, corresponding to the number of distinct categories in the training dataset. For example, the file (sport.xlsx) will contain all words from the training dataset's articles classified by sport. Each term will be recorded and matched by the number of times it appears in the articles categorized in the sport as a whole, and the training will be completed.

To return to the Bag of Words (BoW) model, it is a simplified representation used in NLP and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity [33]. We are not concerned with the sequence of the word's occurrence in the text; rather, we are concerned with the number of times it is repeated because this reflects its impact on understanding the classification of the text in which it is stated. To demonstrate the concept, let’s have the following example text:

(سارة تحب مشاهدة الأفلام، أحمد يحب مشاهدة الأفلام أيضاً. كما أن سارة تحب مشاهدة مباريات كرة القدم)

After lemmatization and deleting the stop words, the bag of words for this text will be:

{سار:2 , أحب:3 , مشاهدة:3 , فيلم:2 , أحمد:1 , مباراة:1 , كرة:1 , قدم:1}

As we can see, each word referenced in the text will be placed in the bag, along with a number showing how many times it appears in the text.

Nonreported words with a frequency of one will not be stored in our system's HashMap because they are repeated once; hence, they will not affect the decision-making process. They are not considered influential keywords.

3.3.4 Make the decision: ABC algorithm phase

The ABC algorithm will determine if the web page belongs to one categorization or another during this step, as we will see in more detail below.

When you enter a web page link into the system, it will establish a connection, access that page, read it, and the text of the news article will be retrieved from the website since the article text defines if the page belongs to a particular categorization or not.

The JSOUP-1.14.1 library [34] was used for connecting and extracting, allowing the extraction of any part of a web page based on a given tag. The article will be extracted and saved as an XLSX file.

To broaden the system's capacity to cope with a web page link or a document including many texts, we will save the extracted text of the article as a string variable in an XLSX file rather than in RAM.

Now that we have a file containing unclassified text (in the case of URL classification or several texts in the case of a document containing multiple texts), we will access it, extract the text using the Apache POI-3.17 library, perform NLP processes on it, and represent it in the bag of words form.

This bag of words will be sent to the ABC algorithm, which will classify it.

Figure 3 depicts the flow chart of the process of classifying web pages based on their links:

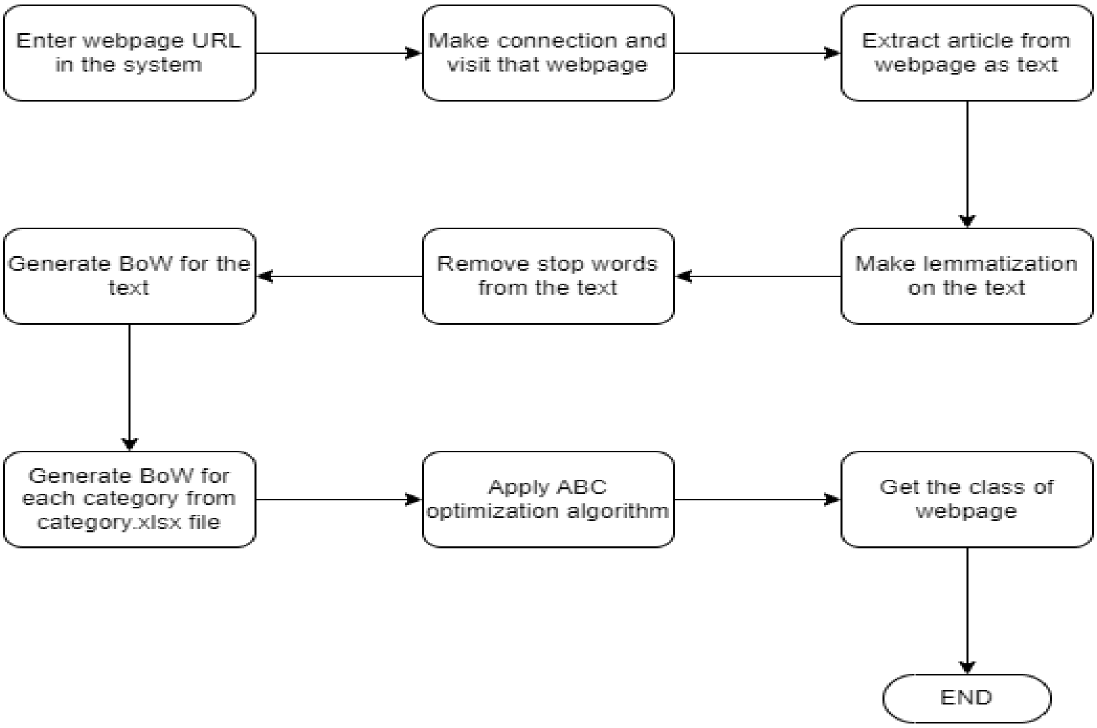

Enter webpage URL in the system
Make connection and visit that webpage
Extract article from webpage as text
Make lemmatization on the text
Remove stop words from the text
Generate BoW for the text
Generate BoW for each category from category.xlsx file
Apply ABC optimization algorithm
Get the class of webpage
END

Figure 3: Proposed system's flowchart

3.4 Applying the ABC optimization algorithm

We can simplify the ABC algorithm's work by saying that it will first generate a set of primitive solutions, then work on improving them, and finally choose the best one.

First, the ABC algorithm's control parameters must be set, which are:

a) NP: The number of bees in the hive is equal to the number of employed bees and onlooker bees combined.
b) Limit: The solution that cannot be improved after the limit attempt will be replaced (a food source that could not be improved through limit trials is abandoned by its employed bee).
c) Max cycle: stopping condition (the number of cycles for foraging).
d) Food number: The number of solutions generated at the start of the algorithm. It will be set automatically because it is equal to half the population of the cell, i.e., food number = NP/2.

The parameters of the classification problem are:

a) Run times: The number of times the algorithm is run as a whole to calculate the algorithm's accuracy.
b) D: The number of texts to be categorized; this parameter will be set automatically; it will be 1 in the case of a web page and D = n in the case of a document containing n articles (text).

3.4.1 Random propagation of bees (initiation phase):

After setting the necessary parameters, the algorithm will start the initialization phase. As previously stated, we have D texts to be classified. During this phase, each text will be assigned to a random category from the categories extracted from the training dataset; this will be the initial solution, and its fitness will be calculated using Eq. (1).

$$\text{fitness} = \sum_{j=1}^{j=D} \sum_{i=1}^{i=m} \mathrm{x_{ij}} * \mathrm{x_{ik}} \qquad (1)$$

where D is the number of texts to be classified, m is the number of words in the text, $\mathrm{x_{ij}}$ is an integer value of the occurrence of the word i in the text j to be classified, so:

$$i \in [1, m\,] \quad , \quad j \in [1, D]$$

and $\mathrm{x_{ik}}$ is an integer value of the occurrence of the word i in the (k.xlsx) file that was generated after the training phase, so $k \in [1, number\ of\ categories\ to\ choose\ from]$, in our case $k \in [1, 5]$.

3.4.2 Memorizing the best food position (save the best solution):

The best solution and its fitness value will be saved after initialization.

3.4.3 Employed bees phase:

During this phase, the system will generate a random category and assign it to one of the random texts, and then calculate the fitness of the produced solution after this change. If the fitness value is less than the prior solution, the old value is kept, and the trial counter is increased by one.

3.4.4 Calculating the probability of choosing a solution:

After that, the system calculates the value of the probability function for each solution; these values will be used in the next phase. The probability for solution (i) will be calculated using Eq. (2).

$$prob[i] = fitness[i] \,/\, maxfit \qquad (2)$$

Where fitness[i] is the fitness value of the solution (i), maxfit is the maximum fitness value among all solutions.

3.4.5 Onlooker bees phase:

After calculating probabilities for each solution, the system will generate a random number smaller than one and then cycle through all solutions. If the solution has a probability greater than this random number, it will change this solution (generate a nearby solution), then calculate a new fitness value, compare it to the old one, and memorize the best solution, as mentioned in the employed bees phase.

3.4.6 Scout bees phase:

If one of the solutions is not fit, i.e., the value of its trial variable has reached a predefined limit, a completely different solution will be randomly generated and replaced, i.e., we will create a new configuration for this solution, which naturally corresponds to a bee flying in a random direction to determine the position of different food, in this case, called a scout bee.

3.4.7 Repeat flights:

The operation will be repeated multiple times from the employed bee phase to the scout bee phase, up to the predefined (maxCycle) value.

3.4.8 Output:

For each run, the solution with the highest fitness value will be chosen, and the accuracy of this solution will be measured by comparing it to the true solution; the average accuracy will be calculated across all runs.

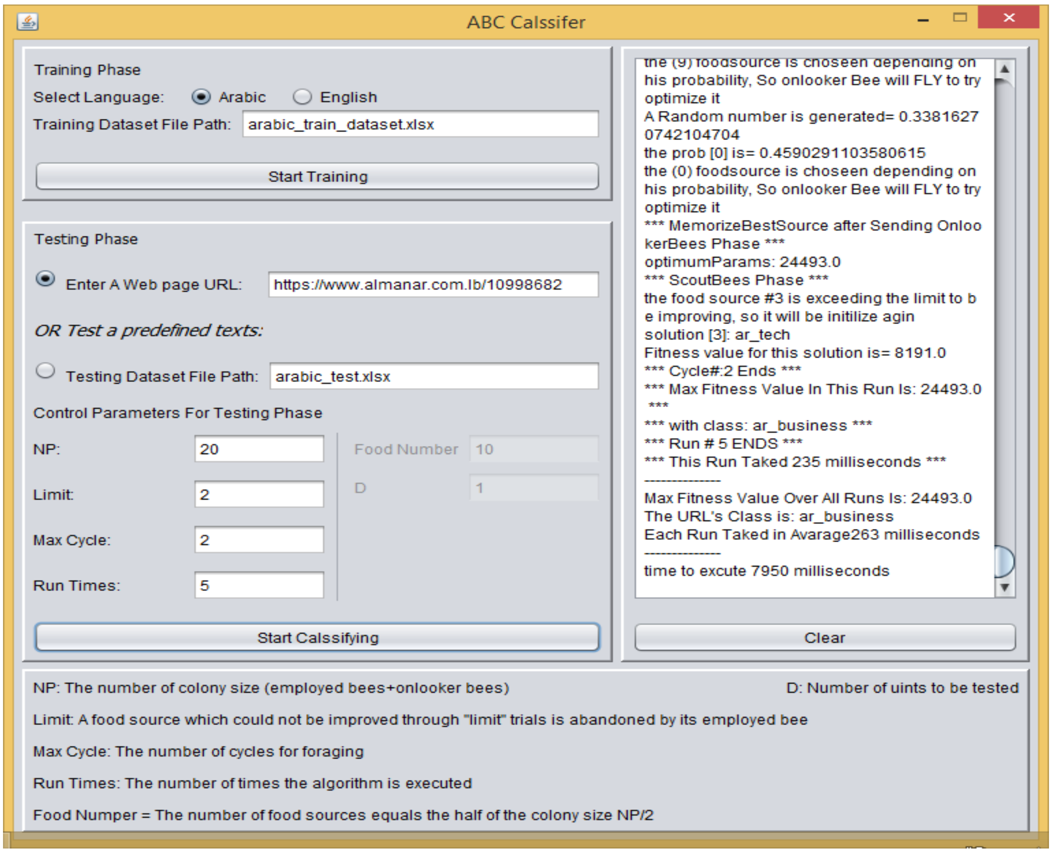
4 Results and Discussion

During the testing phase, we will examine some web pages and their classes. Table 1 displays system outputs for a sample of web pages with execution times obtained from well-known news organizations.

Table 1: The system outputs for a sample of web pages with execution time

#	Web page	Run Times	ABC algorithm parameters			output	time to execute (s)
			Max Cycle	Limit	Np		
1	https://aljazeera.net/health/2023/5/6/رغم-إعلان-الصحة-العالمية-إنهاء	5	2	2	20	health	8.583
2	https://arabic.euronews.com/2023/10/09/iaea-energy-production-rise-climate-change-emissions-raphael-grossi-security	5	2	2	20	business	7.411
3	https://www.bbc.com/arabic/interactivity-67048677	5	2	2	20	politics	9.474
4	https://arabic.cnn.com/sport/article/2023/06/11/erdogan-mohammad-bin-zayed-hug-social-inter-city-football	5	2	2	20	sport	7.829
5	https://cnnbusinessarabic.com/technology/27420/توتير-وغوغل	5	2	2	20	tech	8.300
6	https://www.albawaba.com/ar/صحتكِ-وجماللكِ/سرطان-الثدي-خلال-فترة-الحمل-1449995	5	2	2	20	health	7.382
7	https://www.alhurra.com/usa/2023/10/15/السفارة-الأميركية-في-إسرائيل-تعرض-إجلاء-أميركيين-بحرا-حيفا	5	2	2	20	politics	6.969
8	https://arabic.rt.com/sport/1503926-بيع-أغلى-تذكرة-في-تاريخ-الدوري-السعودي-فمن-هو-صاحب-التذكرة-الذهبية-ولأي-فريق-وما-ثمنها/	5	2	2	20	sport	9.841
9	https://www.almanar.com.lb/10998682	5	2	2	20	business	7.950

And here is a screenshot of the system showing the output of one of the samples in Table 1.



Training Phase

Select Language: ● Arabic ○ English

Training Dataset File Path: arabic_train_dataset.xlsx

Testing Phase

● Enter A Web page URL: https://www.almanar.com.lb/10998682

OR Test a predefined texts:

○ Testing Dataset File Path: arabic_test.xlsx

Control Parameters For Testing Phase

NP: 20

Limit: 2

Max Cycle: 2

Run Times: 5

Food Number 10

D 1

```
the (9) foodsource is choseen depending on
his probability, So onlooker Bee will FLY to try
optimize it
A Random number is generated= 0.3381627
0742104704
the prob [0] is= 0.4590291103580615
the (0) foodsource is choseen depending on
his probability, So onlooker Bee will FLY to try
optimize it
*** MemorizeBestSource after Sending Onloo
kerBees Phase ***
optimumParams: 24493.0
*** ScoutBees Phase ***
the food source #3 is exceeding the limit to b
e improving, so it will be initilize agin
solution [3]: ar_tech
Fitness value for this solution is= 8191.0
*** Cycle#:2 Ends ***
*** Max Fitness Value In This Run Is: 24493.0
 ***
*** with class: ar_business ***
*** Run # 5 ENDS ***
*** This Run Taked 235 milliseconds ***
--------------
Max Fitness Value Over All Runs Is: 24493.0
The URL's Class is: ar_business
Each Run Taked in Avarage263 milliseconds
--------------
time to excute 7950 milliseconds
```

NP: The number of colony size (employed bees+onlooker bees)

D: Number of uints to be tested

Limit: A food source which could not be improved through "limit" trials is abandoned by its employed bee

Max Cycle: The number of cycles for foraging

Run Times: The number of times the algorithm is executed

Food Numper = The number of food sources equals the half of the colony size NP/2

Figure 4: Sample output from Table 1 showing the system's performance

To assess classification accuracy, we will classify a file containing previously extracted texts from web pages, using different control parameters each time and running the algorithm several times to obtain average accuracy, yielding the results shown in Table 2:

Table 2: Accuracy of classifying multiple texts

Accuracy	The number of texts in the document	The number of times the algorithm is repeated (Run Times)	The values of the control parameters of the bee algorithm		
			Max Cycle	Limit	Np
55.814%	86	5	100	50	10
61.628%	86	5	200	100	10
92.79%	86	5	200	100	50
95.232%	86	5	400	200	50
95.349%	86	5	400	200	100
95.349%	86	5	600	400	100

These results show that increasing the control coefficients of the ABC algorithm results in higher accuracy each time until we reach certain limits for these coefficients, at which point the increase in value no longer improves the accuracy of the results but instead causes the execution time to increase and resources to be consumed without benefit. The accuracy we got is 95.349%, which is higher than the accuracy of 92.94% found in [12]. That study used a combination of the convolutional neural network (CNN), a well-known deep learning algorithm used in image processing and pattern recognition, and the term frequency-inverse document frequency (TF-IDF), one of the most interesting vector-word presentation techniques.

The authors of [13] used accurate and well-known learning models like Naive Bayes (NB), random forest, K-nearest neighbor, support vector machines (SVM), and logistic regression models to sort Arabic text into seven new TC datasets. We used SANAD in our research and found that these models were 56% to 68% accurate in determining relevance. However, the SVM model with Khaleej and Arabiya had the maximum accuracy score of 82%, implying that employing the ABC algorithm with the SANAD dataset in our research will provide a more accurate classification.

The authors of [14] employed the chi-square method as the first level of feature selection to improve the performance of the ABC algorithm at the second level. For classification, three algorithms were used as classifiers: Naive Bayes, J48, and SVM. The weighted F1-measure is used as a classification assessment metric, and the NB algorithm achieved a weighted value of 77.72. To compare their results to ours, we will compute a weighted F1-measure to evaluate our classification using the following Eqs. (3) and (4):

$$\text{F1 Measure} = 2.\frac{\text{p. r}}{\text{p} + \text{r}} \qquad (3)$$

$$\text{Weighted F1 Measure} = \frac{\sum_{i=0}^{n} F1 \ \text{measure (Ci)} \ * \ \text{Ni}}{\text{T}} \qquad (4)$$

where T is the total number of documents.
And we achieved a $\text{weighted F1 Measure} = 95.26$ in our research, which was a superior score.

Another study [15] used three standard Nave Bayes classifiers, Multinomial Nave Bayes (MNB), Complemented Nave Bayes (CNB), and Gaussian Nave Bayes (GNB), to classify Arabic short text, and using TF-IDF as a feature extraction method, the result showed that (CNB) gave the best accuracy with 84%.
Table 3 compares the results acquired from the ABC algorithm in this research to those obtained from other algorithms in other studies.

Table 3: Comparison of the results gained with other research

Research	highest obtained scores
our research	accuracy = 95.349%, weighted F1 measure = 95.26
[12]	accuracy = 92.94%
[13]	accuracy = 82%
[14]	weighted F1 measure = 77.72
[15]	accuracy = 84%

The failure of our research to achieve higher classification accuracy can be related to two key reasons:
1) Some web pages may be classified into many categories if an article or news item discusses overlapping issues. For example, the web page (number 5) in Table 1 discusses financial concerns between two technology businesses, Twitter and Google. This web page confuses humans when categorizing it: does it fit under (technical) or (business)? That is, if we just had to make one technical or business decision, our system would define it as technical, but the news agency would classify it as business. This is the pitfall into which our system fell! When we reviewed the erroneous results predicted by our algorithm in Table 2, we found it difficult to classify them under only one of the classes.
2) The training dataset is not large; increasing the size of the samples necessarily means

increasing the number of unique keywords distinguished for each classification and, of course, increasing the number of repetitions. This raises the value of the efficiency function (food quality) that we discussed earlier, bringing the optimal solution closer to what our system expects. The number of samples in the SANAD dataset utilized for training was greater than 75,000; however, the limitations of the available capabilities forced us to limit ourselves to 5000 samples only.

In terms of implementation time, Table 4 illustrates how long the system took during the training and testing phases:

Table 4: Time that the system took in the training and testing process

Time (s)	The number of texts in the document	The number of times the algorithm is repeated (Run Times)	The values of the control parameters of the ABC algorithm		
			Max Cycle	Limit	Np
Training time = 1028.4	5000	ABC is not applied in training			
Testing time = 8.193*	URL	5	2	2	20
Testing time = 323.94	86	1	200	100	50
Testing time = 628.44	86	5	400	200	100

*The arithmetic mean of the classification time for the nine web pages shown in Table 1.

There are two key reasons for the length of training and testing:

1) The time complexity of the mechanism used in Natural Language Processing (NLP) that is applied to the dataset in the process of training and experimentation, then in the process of extracting features (keywords), calculating the number of their repetitions, and storing the results in files on the operating system, as it is well known, operations on files, such as writing and reading (which occur on the hard disc), require a long time as compared to operations on variables (which occur in random memory). As a result, the actions in the files are blamed for the lengthy execution time (it takes over 90% of the overall execution time, according to the experiments we conducted). As previously stated, we saved the training results in files with permanent memory so that we did not have to repeat the training procedure every time we needed to run a test.

2) The available resource specifications of the computer on which we are running the system are as follows:

Computer brand: Acer notebook
Processor and Speed: Intel (R) CoreTM i7-2670QM CPU @ 2.20GHz
RAM: 6.00 GB
Operating system: Windows 10 Pro

5 Prospects

Based on this research, the problems we encountered, and the other research we read, we can conclude that: It is possible to develop our system by having it take several classifications at the same time as one, such as (technology-business), which can be accomplished by using a dataset whose samples are classified in this manner or by designing a system capable of deciding to classify the sample into two categories if these two occur. The two classes have close efficiency function values.

It can also be developed to include sub-categories, such as sports (football). This can be accomplished by manually reclassifying each sample of the training dataset in this manner or by retraining the features taken from the training data under each category in additional, more detailed training data.

6 Conclusions

This research presents a streamlined approach to effective online web page classification. The strategy used comprises two key components that offer a robust solution for optimizing classification tasks.

The first component involves taking advantage of some NLP and text mining techniques, including text extraction, stop-word removal, lemmatization, and feature extraction, with the help of the FARASA library, which was used for the first time for web page classification purposes. These methods enhance data quality and provide a consistent and meaningful representation of text content. Utilizing the Bag of Words (BoW) model further allows for transforming preprocessed text data into a numerical format, effectively capturing word frequencies.

The second critical element of the strategy is the incorporation of the Artificial Bee Colony (ABC) optimization algorithm. Inspired by the foraging behavior of bees, this algorithm provides a strong framework for optimizing classification tasks.
Empirical results have affirmed the effectiveness of this approach, with an impressive accuracy rate of 95.349%. This achievement highlights the potential applicability of our method to diverse domains and languages, promising a more informative web content organization.

Our research contributes to the broader field of online information retrieval, offering a valuable solution that transcends the complexities of specific languages or challenges. By bridging the gap between text mining techniques and optimization algorithms, our approach sets the stage for further advancements in web content classification across various domains.

Data Availability

The data used to support the findings of this research are available upon request from the author.

References

[1] M. Hashemi, "Web page classification: a survey of perspectives, gaps, and future directions," *Multimedia Tools and Applications*, vol. 79, no. 17-18, pp. 11921–11945, 2020.

[2] M. Abdeen, A. Elmahalawy, S. Albouq and S. Shehata, "A Closer Look at Arabic Text Classification," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 10, pp. 677– 688, 2019.

[3] Y. Caballero, R. Bello, D. Alvarez and M. M. Garcia , "Two new feature selection algorithms with Rough Sets Theory," in *IFIP International Federation for Information Processing*, Boston, Springer, pp. 209–216, 2006.

[4] A. M. El-Halees, "Arabic Text Classification Using Maximum Entropy," *The Islamic University Journal (Series of Natural Studies and Engineering)*, vol. 15, no. 1, pp. 157-167, 2007.

[5] M. . S. Khorsheed and A. O. Al-Thubaity , "Comparative evaluation of text classification techniques using a large diverse Arabic dataset," *Language Resources and Evaluation*, vol. 47, no. 1, pp. 513–538, 2013.

[6] M. A. Ahmed, R. A. Hasan, A. H. Ali and M. A. Mohammed, "The classification of the modern Arabic poetry using machine learning," *TELKOMNIKA*, vol. 17, no. 5, pp. 2667-2674, 2019.

[7] A. M. F. Al Sbou, "A Survey of Arabic Text Classification Models," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 4352 - 4355, 2018.

[8] A. Mesleh, "Support Vector Machines Based Arabic Language Text Classification System: Feature Selection Comparative Study," in *Advances in Computer and Information Sciences and Engineering*, Dordrecht, Springer, pp. 11-16, 2008.

[9] M. M. Syiam, Z. T. Fayed and M. B. H. Morgan, "AN INTELLIGENT SYSTEM FOR ARABIC TEXT CATEGORIZATION," *International Journal of Intelligent Computing and Information Sciences*, vol. 6, no. 1, pp. 1-19, 2006.

[10] A. T. Al-Taani and N. A. K. Al-Awad, "An Empirical Analysis of Arabic WebPages Classification using Fuzzy Operators," *International Journal of Computational Intelligence,* vol. 5, no. 1, pp. 853-858, 2009.

[11] C. Emezue et al., "The African Stopwords Project: Curating Stopwords for African Languages," in *African NLP Workshop*, Virtual, 2022.

[12] S. Boukil, M. Biniz, F. El Adnani, L. Cherrat and A. E. El Moutaouakkil, "Arabic Text Classification Using Deep Learning Technics," *International Journal of Grid and Distributed Computing,* vol. 11, no. 9, pp. 103-114, 2018.

[13] A. H. Ababneh, "Investigating the relevance of Arabic text classification datasets based on supervised learning," *Journal of Electronic Science and Technology,* vol. 20, no. 2, pp. 20-39, 2022.

[14] M. Hijazi, A. Zeki and A. R. Ismail, "Arabic Text Classification Using Hybrid Feature Selection Method Using Chi-Square Binary Artificial Bee Colony Algorithm," *International Journal of Mathematics and Computer Science,* vol. 16, pp. 213-228, Jan. 2021.

[15] M. A. Alhakeem, M. F. Ibrahim and N. A. Fadhil, "Evaluation of Naïve Bayes Classification in Arabic Short Text," *Al-Mustansiriyah Journal of Science,* vol. 32, no. 4, pp. 42-50, 2021.

[16] D. KARABOGA, "An idea based on honey bee swarm for numerical optimization, technical report -TR06," Erciyes University, Engineering Faculty, Kayseri/Türkiye, 2005.

[17] Y. Xiaohui , Y. Zhu, W. Zou and L. Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm," *Neurocomputing,* vol. 97, no. 1, pp. 241-250, 2012.

[18] K. Hanbay, "A new standard error based artificial bee colony algorithm and its applications in feature selection," *Journal of King Saud University - Computer and Information Sciences,* vol. 34, no. 7, pp. 4554-4567, 2022.

[19] I. Brajević and P. Stanimirović, "An improved chaotic firefly algorithm for global numerical optimization," *International Journal of Computational Intelligence Systems,* vol. 12, no. 1, pp. 131-148, 2018.

[20] X. Chu, F. Cai, D. Gao, L. Li, J. Cui, S. X. Xu and Q. Qin, "An artificial bee colony algorithm with adaptive heterogeneous competition for global optimization problems," *Applied Soft Computing,* vol. 93, no. 1, p. 1, 2020.

[21] O. Einea, A. Elnagar and R. Al-Debsi, "SANAD: Single-Label Arabic News Articles Dataset for Automatic Text Categorization," Mendeley data, 2 September 2019. [Online]. Available: https://data.mendeley.com/datasets/57zpx667y9/2. [Accessed 22 May 2023].

[22] A. A. Freihat, A. Mourad , G. Bella and F. Giunchiglia, "Towards an Optimal Solution to Lemmatization in Arabic," in *The 4th International Conference on Arabic Computational Linguistics (ACLing 2018),* Dubai, United Arab Emirates, 2018.

[23] D. Khyani, S. B S, N. N M and D. B M, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *Journal of University of Shanghai for Science and Technology,* vol. 22, no. 10, pp. 350-357, 2020.

[24] M. Naili, A. H. Chaibi and H. H. Ben Ghezala, "Comparative Study of Arabic Stemming Algorithms for Topic Identification," in *23rd International Conference on Knowledge-Based and Intelligent Information & Engineering,* Budapest, Hungary, 2019.

[25] M. Boudchiche, A. Mazroui, M. O. A. Ould Bebah, A. Lakhouaja and A. Boudlal, "AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer," *Journal of King Saud University - Computer and Information Sciences,* vol. 29, no. 2, pp. 141-146, 2017.

[26] K. Darwish, H. Mubarak, A. Abdelali, M. Eldesouki and Y. Samih, "Farasa," QCRI, 10 6 2023. [Online]. Available: https://farasa.qcri.org/. [Accessed 10 6 2023].

[27] A. Abdelali, K. Darwish, N. Durran and H. Mubarak, "Farasa: A Fast and Furious Segmenter for Arabic," in *NAACL,* San Diego, California, 2016.

[28] A. Anwar, G. I. Salama and M. Abdelhalim, "VIDEO CLASSIFICATION AND RETRIEVAL USING ARABIC CLOSED," in *The 6th International Conference on Information Technology*, Amman , Jordan, 2013.

[29] D. Khurana, A. Koli, K. Khatter and S. Singh , "Natural language processing: state of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 3713–3744, 2022.

[30] M. T. Alrefaie and T. BAZINE, "mohataher/arabic-stop-words: Largest list of Arabic stop words on Github. 27 "أكبر قائمة لمستبعدات الفهرسة العربية على جيت هاب," May 2016. [Online]. Available: https://github.com/mohataher/arabic-stop-words. [Accessed 29 5 2023].

[31] L. Makara, K. Ogada and D. Njagi, "Feature Selection Techniques and Classification Accuracy of Supervised Machine Learning in Text Mining," *Journal of Information Engineering and Applications,* vol. 9, no. 3, pp. 53-61, 2019.

[32] Apache, "Apache POI - the Java API for Microsoft Documents," Apache, 15 September 2017. [Online]. Available: https://poi.apache.org/index.html. [Accessed 22 May 2023].

[33] S. Deepu, R. Pethuru and R. S, "A Framework for Text Analytics using the Bag of Words (BoW) Model for Prediction," in *1st International Conference on Innovations in Computing & Networking (ICICN16)*, Karnataka, India, 2016.

[34] J. Hedley, "jsoup: Java HTML Parser," jsoup.org, 31 January 2010. [Online]. Available: https://github.com/jhy/jsoup. [Accessed 5 April 2023].

[35] H. Lane, C. Howard and H. Hapke, Natural Language Processing in Action, Shelter Island, NY, USA: Manning Publications Co, 2019.

[36] D. Gunning and S. Ghosh, Natural Language Processing Fundamentals, Birmingham, UK: Packt Publishing, 2019.

[37] J. Ricketts, D. Barry, W. Guo and J. Pelham, "A Scoping Literature Review of Natural Language Processing Application to Safety Occurrence Reports," *Safety,* vol. 9, no. 22, pp. 1-16, 2023.

[38] D. . J. Ladani and N. . P. Desai, "Stopword Identification and Removal Techniques on TC and IR applications: A Survey," in *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020.