Text document clustering using mayfly optimization algorithm with k-means technique

Ratnam Dodda, Alladi Suresh Babu
Department of Computer Science and Engineering, Jawaharlal Nehru Technological University, Ananthapur, India

Article Info

Article history:

Received Feb 27, 2024
Revised Apr 3, 2024
Accepted Apr 6, 2024

Keywords:

ADASYN
K-means algorithm
Machine learning
MOA
Text document clustering

ABSTRACT

Text clustering is a subfield of machine learning (ML) and natural language processing (NLP) that consists of grouping similar sentences or documents based on their content. However, insignificant features in the documents minimize the accuracy of information retrieval which makes it challenging for the clustering approach to efficiently cluster similar documents. In this research, the mayfly optimization algorithm (MOA) with a k-means approach is proposed for text document clustering (TDC) to effectively cluster similar documents. Initially, the data is obtained from Reuters-21678, 20-Newsgroup, and BBC sports datasets, and then pre-processing is established by stemming and stop word removal to remove unwanted phrases or words. The data imbalance approach is established using an adaptive synthetic sampling algorithm (ADASYN), then term frequency-inverse document frequency (TD-IDF) and WordNet features are employed for extracting features. Finally, MOA with the K-means technique is utilized for TDC. The proposed approach achieves better accuracy of 99.75%, 99.54%, and 98.24% when compared to the existing techniques like fuzzy rough set-based robust nearest neighbor-convolutional neural network (FRS-RNN-CNN), TopicStriker, Modsup-based frequent itemset, and rider optimization-based moth search algorithm (Modsup-Rn-MSA), hierarchical dirichlet-multinomial mixture, and multi-view clustering via consistent and specific non-negative matrix (MCCS).

Corresponding Author:

Ratnam Dodda
Department of Computer Science and Engineering, Jawaharlal Nehru Technological University
Ananthapur, India
Email: ratnam.dodda@gmail.com

1. INTRODUCTION

Text document clustering (TDC) is a significant and fast-growing research area because of the large number of text data generated by email, the Internet, social media, and text messages [1]. Document clustering is an approach for dividing a document set into diverse clusters which primarily depend on content similarity. Clustering is a significant way for learning without supervision and determining the intrinsic unlabelled data structure [2]. One of the well-defined approaches is spectral clustering, often employed to address the clustering issue [3]. If the documents are evaluated in one class is called single-label classification and if the documents are assigned to more classes, then it is called multi-label classification [4]. The users are allowed to define the inquiries regarding documents such as the content of the document [5]. The standard structure of document clustering includes knowledge distillation and text refining. During the former phase, the process of modifying a document into an intermediate form will be concept-based or document-based. Then, clustering approaches are used to extract valuable data in the next phase due to the intermediate form [6]. The summarization of scientific documents not only assists researchers in keeping themselves updated, but also minimizes time and effort in reading any scientific article [7]. The usage of ontology in clustering documents acknowledges more significant documents than the conventional term approaches. The documents are provided based on search words while the user examines the documents in the search engine [8]. The approach of cluster analysis is important for the retrieval of information because it assists in increasing the search time and removing inappropriate outcomes from the retrieved documents [9]. The clustering of documents generates an effective approach for working with documents and grouping them into various sets depending on similar documents [10]. The summarization method employs a generic or query-based technique to determine the candidate sentence for summary [11]. Most of the approaches employed for text summarization are drafted for the original text's word evaluation, paying attention to the semantic affiliations [12]. Word embedding is a kind of word representation for text, where words with the same meanings have an identical representation. Each word is denoted as a real-valued vector, generally tens or hundreds of dimensions [13]. The clustering of the document's goal is to group same documents and place different ones into various clusters [14]. The primary technique for establishing document clustering is to employ the bag of words (BOW) technique. In this technique, a unique word vocabulary from the entire document collection, corpus is established. Then, each document is expressed numerically based on the vocabulary, where each term of vocabulary is defined as a score in a specific document [15]. However, insignificant features in the documents minimize the accuracy of information retrieval which makes it challenging for the clustering approach to efficiently cluster similar documents.

Behera and Kumaravelan [16] implemented fuzzy rough set-based robust nearest neighbor-convolutional neural network (FRS-RNN-CNN) for the classification of text documents by employing a modified CNN for feature extraction. The modified CNN was generated by the hyper-parameter tuning functions like optimizer, batch size, and dropout for optimizing the feature extraction performance. This approach contains the usage of early stopping criteria and optimized dropout which manages overfitting effectively. However, the FRS-RNN-based CNN consumes greater time due to hyper-parameter tuning functions. Chandran *et al.* [17] presented a topic striker which was an approach that combined the benefits of supervised string kernels and unsupervised topic modelling (TM) for text document classification. To minimize the corpus document to the sequence of topic words, the topic proportions per document and co-occurring topic words per topic were employed. This minimized representation was utilized for the classification of text with the help of string kernels which enhanced the accuracy and minimized training time. However, dependency on predefined topic kernels limited the flexibility in diverse datasets which led to poor performance in domains with unique patterns. Yarlagadda *et al.* [18] introduced a rider optimization-based moth search algorithm (Modsup-Rn-MSA) for TDC. Initially, the document of input was provided to the pre-processing phase and then feature extraction was provided by employing term frequency-inverse document frequency (TD-IDF) and Wordnet features. Then, the feature extraction was established utilizing frequent items for the feature knowledge. At last, document clustering was employed by integrating return on assets (ROA) and MSA. This approach achieved better performance and was effectively employed for text documents. However, the Modsup-Rn-MSA approach was prone to sampling errors. Bilancia *et al.* [19] developed a hierarchical dirichlet-multinomial mixture for text document clustering. The objective function's explicit expression of the hierarchical approach was determined by employing evidence lower bound (ELBO). The coordinate ascent variational inference's (CAVI) updated equations were derived to determine the ELBO's local maxima. The computational benefits of variational interference were generated by this approach effectively. However, the ELBO was certainly a lower bound on the developed approach but the gap of variation varied among various models which led to a decrease in the performance. Xu *et al.* [20] implemented a multi-view clustering via consistent and specific non-negative matrix (MCCS) for text documents. The consistent coefficient matrix shared a consistent primary matrix, and the view-specific coefficient matrix was employed in the negative matrix formulation (NMF) issue. The regularization of the manifold was embedded into an objective function which avoided the complex geometrical structure of data space. However, the NMF complexity in the MCCS approach led to an enhanced computational demand which was less scalable for large text documents. From the overall analysis, it is seen that the existing researches had limitations like dependency on predefined topic kernels limiting flexibility, prone to sampling errors, gap of variation that varied among various models, leading to decreased performance, making it less scalable with insignificant features in the documents that minimized the accuracy of information retrieval, thereby making it challenging for the clustering approach to efficiently cluster similar documents. To overcome this issue, the mayfly optimization algorithm (MOA) with k-means approach is proposed for TDC to effectively cluster similar documents in this manuscript.

The primary contributions of this research are as follows:

- The pre-processing is established by utilizing stop word removal and stemming to remove the unwanted words or phrases.

- The data imbalance approach is performed using an adaptive synthetic sampling algorithm (ADASYN) to balance the data and then TD-IDF and WordNet features are employed for extracting features.
- MOA with k-means technique is utilized for TDC to effectively cluster similar documents.

The rest of this research paper is structured as follows: section 2 details the proposed method. In section 3 indicates the mayfly optimization algorithm with k-means approach. While section 4 discusses the result, and the conclusion of this research paper is given in section 5.

2. PROPOSED METHOD

In this research, the MOA with the k-means approach is proposed for TDC to effectively cluster similar documents. Initially, the data is obtained from Reuters-21678, 20-Newsgroup, and BBC sports datasets, and then pre-processing is established utilizing stemming and removal of stop words to get rid of the unwanted phrases or words. The data imbalance approach is established using ADASYN and then TD-IDF is employed to extract the features. Finally, MOA with the k-means technique is utilized for TDC. Figure 1 illustrates a block diagram for the proposed approach.

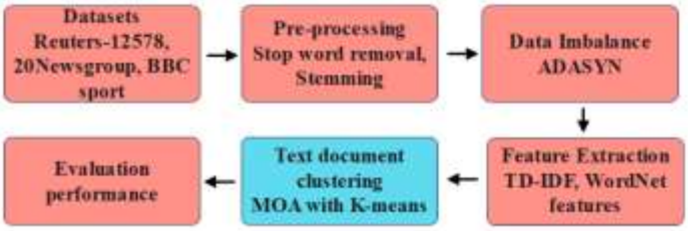

Figure 1. Block diagram of the proposed approach

2.1. Datasets

TDC is analysed using three benchmark datasets: Reuters-21578, 20 Newsgroups, and the BBC sports dataset. Reuters-21578 has 135 data classes and 21,578 texts that are labelled manually. 20-Newsgroups has nearly 20,000 documents which are separated into 20 groups. BBC sport dataset has 737 text documents which are elaborately described below.

2.1.1. Reuters-21578

The Reuters-21578 dataset [21] is one of the primary influential and commonly employed newswire article collections from the service of Reuters financial newswire that is a significant benchmark for text clustering. It includes 135 data classes and 21,578 texts that are labelled manually. There is an imbalance in the document distribution towards the class, hence there are less than 10 documents in 1 class, and over 4,000 in another. The extensive repository generates a range of significant insights into topics that are enclosed by financial publications and are accessible in various splits for optimal machine learning (ML) exploration.

2.1.2. 20-Newsgroups

The 20-Newsgroups [22] is composed of nearly 20,000 documents which are separated into 20 groups. Ken Leng collected this dataset where every new group indicates a topic. It contains 18,821 posts of newsgroups on 20 topics and the documents distribution in various classes is balanced. Certain classes are very nearer to one another and certain are far apart.

2.1.3. BBC sport dataset

The BBC dataset [23] is part of a greater collection that is generated to employ as a benchmark for the research of text mining and ML. It includes 737 text documents which contain 4613 terms football, tennis, rugby, athletics, and cricket. The number of text files associated with football, tennis, rugby, athletics, and cricket are 265, 100, 147, 101, and 124, respectively.

2.2. Pre-processing

After data collection, this phase is established utilizing stop word removal and stemming. Input data includes unwanted phrases or words that affect the process of clustering. Assume D is the database set and it contains n number of documents in datasets. Hence, the pre-processing phase is established to eliminate the redundant words from the database of text.

– Stop word removal: it is the unwanted words provided in text documents such as a, an, in, the, and so on. Removing stop words minimizes vocabulary size which speeds up processing in text clustering. Stop words frequently occur, but have little semantic value.
– Stemming: it covers terms in this step which is ineffectively not a meaningful word to its root from the language. The number of documents present in the database D is expressed in (1). The unique keywords W are acquired after extracting the keywords which is formulated in (2).

$$\mathrm{d_i} = \{\mathrm{w_j^i}, 1 \leq \mathrm{j} \leq \mathrm{m_i}\} \quad (1)$$

$$\mathrm{W} = \{\mathrm{b_i}, 1 \leq \mathrm{x} \leq \mathrm{k}\} \quad (2)$$

Where, $\mathrm{m_i}$ indicates the extracted words from documents i, and k indicates overall words in the document's unique keywords or dictionary. Stemming minimizes the unique word number that is required to be processed by an approach which increases its performance. Thus, dictionary words are acquired from the pre-processing phase and then these words are fed into the data imbalanced process.

2.3. Data imbalance using ADASYN

After the pre-processing phase, the ADASYN approach [24] is employed to balance the imbalanced data. To deal with imbalanced data is the primary aim of ADASYN for addressing the distribution of the Inherent Imbalanced class. Here, it is used to learn the imbalanced datasets. The ADASYN generates the distribution of weight between the minority class, as it faces challenges for learning synthetic data which makes the approach difficult to learn the sample compared to the majority class. There are two ways in which the ADASYN technique is enhanced. First, it minimizes the established bias due to class imbalance. Second, the decision boundary of classification is shifted regarding various data samples. The dataset $\mathrm{D_{tr}}$ is trained with samples m that comprise $\{\mathrm{x_i}, \mathrm{y_i}\}, \mathrm{i} = 1, .., \mathrm{m}$, where x_i represents instances that have a X feature space with n dimension. The degree of class imbalance d is computed which is denoted in (3). The synthetic data G is calculated for minority class is formulated in (4).

$$d = \frac{m_s}{m_l} \quad (3)$$

$$G = (m_l - m_s) \times \beta \quad (4)$$

Where, m_s represent minority class, m_l indicates majority class, and $d \in (0,1]$, $\beta \in [0,1]$ denotes a parameter to determine the defined balance which provides the synthetic data. When $\beta = 1$ the dataset is balanced, established during the process of generalization. By employing Euclidean distance, the minority class is determined by the nearest neighbours, then the majority class computes the ratio by employing the (5). The r_i is normalized by using (6).

$$r_i = \Delta_i \quad (5)$$

$$\widehat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \quad (6)$$

Where, $\widehat{r_i}$ indicates the density distribution. The ADASYN employs the r_i density distribution to evaluate the number of samples automatically for synthetic data which is provided by balancing minority data. The r_i is a distribution that denotes minority class weight for learning imbalanced data. To address the imbalance issue, the synthetic data is included for minority weights which is formulated in (7).

$$g_i = \widehat{r_i} \times G \quad (7)$$

Where, G denotes the number of synthetic samples. The minority class for data x_i, and synthetic data g_i is included for selecting the random minority data and synthetic data is produced. The ADASYN not only generates the distribution of data to β balance coefficient level, but it also learns the approach by addressing the challenge during the process of learning. Then, the data imbalance process is fed into feature extraction.

2.4. Feature extraction

The feature extraction is performed after the data imbalance process to extract keywords from the documents using TF-IDF and Wordnet features. These approaches are extracted to capture semantic meaning and significance of words in the text documents. TF-IDF is used for calculating the occurrence of each word and essential words in the document. Wordnet has two semantic relations: synonym and hyponymy which are discussed:

2.4.1. TD-IDF

TF is employed to calculate each word occurrence in a document and IDF is utilized to compute the essential word occurring in a document. It is a statistical measure widely utilized in natural language processing (NLP) to analyze the significance of a term within a document. It assumes both the frequency of term (TF) and the rarity of terms among the whole corpus document (IDF). The IDF's mathematical formula is expressed in (8). The $Q(b_l, D)$ indicate IDF of b_l^{th} word in D database, b_l denotes document words, d represents documents, and n denotes the document's collection.

$$Q(b_l, D) = log \frac{n}{|d\epsilon D: b_1 \epsilon d\}|} \tag{8}$$

2.4.2. Wordnet features

The next significant feature extraction from documents of Word is WordNet features. It is a lexical English language database that converts words into synonyms set called synsets associated with semantic relations. By using Wordnet, text analysis captures effective semantic relationships among words. The number of extracted features for three datasets are 23,944; 14,400; and 590. It describes two types of semantic relations, synonymy and hyponymy. The WordNet ontology is employed to determine the word's semantic relations that assist in the extraction procedure in three basics: i) Determining the required lexical paraphrase amount from the document of the word, ii) determining semantic net for localization, and iii) base generation for a document like the ontology of domain that is established for a document. WordNet ontology approach is established for the semantic feature realization of word documents which address the synset usage for providing paraphrases. Additionally, it finds two synonyms interchangeably in a specific context.

The ontology of the WordNet is assumed as a data processing source with synset's collection. A synset is described as a synonym set that gathers topics with significance. It extracts two kinds of features such as hyponyms and synonyms from documents. For instance, keyword synonymy and hyponym b_l in the document, and the word is indicated as C^{b_l} and B^{b_l}. After extraction of hyponyms and synonyms for each keyword, the dictionary has a $3 * k$ keyword number. By utilizing these keywords, the matrix feature is established by determining TF, IDF, synonym, and hyponym features. The extracted features are represented in (9).

$$F = \{f_{ij}, 1 \leq i \leq n; 1 \leq j \leq g\} \tag{9}$$

Where, n represents overall documents, and g indicates overall amount of extracted feature dimensions by employing hyponymy, TF, IDF, and synonymy. Using TD-IDF and wordnet features provides text document representations by incorporating not only the significance of statistical terms, but also their semantic associations which enhances the model's effectiveness. After feature extraction, the TDC is performed using MOA with a k-means approach.

3. MAYFLY OPTIMIZATION ALGORITHM WITH K-MEANS CLUSTERING

After feature extraction, the MOA with k-means is used for TDC. The MOA is an enhancement of the particle swarm approach. It integrates the benefits of particle swarm optimization (PSO), firefly approach (FA), and genetic algorithm (GA) to generate a structure of a strong hybrid approach. The technology of crossover and local search are employed based on mayfly's social behaviour. MOA differs from PSO with its optimization mechanism. While both approaches are inspired by natural phenomena, MOA has the mayfly behavior in swarm intelligence which focuses on dynamic cluster size adjustments. Whereas PSO depends on collective particle behavior moving via search space to determine optimal solutions. MOA dynamically adjusts the cluster size depending on the evolving characteristics of the dataset which enables it a dynamic nature. This means that the number of clusters vary during the optimization process which adopts the data structure. The dimensionality features used in this optimization for Reuters 21,578; 20-Newsgroup; and BBC sports datasets are 19,155; 11,520; and 472 features. Consider that the mayfly always is an adult, and then the strongest mayfly survives. In the search space, each mayfly's position indicates a potential solution. The working principle of this approach is that two groups of mayflies' female and male are generated randomly in the issue space as candidate solutions, which is indicated by a d-dimensional vector $x = (x_1, x_2, \ldots, x_d)$ and its performance is determined by the function of predefined objective $f(x)$. A vector of speed is represented as $v = (v_1, v_2, \dots, v_d)$ to determine the change in mayflies' position. Each mayfly's flight direction is a dynamic interaction between social flight and individual experience.

Initializing the male and female mayfly population and speed parameters are established. Consider that p_i^t indicates the male mayfly's present position i at the time step t in the search space. The male mayfly's speed is expressed in (10). The procedure in which the female mayfly fascinates male mayfly is adapted and the female mayfly's speed is calculated in (11).

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2}\left(pbest_{ij} - p_{ij}^t\right) + a_2 e^{-\beta r_p^2}\left(gbest_j - p_{ij}^t\right) \tag{10}$$

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_2 e^{-\beta r_{mf}^t (p_{ij}^t - q_{ij}^t)}, if\ f(q_i) > f(p_i) \\ \quad v_{ij}^t + fl * r, if\ f(q_i) \le f(p_i) \end{cases} \tag{11}$$

Where, v_{ij}^t indicates mayfly's speed i $(I = 1,2, \dots, n)$ in dimension j $(j = 1,2, \dots, d)$ at t time step, p_{ij}^t and q_{ij}^t represents male and female mayfly's position i at t time step in j dimension, a_1 and a_2 denotes positive attraction constant, β represents fixed coefficient visibility, r_{mf} indicates cartesian distance among male and female mayflies. At last, fl represents the coefficient of random walking employed while the female does not attract the male, hence randomly it flies, and r denotes the random value range [-1, 1].

To acquire $pbest_i$ and $gbest$, the fitness value is calculated. The $pbest_i$ denotes the optimal position that mayfly i has not been to. For minimization issues, individual $pbest_i$ optimal solution at the next $t + 1$ time step is computed which is expressed in (12). The objective function indicates f that characterizes the solution quality, and global optimal solution $gbest$ is formulated in (13).

$$pbest_i = \begin{cases} x_i^{t+1}, if\ f(x_i^t) < f(pbest_i\ ) \\ is\ \text{kept}\ the\ same, otherwise \end{cases} \tag{12}$$

$$gbest \in \{pbest_1, pbest_2, \dots, pbest_N | f(cbest)\} = min\{f(pbest_1), f(pbest_2), \dots, f(best_2)\} \tag{13}$$

The crossover operator indicates the two mayflies' mating procedure. The parents are chosen in an identical way as females to attract males. Generally, the selection is random or depends on their fitness function. The best male is matched by the best female, the second male is matched by the second female, and so on. The outcomes of two cross offspring are expressed in (14).

$$\begin{aligned} offspring1 &= L * male + (1 - L) * female \\ offspring2 &= L * female + (1 - L) * male \end{aligned} \tag{14}$$

Where, male indicates the male parent and female indicates the female parent, L represents a random number in a particular range and initial offspring velocity is set to zero. The male and female mayflies are updated in turn and mate. The next position is acquired by including a velocity v_i^{t+1} which is formulated in (15) and (16).

$$p_i^{t+1} = p_i^t + v_i^{t+1} \tag{15}$$

$$q_i^{t+1} = q_i^t + v_i^{t+1} \tag{16}$$

Where $p_i^0 \cup (p_{min}, p_{max}), q_i^0 \cup (q_{min}, q_{max})$

The fitness function is computed and $pbest$ and $gbest$ are updated until the termination condition is performed. In the MOA, the k-means process is added as an intermediate approach for the automatic clustering of text documents. The k-means approach [25] is one of the most popular approaches in the clustering process. It classifies a dataset as a K cluster set that explicitly the start of the process. The primary goal is to minimize the distance between the cluster and the cluster head member. Initially, the approach selects an initial cluster number as K. For this, k-means chooses random K points x_i with data $1 \le i \le K$ in the dataset as centroids where every centroid pertains to a C cluster. Then, the approach assigns every point in the dataset to a neighbor centroid. This procedure depends on the function of the objective that computes the sum of all squared distance in clusters. The estimation is performed by utilizing the objective function which is expressed in (17).

$$avgmin_c \sum_{i=1}^{K} \sum_{x_j \in C_i} d(x_j, u_i) = avgmin_c \sum_{i=1}^{K} \sum_{x_j \in C_i} |x_j - u_i|^2 \tag{17}$$

Where, $d(x_j, u_i) = |x_j - u_i|^2$ represents the distance among point and cluster centroid. x_j denotes the position of point, u_i indicates the centroid position with $i = 1, \ldots, K$, K representing the number of clusters. After assigning each cluster's points, the k-means approach updates each centroid's position by employing (18).

$$u_i = \frac{1}{|C_i|} \sum_{j \in C_i} x_j, \forall i \tag{18}$$

At last, the clusters are established. Using MOA with k-means approach effectively groups the closely associated text documents by reducing the similarity in various clusters. Notation description is given in Table 1.

Table 1. Notation description

Symbol	Description
m_i	extracted words from documents i
W	unique keywords
k	overall words in the document's unique keywords or dictionary
d	class imbalance degree
m_s	minority class
m_l	majority class
$\hat{r}_i$	density distribution
G	number of synthetic samples
x_i	minority class for data
g_i	synthetic data
$Q(b_l, D)$	IDF of b_l^{th} word in D database
b_l	document words
d	documents
n	number of documents
g	total number of extracted feature dimension
v_{ij}^t	mayfly's speed
$p_{\mathrm{ij}}^{\mathrm{t}}$ and $\mathrm{q}_{\mathrm{ij}}^{\mathrm{t}}$	male and female mayfly's position i at t time step in j dimension
a_1 and a_2	positive attraction constant
β	fixed coefficient visibility
r_{mf}	cartesian distance among male and female mayflies
fl	random coefficients
f	objective function
$\mathrm{d}(\mathrm{x_j}, \mathrm{u_i}) = \lvert\mathrm{x_j} - \mathrm{u_i}\rvert^2$	the distance between the point and cluster centroid
$\mathrm{x_j}$	position of the point
$\mathrm{u_i}$	centroid position

4. RESULTS AND DISCUSSION

In this research, the MOA with k-means approach is simulated using a Python environment with the system configuration of RAM: 16 GB, processor: Intel core i7, and operating system: Windows 10. The F1-score, recall, precision, and accuracy are employed as evaluation metrics to assess the performance of the proposed approach. The mathematical formula for these metrics is expressed in (19)-(24). TP indicates (true positive), FP shows (false positive), FN represents (false negative), and TN determines (true negative).

$$\mathrm{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TN+TP+FN+FP}} \times 100 \tag{19}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{20}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{21}$$

$$\mathrm{F1-Score} = \frac{2\times\mathrm{TP}}{2\times\mathrm{TP+FP+FN}} \times 100 \tag{22}$$

$$\mathrm{NMI} = \frac{2\times\mathrm{I(cl{:}k)}}{[\mathrm{e(cl)+e(k)}]} \tag{23}$$

$$\mathrm{ARI} = \frac{\Sigma_{\mathrm{i,j}}\binom{\mathrm{n_{i,j}}}{2} - [\Sigma_{\mathrm{j}}\binom{\mathrm{n_i}}{2} - \Sigma_{\mathrm{j}}\binom{\mathrm{n_j}}{2}]/\binom{\mathrm{n}}{2}}{\frac{1}{2}[\Sigma_{\mathrm{j}}\binom{\mathrm{n_i}}{2}] - [\Sigma_{\mathrm{j}}\binom{\mathrm{n_i}}{2}\Sigma_{\mathrm{j}}\binom{\mathrm{n}}{2}]/\binom{\mathrm{n}}{2}} \tag{24}$$

4.1. Qualitative and quantitative analysis

This section illustrates the quantitative and qualitative analysis of MOA with k-means in terms of precision, F1-score, accuracy, recall, and normalized mutual information (NMI) are provided in Tables 2-4. Table 2 indicates the performance analysis of TDC using Reuters-21578. The performance of artificial bee colony (ABC) with k-means, seagull optimization algorithm (SOA) with k-means, and golden eagle optimizer (GEO) with k-means are compared with the proposed MOA with k-means approach. Figure 2 illustrates the graphical representation of TDC employing Reuters-21578 dataset. The acquired outcomes represent that MOA with k-means achieves better accuracy of 99.75%.

Table 3 represents the performance analysis of TDC using 20-Newsgroup. The performance of ABC with k-means, SOA with k-means, and GEO with k-means are compared with the proposed MOA with k-means approach. Figure 3 indicates the graphical representation of TDC employing 20-Newsgroup dataset. The acquired outcomes denote that MOA with k-means achieves better accuracy of 99.54%.

Table 2. Performance analysis of TDC using Reuters-21578

Performance metrics	ABC with k-means	SOA with k-means	GEO with k-means	Proposed MOA with k-means
Accuracy	95.47	96.82	97.36	99.75
Precision	95.12	95.31	96.48	99.25
Recall	96.02	96.44	96.23	99.30
F1-score	95.33	95.17	96.16	99.42

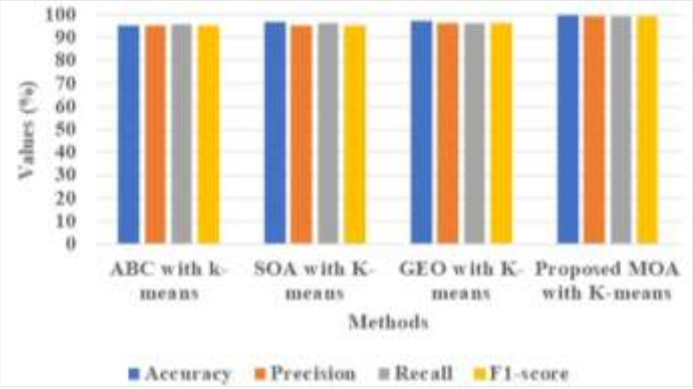

Figure 2. Graphical representation of TDC using Reuters 21578

Table 3. Performance analysis of TDC using 20-Newsgroup

Performance metrics	ABC with k-means	SOA with k-means	GEO with k-means	Proposed MOA with k-means
Accuracy	95.82	96.05	97.50	99.54
Precision	94.15	95.15	96.23	99.12
Recall	94.36	95.67	97.34	99.43
F1-score	94.28	96.38	96.41	99.06

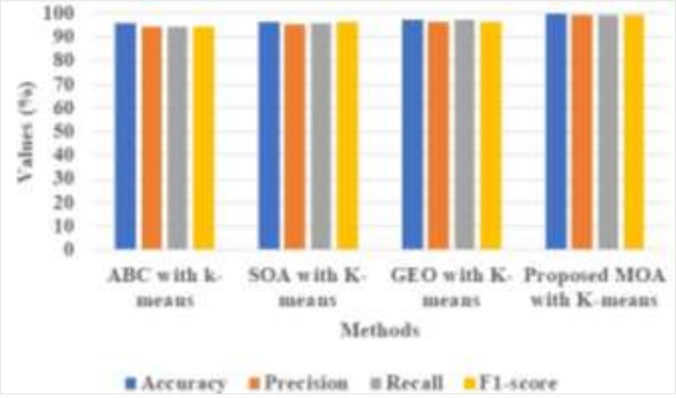

Figure 3. Graphical representation of TDC using 20-News group

Table 4 represents the performance analysis of TDC using the BBC sports dataset. The performance of ABC with k-means, SOA with k-means, and GEO with k-means are compared with the proposed MOA with k-means approach. Figure 4 displays a graphical representation of TDC employing the BBC sport dataset. The acquired outcomes denote that MOA with k-means achieves better accuracy of 98.24% respectively.

Table 4. Performance analysis of TDC using BBC sport

Performance metrics	ABC with k-means	SOA with k-means	GEO with k-means	Proposed MOA with k-means
Accuracy	95.48	96.58	97.36	98.24
Precision	94.25	95.20	95.52	97.63
Recall	95.64	94.58	96.08	97.32
F1-score	94.08	96.34	95.22	98.05
NMI	94.37	95.05	96.59	96.76
ARI	0.54	0.62	0.66	0.76

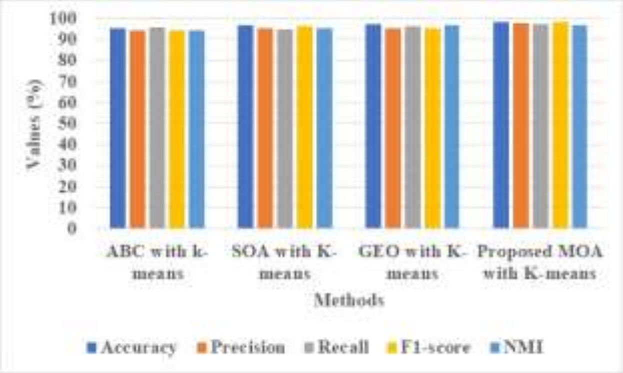

Figure 4. Graphical representation of TDC using BBC sport

4.2. Comparative analysis

This section presents the comparative analysis of MOA with the K-means technique utilizing Reuters-21578, 20-Newsgroup, and BBC sport datasets which are shown in Tables 5-7. Existing techniques like FRS-RNN-CNN [16] and TopicStriker [17] are employed to access MOA with the k-means approach utilizing the Reuters-21578 dataset. When compared to the FRS-RNN-CNN and TopicStriker, the proposed MOA with k-means attains better accuracy, recall, precision, and F1-score of 99.75%, 99.30% 99.25%, 99.42% respectively using Reuter-21578. The existing approaches like FRS-RNN-CNN and Modsup+Rn-MSA [18] are used to evaluate MOA with k-means using the 20-Newsgroup dataset. The proposed MOA with k-means attains accuracy, recall, precision, and F1-score of 99.54, 99.43% 99.12%, and 99.06% compared to existing approaches like FRS-RNN-CNN and Modsup+Rn-MSA. The hierarchical mixture of dirichlet-multinomial [19] and MCCS [20] are the two existing approaches to access MOA with the k-means technique. The MOA with k-means achieves accuracy, while that of the NMI is 98.24% and 96.76%, which is lesser when compared to the existing approach.

Table 5. Comparative analysis with existing methods using Reuters-21578

Performance metrics	FRS-RNN-CNN [16]	TopicStriker [17]	Proposed MOA with k-means
Accuracy	98.5	96.14	99.75
Precision	98.56	N/A	99.25
Recall	98.5	N/A	99.30
F1-score	98.5	N/A	99.42

Table 6. Comparative analysis with existing methods utilizing 20-Newsgroup

Performance metrics	FRS-RNN-CNN [16]	Modsup+Rn-MSA [18]	Proposed MOA with k-means
Accuracy	96.98	94.38	99.54
Precision	97.09	95.90	99.12
Recall	96.98	94.37	99.43
F1-score	97	95.57	99.06

Table 7. Comparative analysis with existing methods utilizing BBC sport

Performance metrics	Hierarchical mixture of dirichlet-multinomial [19]	MCCS [20]	Proposed MOA with k-means
Accuracy	74.22	93.72	98.24
NMI	N/A	83.18	96.76
ARI	0.58	N/A	0.76

4.3. Discussion

This section represents the benefits of the proposed technique and the limitations of existing approaches. The existing techniques have certain limitations like FRS-RNN-CNN [16], consuming greater time due to the hyper-parameter tuning functions. TopicStriker [17] has a dependency on predefined topic kernels limit flexibility in diverse datasets, leading leads to poor performance in domains with unique patterns. The Modsup+Rn-MSA [18] was prone to sampling errors. The proposed MOA with the k-means technique overcomes these existing approaches' limitations. Inspired by the behaviour of flight and Mayflie's mating procedure, it integrates the major benefits of the evolutionary approach and swarm intelligence. Also, it has the advantage of global search ability and fast convergence speed. The k-means clustering scales to large datasets and guarantees convergence. It establishes various cluster sizes and shapes like elliptical clusters. MOA enhances k-means clustering by adjusting dynamic centroids, therefore increasing convergence and minimizing sensitivity to initial centroids, also enhancing the quality of clustering for text documents. Therefore, by combining this approach, it accomplishes superior performance. In contrast to the existing techniques like FRS-RNN-CN, TopicStriker, Modsup+Rn-MSA, hierarchical mixture of dirichlet-multinomial, and MCCS, the MOA with k-means technique achieves a better accuracy of 99.75%, 99.54%, and 98.24% using Reuters-215678, 20-Newsgroup, and BBC sports datasets respectively. The limitations of the proposed MOA with k-means suffers from high-dimensional sparse data due to the dependency on centroid-based distance metrics.

5. CONCLUSION

In this research, the MOA with k-means technique is proposed for TDC to effectively cluster similar documents. TF-IDF and Wordnet features effectively extract the features, where TF-IDF is used for calculating occurrence of each word and essential words in document. WordNet ontology is employed to determine the word's semantic relations that assist in the extraction process. For text document clustering, MOA provides high convergence and reduces the risk of converging to local optima which results in accurate and robust performance. Combining MOA with k-means optimizes both the quality of the cluster and efficiency in text document clustering. The MOA with k-means technique achieves better accuracy of 99.75%, 99.54%, and 98.24% using Reuters-215678, 20-Newsgroup, and BBC sports datasets, in contrast to FRS-RNN-CN, TopicStriker, Modsup+Rn-MSA, hierarchical mixture of dirichlet-multinomial, and MCCS. In the future, the feature representation will be improved for semantic text features.

REFERENCES

[1] T. Bezdan *et al.*, "Hybrid fruit-fly optimization algorithm with k-means for text document clustering," *Mathematics*, vol. 9, no. 16, p. 1929, Aug. 2021, doi: 10.3390/math9161929.

[2] S. M. Sadjadi, H. Mashayekhi, and H. Hassanpour, "A semi-supervised framework for concept-based hierarchical document clustering," *World Wide Web*, vol. 26, no. 6, pp. 3861–3890, Nov. 2023, doi: 10.1007/s11280-023-01209-4.

[3] N. Alami, M. Meknassi, N. En-nahnahi, Y. El Adlouni, and O. Ammor, "Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling," *Expert Systems with Applications*, vol. 172, p. 114652, Jun. 2021, doi: 10.1016/j.eswa.2021.114652.

[4] R. Janani and S. Vijayarani, "Automatic text classification using machine learning and optimization algorithms," *Soft Computing*, vol. 25, no. 2, pp. 1129–1145, Jan. 2021, doi: 10.1007/s00500-020-05209-8.

[5] M. Kayest and S. K. Jain, "Optimization driven cluster based indexing and matching for the document retrieval," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 851–861, 2022, doi: 10.1016/j.jksuci.2019.02.012.

[6] B. Diallo, J. Hu, T. Li, G. A. Khan, and A. S. Hussein, "Multi-view document clustering based on geometrical similarity measurement," *International Journal of Machine Learning and Cybernetics*, vol. 13, no. 3, pp. 663–675, Mar. 2022, doi: 10.1007/s13042-021-01295-8.

[7] S. K. Mishra, N. Saini, S. Saha, and P. Bhattacharyya, "Scientific document summarization in multi-objective clustering framework," *Applied Intelligence*, vol. 52, no. 2, pp. 1520–1543, Jan. 2022, doi: 10.1007/s10489-021-02376-5.

[8] G. Urkude and M. Pandey, "Design and development of density-based effective document clustering method using ontology," *Multimedia Tools and Applications*, vol. 81, no. 23, pp. 32995–33015, Sep. 2022, doi: 10.1007/s11042-022-12506-x.

[9] B. Inje, K. K. Nagwanshi, and R. K. Rambola, "An efficient document information retrieval using hybrid global search optimization algorithm with density based clustering technique," *Cluster Computing*, vol. 27, no. 1, pp. 689–705, Feb. 2024, doi: 10.1007/s10586-023-03976-1.

[10] D. Mustafi, A. Mustafi, and G. Sahoo, "A novel approach to text clustering using genetic algorithm based on the nearest neighbour heuristic," *International Journal of Computers and Applications*, vol. 44, no. 3, pp. 291–303, Mar. 2022, doi: 10.1080/1206212X.2020.1735035.

[11] R. C. Belwal, S. Rai, and A. Gupta, "Extractive text summarization using clustering-based topic modeling," *Soft Computing*, vol. 27, no. 7, pp. 3965–3982, Apr. 2023, doi: 10.1007/s00500-022-07534-6.
[12] S. Gupta, A. Sharaff, and N. K. Nagwani, "Frequent item-set mining and clustering based ranked biomedical text summarization," *Journal of Supercomputing*, vol. 79, no. 1, pp. 139–159, Jan. 2023, doi: 10.1007/s11227-022-04578-1.
[13] S. Hosseini and Z. A. Varzaneh, "Deep text clustering using stacked AutoEncoder," *Multimedia Tools and Applications*, vol. 81, no. 8, pp. 10861–10881, Mar. 2022, doi: 10.1007/s11042-022-12155-0.
[14] J. X. Chen, Y. J. Gong, W. N. Chen, and X. Xiao, "Adaptive encoding-based evolutionary approach for Chinese document clustering," *Complex and Intelligent Systems*, vol. 9, no. 3, pp. 3385–3398, Jun. 2023, doi: 10.1007/s40747-022-00934-z.
[15] V. Mehta, S. Bawa, and J. Singh, "WEClustering: word embeddings based text clustering technique for large datasets," *Complex and Intelligent Systems*, vol. 7, no. 6, pp. 3211–3224, Dec. 2021, doi: 10.1007/s40747-021-00512-9.
[16] B. Behera and G. Kumaravelan, "Text document classification using fuzzy rough set based on robust nearest neighbor (FRS-RNN)," *Soft Computing*, vol. 25, no. 15, pp. 9915–9923, Aug. 2021, doi: 10.1007/s00500-020-05410-9.
[17] N. V. Chandran, V. S. Anoop, and S. Asharaf, "TopicStriKer: A topic kernels-powered approach for text classification," *Results in Engineering*, vol. 17, p. 100949, Mar. 2023, doi: 10.1016/j.rineng.2023.100949.
[18] M. Yarlagadda, K. Gangadhara Rao, and A. Srikrishna, "Frequent itemset-based feature selection and Rider Moth Search Algorithm for document clustering," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 4, pp. 1098–1109, Apr. 2022, doi: 10.1016/j.jksuci.2019.09.002.
[19] M. Bilancia, M. Di Nanni, F. Manca, and G. Pio, "Variational Bayes estimation of hierarchical dirichlet-multinomial mixtures for text clustering," *Computational Statistics*, vol. 38, no. 4, pp. 2015–2051, Dec. 2023, doi: 10.1007/s00180-023-01350-8.
[20] H. Xu, L. Gong, H. Xuan, X. Zheng, Z. Gao, and X. Wen, "Multiview clustering via consistent and specific nonnegative matrix factorization with graph regularization," *Multimedia Systems*, vol. 28, no. 5, pp. 1559–1572, Oct. 2022, doi: 10.1007/s00530-022-00905-x.
[21] "Dataset link for Reuter-21578," https://www.kaggle.com/datasets/thedevastator/uncovering-financial-insights-with-the-reuters-2.
[22] "Dataset link for 20-Newsgroup," https://www.kaggle.com/datasets/crawford/20-newsgroups.
[23] "Dataset link for BBC Sport," https://www.kaggle.com/datasets/maneesh99/sports-datasetbbc.
[24] A. Balaram and S. Vasundra, "Prediction of software fault-prone classes using ensemble random forest with adaptive synthetic sampling algorithm," *Automated Software Engineering*, vol. 29, no. 1, p. 6, May 2022, doi: 10.1007/s10515-021-00311-z.
[25] K. Kandali, L. Bennis, and H. Bennis, "A New Hybrid Routing Protocol Using a Modified K-Means Clustering Algorithm and Continuous Hopfield Network for VANET," *IEEE Access*, vol. 9, pp. 47169–47183, 2021, doi: 10.1109/ACCESS.2021.3068074.

BIOGRAPHIES OF AUTHORS

Ratnam Dodda he is Research Scholar in department of Computer Science and Engineering at Jawaharlal Nehru Technological University Anantapur, Andhra Pradesh, India. He is completed M.Tech. in Computer Science and Engineering at Acharya Nagarjuna University and B.Tech. in Computer Science and Engineering from Jawaharlal Nehru Technological University. His research areas are natural language processing and machine learning. He also authored or co-authored more than 10 publications. He can be contacted at email: ratnam.dodda@gmail.com.

Alladi Suresh Babu he a Professor and Director, Software Development Centre at Jawaharlal Nehru Technological University Anantapur, Andhra Pradesh, India. He holds a Ph.D. degree in Computer Science and Engineering with specialization in Data Mining. His research areas are data mining, big data, machine learning, natural language processing, and artificial intelligence. He is supervised and co-supervised more than 80 masters and awarded two Ph.D. under his supervision. Presently 8 scholars are perusing Ph.D. under his guidance. He is also authored or coauthored more than 40 publications: 10 proceedings and 30 journals. He can be contacted at email: sureshalladi.cse@jntua.ac.in.