Ingénierie des Systèmes d'Information
Vol. 29, No. 5, October, 2024, pp. 1981-1989
Journal homepage: http://iieta.org/journals/isi

Discrete Bat Algorithm for Efficient Multi-Document Summarization

K. Adi Narayana Reddy[1*], L. Lakshmi[2], K. Dhana Sree Devi[3], K. Srinivasa Rao[1], S. Naga Prasad[4], Venu Nookala[5]

[1] Department of Computer Science & Engineering, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education (IFHE), Hyderabad 501203, India
[2] Department of Artificial Intelligence and Data Science, Faculty of Science and Technology (IcfaiTech), ICFAI Foundation for Higher Education (IFHE), Hyderabad 501203, India
[3] Department of Computer Science and Engineering, Gitam University, Hyderabad 502329, India
[4] Department of Computer Science, Tara Government College, Sangareddy 502001, India
[5] Center for IOT, Madhav Institute of Technology & Science, Gwalior 474005, India

Corresponding Author Email: aadi.iitkgp@gmail.com

https://doi.org/10.18280/isi.290529

Received: 5 December 2023
Revised: 22 August 2024
Accepted: 10 September 2024
Available online: 24 October 2024

Keywords:
text summarization, classification, document summarization, Discrete Bat Algorithm, ROUGE score

ABSTRACT

The World Wide Web gives us an enormous amount of information that was once hard to find. Now, it's tough to search through lots of data for useful insights. Recently, text summarization has become a smart way to pull out important info from huge piles of text. We can think in two ways "single document" "multi document." first is about one source, while second involves many documents. often find it harder to create a precise summary when dealing with multiple sources instead of just one. To tackle this issue, this study presents the Discrete Bat Algorithm Optimization-based multi document summarizer (DBAT-MDS). This tool is designed to make multi-document summarization better. It gets compared to three different summarization techniques that are inspired by natural phenomena. All methods are tested using benchmark datasets from the Document Understanding Conference (DUC). Various metrics like the ROUGE score & F score are used for assessment. The proposed approach shows a strong improvement over the other summarizers included in this research.

1. INTRODUCTION

A lot of data exists today thanks to the growth of technology. Analyzing this data can be tough and time-consuming. However, summarization is a great method for understanding the essence of the text. Essentially, summarization provides a concise version without losing important details and semantics [1]. This not only saves time but also serves as a quick reference for gaining insights into a topic. There are two main ways to create: extractive and abstractive methods [2, 3]. An extractive summary uses parts of the original text that are most important. On the other hand, when it comes to abstractive summaries [4, 5], crafting stronger sentences from the original is crucial.

Extractive summaries [6] can be divided into two types: general and query-focused. A general summary gives a brief overview of the main points without providing any context or background. Meanwhile, query-focused summaries [7, 8] only show information that answers specific questions. Summarizing text can come from one document or many documents at once, leading to single-document summarization (SDS) or multi-document summarization (MDS) [9, 10]. SDS might miss out since it doesn't incorporate the latest or most related texts. MDS can create more useful and precise summaries by combining information from various documents written at different times and viewpoints. But it's trickier because it can contain repeated information [11].

Models often face challenges in keeping the essential details from complex data while also delivering a clear, non-redundant, factually correct, and grammatical summary. Thus, MDS requires models that can analyse documents effectively and pull together consistent information. The area for searching in multi-document summarizing is broader compared to SDS, making it harder to find key phrases. Looking at MDS this way, it becomes an optimization issue where solving it gives an outstanding summary of the most informative phrases in the original documents. Uses of MDS include summarizing product reviews, news articles, scientific papers, feedback forms, Wikipedia entries, medical texts & software interactions.

Text summarization has recently emerged as an effective method for extracting key information from large volumes of data. Based on the number of documents involved, summarization is categorized into single-document and multi-document approaches, both of which present significant challenges for researchers aiming to produce accurate summaries [12].

Multi-document text summarization is a powerful technique for generating summaries by clustering texts related to a common topic. This approach can be enhanced using optimization methods. While most optimization algorithms in the literature are single-objective, recent advancements have introduced multi-objective optimization (MOO) techniques, which have shown superior performance compared to single-objective methods. Additionally, metaheuristic-based approaches are becoming increasingly effective in the study of MOO [13].

The current research on text summarization, particularly in the context of MDS, has several limitations. Firstly, SDS often fails to capture the most recent or relevant information, as it is restricted to a single source. On the other hand, MDS, while capable of providing more comprehensive and precise summaries by combining data from multiple documents, presents additional challenges such as redundancy, factual accuracy, and maintaining grammatical coherence.

Another limitation is that many existing models struggle to effectively distill essential information from complex datasets without producing summaries that are either redundant or lose critical details. Additionally, most of the optimization algorithms applied in MDS have been single-objective, which may not adequately handle the complexity of MDS tasks. Though recent studies have introduced MOO techniques, their implementation and application still face challenges, particularly in generating non-redundant, coherent summaries.

Moreover, while metaheuristic-based methods have shown promise in enhancing MOO for MDS, their effectiveness is still being explored, and they have not been fully optimized for large-scale or highly complex datasets. The broader search space in MDS compared to SDS makes it harder to identify the most informative phrases, turning it into a challenging optimization problem that requires more refined techniques.

The proposed research offers several key advantages

Novel optimization approach: A new Discrete Bat Algorithm is introduced to tackle binary optimization problems, which has not been explored extensively in the context of text summarization.

Effective handling of MDS: By representing the multi-document summarization task as a binary optimization problem, the proposed Discrete Bat Algorithm is well-suited to manage the complexities of MDS, including minimizing redundancy and enhancing coherence.

Rigorous evaluation: The proposed model is validated using well-known datasets and evaluated with the ROUGE tool, ensuring robust performance assessment and comparability with existing approaches.

These contributions aim to enhance the quality and effectiveness of multi-document summarization, addressing the shortcomings of current methods.

The structure of this paper is as follows: Section 2 covers related works on MDS. Section 3 identifies the multi-document summarization problem along with its objective function. In Section 4, we discuss standard versus Discrete Bat Algorithms. Experimental analysis proving the importance of our proposed algorithm is covered in section 5. Finally, section 6 wraps up with future work recommendations.

2. LITERATURE REVIEW

The research [14] provides a comprehensive overview of text summarization techniques, focusing on both extractive and abstractive methods, and evaluates several algorithms, including TextRank, Seq2Seq, and BART. While TextRank is noted for its simplicity and speed, particularly with short texts, Seq2Seq and BART leverage deep learning to produce high-quality summaries, with BART showing superior performance on benchmark datasets. However, the study is limited in scope, as it does not explore other potential models or optimization techniques that could enhance summarization performance, particularly for abstractive methods or multi-document scenarios. Additionally, the analysis does not address the specific challenges of generating coherent and accurate summaries across multiple documents.

The study by Rautray et al. [15] highlights those conventional techniques for multi-document summarization, including TF-IDF, graph-based methods, latent semantic analysis (LSA), and clustering, depend extensively on manually designed features such as sentence length, proper nouns, sentence placement, and relationships between sentences. Despite their goal of producing coherent summaries, these methods encounter considerable difficulties in efficiently summarizing multiple documents and minimizing redundancy while retaining relevance. Moreover, previous statistical approaches have not consistently delivered high-quality text extraction, underscoring the need for more advanced and effective summarization techniques.

The research emphasizes the move towards global optimization techniques like particle swarm optimization, differential evolution, and genetic algorithms (GA) to overcome the shortcomings of traditional statistical methods in text summarization [16]. GA was first employed to frame summarization as an optimization issue while retaining the sequence of documents. Further developments, including the Ide dec-hi technique, showed improvements in preserving document order within GA-based approaches and adapting to fuzzy retrieval systems, enhancing information extraction through adaptive query modelling. Nonetheless, the study might not thoroughly examine how these global optimization methods compare to and integrate with newer summarization techniques.

The research explores the development of PSO-based summarizers tailored for single documents. Rautray et al. [15] created a PSO summarizer that targeted content coverage and redundancy by integrating these features into a single goal function. Subsequent studies built on this approach by applying the same objective function but focusing on text attributes instead of sentence weights. Further investigations introduced PSO-based extractive summarizers that used objective functions aligned with ROUGE scores, highlighting factors such as readability, summary length, and breadth of information. However, these studies may not fully assess the applicability of these methods to multi-document summarization or compare their effectiveness with other optimization approaches.

Table 1. Literature summary

Research Focus	Advantages	Gaps
Text Summarization Techniques [14]	Comprehensive review of extractive and abstractive methods; Evaluates algorithms (TextRank, Seq2Seq, BART); Highlights BART's superior performance on benchmarks.	Limited exploration of models beyond TextRank, Seq2Seq, and BART; Lacks focus on abstractive summarization for multi-document scenarios; Does not address challenges in coherence across multiple

		documents.
Global Optimization Techniques [16]	Discusses use of PSO, DE, GA for text summarization; Highlights GA's role in preserving document order and adapting to fuzzy retrieval systems.	Lacks detailed comparison with newer techniques; Limited integration analysis of global optimization methods with other modern approaches.
Global Optimization Techniques [16]	Discusses use of PSO, DE, GA for text summarization; Highlights GA's role in preserving document order and adapting to fuzzy retrieval systems.	Lacks detailed comparison with newer techniques; Limited integration analysis of global optimization methods with other modern approaches.
PSO-Based Summarizers for Single Documents [17]	Describes PSO methods targeting content coverage and redundancy; Aligns with ROUGE scores for evaluating readability and length.	Limited applicability to multi-document summarization; Lack of comparative analysis with other optimization approaches.
PSO in Multi-Document Summarization [18]	Demonstrates use of PSO for clustering sentences to ensure coverage and diversity; Balances content coverage, variety, and length.	Inadequate evaluation of performance on complex document collections; Does not address scalability or integration challenges with other methods.
Text Summarization with Word Association and Data Mining [19]	Uses word association and data mining (K-Means, TF-IDF) for summarization; Effectively removes irrelevant words and clusters similar texts; Potential for AI integration.	Does not examine challenges in technique integration; Limited assessment of performance on diverse datasets.
Extractive Automatic Text Summarization System [20]	Proposes an MOO-based extractive summarization system; Utilizes evolutionary algorithms for non-dominated summary selection; Evaluated on DUC datasets with ROUGE metrics.	Lacks evaluation on diverse or real-world datasets; Limited exploration of performance in practical applications.
Evaluation Metrics and Datasets [21]	Reviews metrics and datasets; Surveys recent models (e.g., transformers, graph-based); Discusses challenges and applications in news, science, and social media.	Does not deeply explore limitations of current models; Insufficient insights into emerging techniques for overcoming existing challenges.

The research highlights the application of PSO in multi-document summarization [18], where it groups sentences by similarity and relevance to achieve thorough content coverage and diversity. Another approach examined the use of similarity measures to balance content coverage, variety, and summary length across various document sets. However, the study may not fully evaluate how well these methods perform with highly diverse or complex document collections and does not address potential issues related to scalability or the integration of these techniques with other summarization methods.

The research [19] introduces a robust text summarization technique that utilizes a word association algorithm to separate and merge sentences, along with data mining tools like the K-Means algorithm and Term Frequency-Inverse Document Frequency (TF-IDF) for assessing text attributes. The results demonstrate that this method effectively improves summarization by eliminating irrelevant words and efficiently clusters similar texts. There is potential for further enhancement by integrating this approach with other AI techniques, such as fuzzy logic or evolutionary algorithms, to boost summarization effectiveness and accelerate clustering. However, the research does not address the possible challenges or limitations of integrating these techniques or their performance with diverse and complex datasets.

The extractive automatic text summarization system [20] designed to extract a concise subset of sentences from large multi-document texts. It starts with preprocessing steps, including sentence segmentation, word tokenization, stop-word removal, and stemming to organize the original document collection. The summarization challenge is then approached as a MOO problem, aiming to balance content coverage with the reduction of redundancy. An evolutionary sparse multi-objective algorithm is employed to solve this MOO problem, generating a set of non-dominated summaries (Pareto front). A new criterion is introduced for selecting the optimal summary from this set. The system is evaluated using DUC datasets, with the summaries assessed through ROUGE metrics and compared with other methods in the literature. Nonetheless, the research may not thoroughly address how well the method performs with varied or complex document collections or in practical real-world scenarios beyond the test datasets.

The comprehensive review of the evaluation metrics used to measure the performance of text summarizers, as well as the datasets commonly used for their training and testing [21]. It surveys the latest advancements in text summarization, including deep learning models like transformers and graph-based techniques. The study also discusses major challenges and open problems in the field, such as producing summaries that are coherent and accurately represent the main ideas of the original text. Furthermore, it highlights the various applications of text summarization, such as in summarizing news articles, scientific papers, and social media content in Table 1. However, the paper may not sufficiently examine the limitations of current models or offer detailed insights into new techniques that could help overcome these challenges.

In recent years, evolutionary algorithms have been extensively applied across various domains. However, there has been no research to date that specifically adapts the BAT algorithm into a discrete format to tackle the challenges associated with multi-document summarization.

3. PROBLEM STATEMENT

In this section, the multi-document summarization has been defined and the processes that the multiple documents undergo so that it can be addressed by evolutionary optimization algorithms were discussed.

3.1 Multi-document summarization

An automated procedure that creates a succinct and complete document from numerous documents is called Multi-Document Summarization. For summarizing the contents of multiple documents into a single concise document that holds the information of complete documents contents can be processed in three phases namely preprocessing, Computation of sentence score and Sentence similarity computation. In this section, all three phases are discussed in detail.

3.2 Preprocessing

The collection of documents $D = \{D_1, D_2, \dots, D_N\}$ where D_i denotes the i$^{\text{th}}$ document.

3.3 Segmentation

Every document D_i , will be subject to sentence fragmentation and it can be represented as $D_i = \{S_{i1}, S_{i2}, \dots, S_{in}\}$ where $S_{i,j}$ represents the j^{th} sentence of i$^{\text{th}}$document. And the n represents the maximum number of sentence in document i.

3.4 Token

Every sentence $S_{i,j}$ are subject to further break out as terms in the sentence and it can be represented as $S_{i,j} = \{t_{ij1}, t_{ij2}, \dots, t_{ijm}\}$ representing the distinct terms in the sentence j of i$^{\text{th}}$ document. And m represents the total number of distinct terms in the respective sentence and it varies from sentence to sentence.

Removal of stop words: It is a standard procedure in document summarization where the articulation words such as “a, an, the, etc.” will be removed from the document.

3.5 Stemming

Stemming is the process of fixing the derived words with its root word. For example, “playing”, “plays”, “played” and all are connected to the root word called “play”. Hence the places where these stems extended words are there in the tokens, it can be replaced with the stem word. This process will be carried out in our preprocessing of multi-document summarization to reduce the time distinct number of words in the token that will impact the complexity of the problem in a huge number

3.6 Sentence score computation

The sentence of every document will be subject to quantification for computation purposes. In this regard, since the representation of multi-document summarization does not need any tracking of which document the sentence comes from, the index of the document can be relaxed from representation. From now on, the sentences of all the documents shall be serialized and the tokens of each sentence will be marked as $t_{i,j}$ where i represents the sentence and j represents the j^{th} word of i^{th} sentence.

For each sentence S_i a sum of term frequencies value needs to be computed for quantification of the sentence and it is called as sentence score (S). The computation of sentence score can be represented mathematically as

$$S_{i,j} = T \times log\left(\frac{n}{n_j}\right) \tag{1}$$

$$T = |t_j| \in S_i \tag{2}$$

where, T represents the term frequency of the term j in sentence i, (i.e., the number of times term j occurs in sentence S_i).

3.7 Sentence similarity computation

The similarity between the sentences can be quantified mathematically as:

$$SIM(S_i, S_k) = \frac{\sum_{j=1}^{m} \mathbb{S}_{i,j} \times \mathbb{S}_{k,j}}{\sqrt{\sum_{j=1}^{m} \mathbb{S}_{i,j}^2 \times \mathbb{S}_{k,j}^2}} \tag{3}$$

where, S_i and S_k represents two different documents and j represents the term in the document.

3.8 Objective function

The objective of multi-document summarization is to concise the content of multiple documents in a readable form without repetition of contents and with all vital information. All these three are three different objectives, and hence the computational factor of these three objectives to be carried out for every generated summary in different forms.

3.9 Coverage of vital content

The coverage of vital content present in the sentence S_i with respect to the actual output summary (O)can be formulated as

$$f_1(S_i) = SIM(S_i, O) \tag{4}$$

where, i ranges from 1 to n (i.e., the sum of total number of sentences in D)$.O$ is a collection of sentences of the final summary and it can be represented as $O = \{S_{O1}, S_{O2}, \dots, S_{Oy}\}$ such that y is the total number of sentences in O.

3.10 Cohesion between sentences

Similarity between the contents in the sentences can be evaluated as:

$$f_2(S_i) = 1 - SIM(S_i, S_k) \tag{5}$$

where, i is the current sentence and $k = 1, \dots, n$ representing all the other sentences in D.

3.11 Readability

Readability defines the readiness of the document to be the summary of relevant information and it can be represented as:

$$f_3(S_i) = SIM(S_i, S_k) \tag{6}$$

where, i is the current sentence and $k = 1, \dots, n$ representing all the other sentences in D.

Summarizing all the objectives of every sentence, the objective of every sentence can be represented as:

$$f(S_i) = \sum_{z=1}^{3} f_z(S_i) \tag{7}$$

And the objective formulation of multi-document summarization can be defined as:

$$Maximize \sum_{i=1}^{i=R} f(S_i) \quad (8)$$

where, R is the number of sentences in the final predicted summary.

4. DISCRTE BAT ALGORITHM FOR MULTI-DOCUMENT SUMMARIZATION

In this section, the standard bat algorithm and the proposed Discrete Bat Algorithm for Multi-Document Summarization are described.

4.1 Standard BAT algorithm

In 2010, Yang [22] presented the Bat technique for addressing continuous optimization issues. It was designed to address problems with single-objective optimization. The foundations of the bat algorithm are as follows. Echolocation is a kind of sonar that bats may utilize to locate prey. Typically, bats locate objects by making a loud noise and listening for the echo. This method is based on bat behavior and takes the following aspects into account:

•Bats utilize echolocation to detect distance and can distinguish between food, prey, and obstacles.

•From the wavelengths x_i point, bats fly at a random velocity v_i and volume A_0.The wavelength of each bat may be dynamically adjusted based on the target's distance. The loudness is a dynamic value ranging between A_0 and A_{min} [23].

4.2 Motion of BATS

Each bat moves closer to those with better responses. In the interim, the frequency and speed of each bat are updated across the number of iterations. For the subsequent iteration (*t*+1), the following adjustments are made to each of the bat:

$$f_i = Fit_{min} + (Fit_{max} - Fit_{min}) \times \beta \quad (9)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x^*) \times f_i \quad (10)$$

$$x_i^{t+1} = v_i^{t+1} + x_i^t \quad (11)$$

where, β is a random integer between 0 and 1, and x^* is the best global solution found from iteration 1 to t. In the bat algorithm, the neighborhood search is conducted using a random walk, which is represented by a random number

$$x_{new} = x_{old} + \varepsilon A^t \quad (12)$$

where, A^t is the average loudness of all bats and ε is a vector with values ranging from -1 to +1 [23].

4.3 Pulse emission and loudness

The pulse emission and loudness of a bat are inversely associated; when the bat finds food, its loudness decreases and its pulse emission increases, and vice versa. The pulse emission and loudness are mathematically represented as

$$A_i^{t+1} = \alpha A_i^t \quad (13)$$

$$r_i^t = r_i^0[1 - exp\,(-\gamma t)] \quad (14)$$

where, α and γ are constants.

4.4 Discrete Bat Algorithm

The representation of solution is in binary form where 1 represents the sentence is selected to be placed in the predictive summary and the 0 represents it is not.

Hence the position update policy and velocity update equations can be reframed as:

$$v_i^{t+1} = v_i^t \oplus (x_i^t \ominus x^*) \quad (15)$$

$$x_i^{t+1} = v_i^{t+1} \oplus x_i^t \quad (16)$$

where, $\ominus$ represents the Boolean difference operator and $\oplus$ represents the Boolean adder operator.

$$x_{new} = x_{old} \oplus \varepsilon A^t \quad (17)$$

Algorithm 1: DiscreteBat algorithm for DBAT-MDS

```
Input: The parameters $\rho, \alpha, \beta, \gamma$ upper and lower bound
Objective Function $f$
Set the parameters Pulse Frequency $PF_i$, Pulse Rates $r_i$ and
Loudness $A_i$ [23]
Initialize $Pop$- Number of Bats, $M_t$, $t = 1$
for each $i \in 1, \ldots, Pop$ do
$x_i \leftarrow rand(0,1,n)$
for each $z \in 1:3$ do
$[Fit_{i,z}] \leftarrow f_z(x_i)$
end for
end for
repeat
for each $z \in 1:3$ do
$[GBest_z] \leftarrow x(max([Fit_z]))$
end for
for each $i \in 1, \ldots, Pop$ do
Update the velocity using Eq. (15) w.r.t. all objectives
end for
for each $i \in 1, \ldots, Pop$ do
Update the position using Eq. (16) w.r.t. all objectives
end for
$y_i \leftarrow$ Initializing Random Solution using Eq. (17)
    if ($rand<A_i$&& $Fit(x_i) < f(Gbest_z)$) then
$x_i \leftarrow y_i$
        Increase $r_i$
        Deduce $A_i$
end if
$t \leftarrow t + 1$
until ($M_t \geq t$)
Output: Gbest for objectives z=1,2,3
```

The DBAT is an evolutionary optimization technique inspired by the natural echolocation behavior of bats, tailored specifically for discrete optimization tasks such as MDS. In the context of MDS, the goal is to extract a concise yet informative summary from a collection of documents, optimizing for multiple objectives like relevance, redundancy minimization, and informativeness.

DBAT uses a population of "bats," each representing a potential solution or summary. These bats explore the solution space using parameters like pulse frequency, pulse rates, and loudness, which guide the balance between exploration (searching new areas) and exploitation (refining known good solutions). The algorithm initializes with a set of randomly generated solutions and iteratively updates each bat's position and velocity based on the fitness of solutions according to multiple objective functions.

During each iteration, the algorithm identifies the global best solutions across all objectives and updates the velocity and position of each bat to move closer to these optimal points. The bats dynamically adjust their pulse rates and loudness to fine-tune the search process, with pulse rates increasing and loudness decreasing as they converge toward better solutions.

A random solution generation step introduces diversity into the population, helping to avoid premature convergence to suboptimal solutions. The process repeats until a predefined stopping criterion, such as the maximum number of iterations, is reached. The final output is a set of globally optimal solutions for each objective, representing the best possible summaries according to the criteria defined. The DBAT's ability to handle multiple discrete objectives simultaneously makes it a versatile tool for complex optimization problems beyond just summarization, such as feature selection, scheduling, and other combinatorial optimization tasks.

Characteristics of DBAT-MDS:

Discrete search space: Unlike continuous optimization, DBAT operates in a discrete space where solutions are represented in binary or categorical forms.

Multi-objective optimization: DBAT-MDS handles multiple objectives simultaneously, allowing for the generation of summaries that are balanced according to several criteria.

Exploration vs. exploitation balance: The algorithm dynamically balances exploration (searching new areas) and exploitation (refining known good solutions) through the adjustment of parameters like pulse rate and loudness.

5. EXPERIMENTAL ANALYSIS

In this part, the experimental setup used to test the proposed model is described, together with the performance measures, dataset, and experimental findings.

5.1 Experimental setup and dataset

The proposed algorithm is implemented in MATLAB 2018a version in a computer system with Intel Core i5 processor with 2.1 GHz clock speed with 8 GB RAM and 512 SSD. The datasets used to evaluate the proposed model include Document Understanding Conference (DUC). There are 2 datasets in DUC in which one with 50 clusters and the other with 45 clusters respectively. Each cluster will have 25 documents that are to be summarized to a maximum of 250 words. The average number of sentences in every document in DUC2006 is 30.12 and in DUC2007 it is 37.5 sentences.

5.2 Performance metrics

To evaluate the performance of the proposed model, an evaluation metric tool namely ROUGE-1.5.5 is used. Among the methods available in ROUGE for evaluation, we used ROUGE-N where the N represents the N-gram match between predicted and actual summaries.

$$ROUGE - N = \frac{\sum_{S\in O}^{\square} \sum_{N-gram\in S}^{\square} C_M}{\sum_{S\in O}^{\square} \sum_{N-gram\in S}^{\square} C} \quad (18)$$

where, N represents the N-gram value, C_M represents the maximum number of occurrence of words both in predicted and actual summary and C represents the total occurrence of words in actual summary. And N represents the words count. For example, $N = 1$ represents the single words and $N = 2$ represents the two words together as like actual summary.

Apart from ROUGE-N, we computed the statistical metrics such as F-Score, Precision and Recall using Actual summary (O_A) and predicted summary (O_P).

$$Precision = \frac{|O_A \cap O_P|}{|O_P|} \quad (19)$$

$$Recall = \frac{|O_A \cap O_P|}{|O_A|} \quad (20)$$

$$F - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (21)$$

5.3 Performance analysis

To demonstrate the effectiveness of the proposed model, it is compared against several state-of-the-art algorithms, specifically focusing on key performance metrics. The algorithms used for comparison include the Cat Search Optimization algorithm (Cat) [23], the Harmony Search algorithm (Har) [24], and the Particle Swarm Optimization algorithm (Par.Swarm) [25]. While the experiments utilize the well-established DUC dataset and ROUGE evaluation metrics, ensuring a rigorous evaluation process, they lack a direct comparative analysis with other existing optimization algorithms on the same dataset. This gap prevents a full demonstration of the proposed method's superiority.

On the DUC2006 dataset, DBAT-MDS outperforms other methods significantly as shown in Table 2. It achieves a ROUGE-1 score of 0.447, exceeding Cat (0.423), Har (0.415), and Par.Swarm (0.422), indicating superior unigram overlap and content relevance. DBAT-MDS also leads with a ROUGE-2 score of 0.09, compared to Cat (0.08), Har (0.07), and Par.Swarm (0.07), showcasing its enhanced ability to capture bigram information and provide more detailed summaries. Furthermore, its F-Score of 0.590 surpasses the scores of Cat (0.521), Har (0.472), and Par.Swarm (0.443), demonstrating its overall effectiveness in generating summaries with high relevance and comprehensive coverage. The performance analysis is represented in Figure 1.

Table 2. Comparison of proposed vs existing algorithms w.r.t. performance matrices

Algorithms	**DUC2006**		
	ROUGE-1	**ROUGE-2**	**F-Score**
Cat	0.423	0.08	0.521
Har	0.415	0.07	0.472
Par.Swarm	0.422	0.07	0.443
DBAT-MDS	**0.447**	**0.09**	**0.590**
Algorithms	**DUC2007**		
	ROUGE-1	**ROUGE-2**	**F-Score**
Cat	0.431	0.432	0.432
Har	0.422	0.423	0.422
Par.Swarm	0.401	0.401	0.401
DBAT-MDS	**0.492**	**0.493**	**0.493**

On the DUC2007 dataset, DBAT-MDS demonstrates superior performance across all metrics as shown in Table 3. It achieves the highest ROUGE-1 score of 0.492, surpassing Cat (0.431), Har (0.422), and Par.Swarm (0.401), reflecting its superior ability to capture relevant content. DBAT-MDS also leads with a ROUGE-2 score of 0.493, compared to Cat (0.432), Har (0.423), and Par.Swarm (0.401), indicating its effectiveness in capturing bigram information and producing more coherent summaries. Additionally, DBAT-MDS's F-Score of 0.493 exceeds that of Cat (0.432), Har (0.422), and Par.Swarm (0.401), showcasing its ability to balance precision and recall, resulting in more accurate and comprehensive summaries. The performance analysis is represented in Figure 2.

When evaluating ROUGE-1 scores on the DUC2007 dataset, DBAT-MDS surpasses the Cat algorithm by 13.4%, Harmony Search by 14.4%, and Particle Swarm by 18%. In terms of ROUGE-2, DBAT-MDS exceeds the Cat algorithm by 0.6%, Harmony Search by 9.6%, and Particle Swarm by 15.4%. For the F-Score, DBAT-MDS outperforms the Cat algorithm by 7.8%, Harmony Search by 15.1%, and Particle Swarm by 28.9%.

In comparing the performance of DBAT-MDS with other algorithms based on ROUGE-1 and ROUGE-2 metrics, for DUC2006 Dataset, DBAT-MDS generally outperforms its counterparts as shown in Table 2.

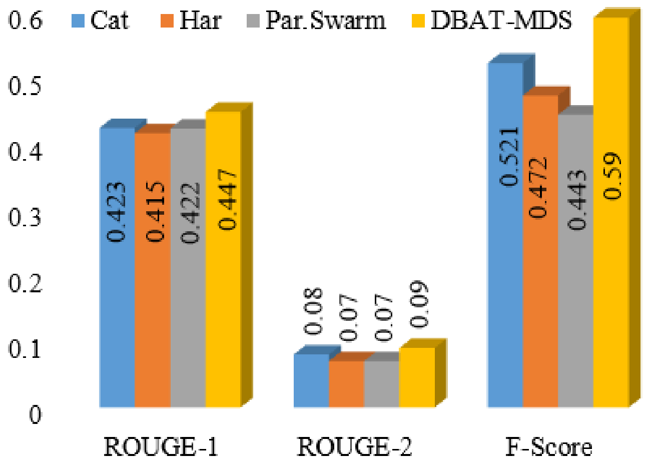

Figure 1. Graphical interpretation of algorithms w.r.t. performance metrics for DUC2006 dataset

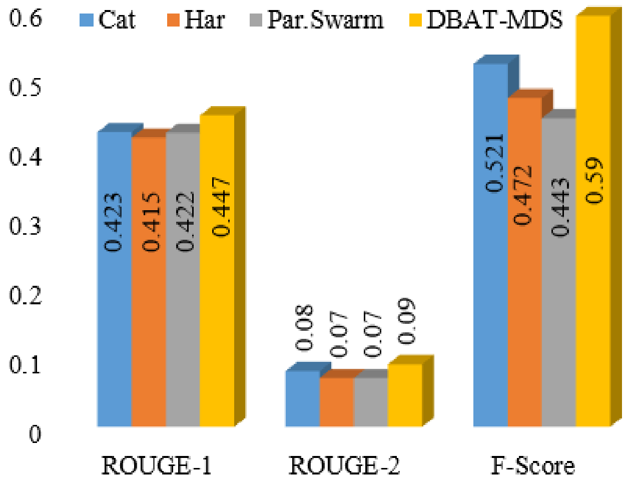

Figure 2. Graphical interpretation of algorithms w.r.t. performance metrices for DUC2007

For ROUGE-1, DBAT-MDS achieves the highest best score of 0.447 and a strong mean score of 0.421, indicating superior relevance in content summarization. This surpasses Cat (best: 0.443, mean: 0.382), Har (best: 0.434, mean: 0.412), and Par.Swarm (best: 0.422, mean: 0.414). Although DBAT-MDS has a lower worst score of 0.345, its higher best and mean scores highlight its effectiveness in generating relevant summaries as shown in Figure 3.

Table 3. Comparison of proposed vs existing algorithms w.r.t. ROUGE on final population individuals for DUC2006 dataset

Algorithms	ROUGE-1		
	Best	Worst	Mean
Cat	0.443	**0.419**	0.382
Har	0.434	0.392	**0.412**
Par.Swarm	0.422	0.389	0.414
DBAT-MDS	**0.447**	0.345	0.421
Algorithms	ROUGE-2		
	Best	Worst	Mean
Cat	0.093	**0.071**	0.081
Har	0.084	0.064	0.072
Par.Swarm	0.072	0.055	0.066
DBAT-MDS	**0.094**	0.049	**0.088**

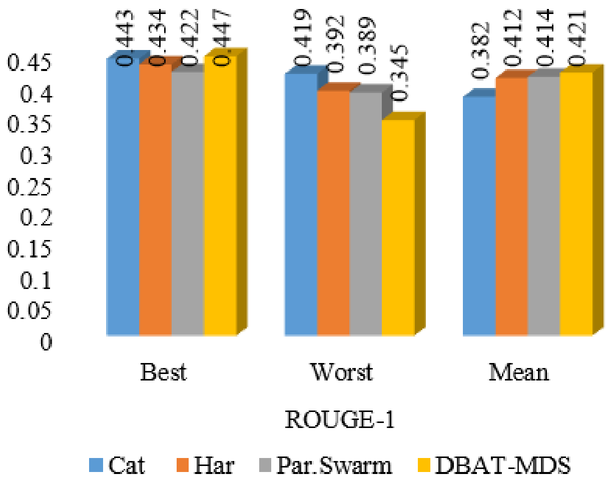

Figure 3. Graphical interpretation of algorithms w.r.t. ROUGE-N values of final population for ROUGE-1 on DUC2006 dataset

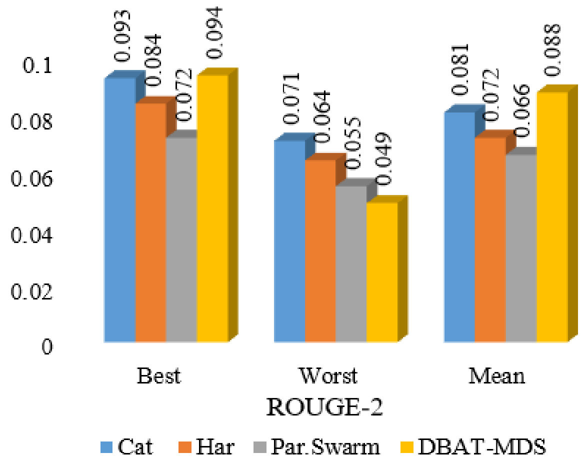

Figure 4. Graphical interpretation of algorithms w.r.t. ROUGE-N values of final population for ROUGE-2 on DUC2006 dataset

Table 4. Comparison of proposed vs existing algorithms w.r.t. ROUGE on final population individuals for DUC2007 dataset

Algorithms	ROUGE-1		
	Best	Worst	Mean
Cat	0.422	**0.412**	0.419
Har	0.416	0.402	0.414
Par.Swarm	0.411	0.411	0.402
DBAT-MDS	**0.492**	0.316	**0.453**
Algorithms	**ROUGE-2**		
	Best	**Worst**	**Mean**
Cat	0.091	**0.082**	**0.082**
Har	0.082	0.073	0.077
Par.Swarm	0.074	0.075	0.076
DBAT-MDS	**0.091**	0.071	0.085

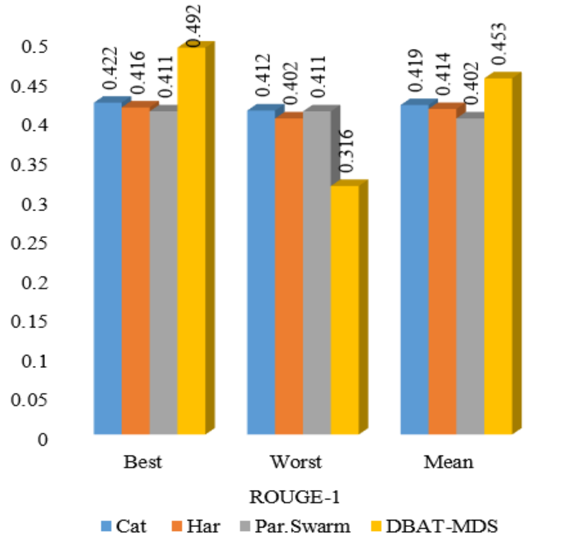

Figure 5. Graphical interpretation of algorithms w.r.t. ROUGE-N values of final population for ROUGE-1 on DUC2007

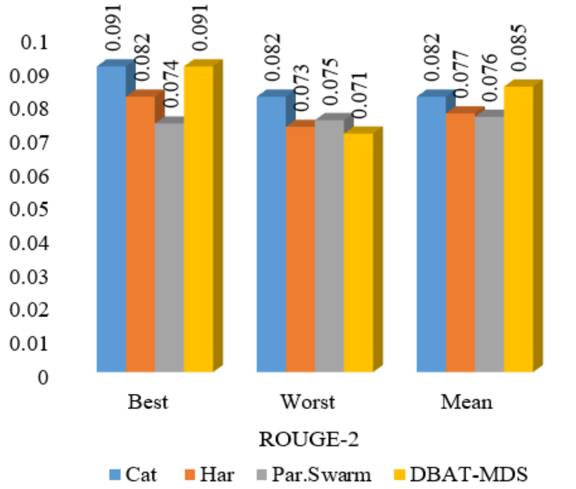

Figure 6. Graphical interpretation of algorithms w.r.t. ROUGE-N values of final population for ROUGE-2 on DUC2007 dataset

For ROUGE-2, DBAT-MDS also leads with the highest best score of 0.094 and the highest mean score of 0.088, outperforming Cat (best: 0.093, mean: 0.081), Har (best: 0.084, mean: 0.072), and Par.Swarm (best: 0.072, mean: 0.066). Despite having the lowest worst score of 0.049, the overall higher scores of DBAT-MDS indicate its enhanced ability to capture bigram information, resulting in more detailed and coherent summaries compared to the other algorithms (refer to Figure 4).

These results suggest that DBAT-MDS is particularly effective in both ROUGE-1 and ROUGE-2 evaluations, offering more accurate and comprehensive summarization performance relative to the other algorithms tested.

The worst case of proposed model in ROUGE-2 for DUC2006 dataset is deviated by 32% which shows the diversity is collection of results throughout the search. And the average case of proposed model, falls under the positive curve of the proposed model which intends to show that the proposed model holds a greater number of optimal results at the end of the search.

A comparative analysis of the algorithms using ROUGE-1 and ROUGE-2 metrics shows that DBAT-MDS generally outperforms the others is shown in Table 4.

For ROUGE-1, DBAT-MDS achieves the highest best score of 0.492 and the highest mean score of 0.453, indicating its superior ability to capture relevant content. In comparison, Cat has a best score of 0.422 and a mean of 0.419, Har has a best score of 0.416 and a mean of 0.414, while Par.Swarm scores 0.411 for both its best and mean. Although DBAT-MDS records a lower worst score of 0.316, its significantly higher best and mean scores demonstrate its overall effectiveness in summarization as shown in Figure 5.

For ROUGE-2, DBAT-MDS again leads with a best score of 0.091 and a mean score of 0.085, outperforming Cat (best: 0.091, mean: 0.082), Har (best: 0.082, mean: 0.077), and Par.Swarm (best: 0.074, mean: 0.076). Despite having the lowest worst score of 0.071, the superior best and mean scores of DBAT-MDS highlight its enhanced capability to capture bigram overlaps, resulting in more coherent and informative summaries as shown in Figure 6.

6. CONCLUSION

This research introduces a DBAT-MDS designed to generate high-quality extractive summaries by selecting the most relevant sentences from multiple documents. The performance of DBAT-MDS, along with other summarizers, is evaluated using DUC dataset, with effectiveness measured by ROUGE metrics and F-scores. The results indicate that DBAT-MDS consistently outperforms other methods, including Harmony Search, Particle Swarm Optimization, and Cat Swarm Optimization, achieving higher scores in both ROUGE and F-score evaluations.

Despite its advantages, DBAT-MDS relies on an evolutionary strategy that presents challenges, such as computational time and the need for precise parameter tuning. These parameters are currently set through a data-driven process, which may not always yield optimal results. To address this, future research will focus on developing more systematic parameter optimization techniques and exploring other nature-inspired algorithms, such as genetic algorithms or ant colony optimization, to further enhance the performance and efficiency of DBAT-MDS.

REFERENCES

[1] Aliguliyev, R.M. (2009). A new sentence similarity

measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications, 36(4): 7764-7772. https://doi.org/10.1016/j.eswa.2008.11.022

[2] Ježek, K., Steinberger, J. (2008). Automatic text summarization (the state of the art 2007 and new challenges). In Proceedings of Znalosti, pp. 1-12.

[3] Lloret, E., Palomar, M. (2012). Text summarisation in progress: A literature review. Artificial Intelligence Review, 37: 1-41. https://doi.org/10.1007/s10462-011-9216-z

[4] Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. Expert Systems with Applications, 41(9): 4158-4169. https://doi.org/10.1016/j.eswa.2013.12.042

[5] Oliveira, H., Ferreira, R., Lima, R., Lins, R. D., Freitas, F., Riss, M., Simske, S.J. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. Expert Systems with Applications, 65: 68-86. https://doi.org/10.1016/j.eswa.2016.08.030

[6] Alguliev, R.M., Aliguliyev, R.M., Hajirahimova, M.S. (2012). GenDocSum+MCLR: Generic document summarization based on maximum coverage and less redundancy. Expert Systems with Applications, 39(16): 12460-12473. https://doi.org/10.1016/j.eswa.2012.04.067

[7] Mani, I., Maybury, M.T. (1999). Advances in Automatic Text Summarization. MIT Press.

[8] Wan, X., Xiao, J. (2010). Exploiting neighborhood knowledge for single document summarization and keyphrase extraction. ACM Transactions on Information Systems (TOIS), 28(2): 8. https://doi.org/10.1145/1740592.1740596

[9] Fattah, M.A., Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech & Language, 23(1): 126-144. https://doi.org/10.1016/j.csl.2008.04.002

[10] França, V.V. (2017). Entanglement and exotic superfluidity in spin-imbalanced lattices. Physica A: Statistical Mechanics and its Applications, 475: 82-87. https://doi.org/10.1016/j.physa.2017.02.013

[11] Alguliev, R.M., Aliguliyev, R.M., Isazade, N.R. (2013). Multiple documents summarization based on evolutionary optimization algorithm. Expert Systems with Applications, 40(5): 1675-1689. https://doi.org/10.1016/j.eswa.2012.09.014

[12] Rautaray, J., Panigrahi, S., Nayak, A.K. (2024). Single document text summarization based on the modified cat swarm optimization (Mcso) algorithm. Research Square. https://doi.org/10.21203/rs.3.rs-3936341/v1

[13] Muddada, M.K., Vankara, J., Nandini, S.S., Karetla, G. R., Naidu, K.S. (2024). Multi-Objective ant colony optimization (MOACO) approach for multi-document text summarization. Engineering Proceedings, 59(1): 218. https://doi.org/10.3390/engproc2023059218

[14] Sharma, G., Sharma, D. (2022). Automatic text summarization methods: A comprehensive review. SN Computer Science, 4(1), 33. https://doi.org/10.1007/s42979-022-01446-w

[15] Rautray, R., Balabantaray, R.C., Bhardwaj, A. (2015). Document summarization using sentence features. International Journal of Information Retrieval Research, 5(1): 36-47. https://doi.org/10.4018/IJIRR.2015010103

[16] Gordon, M. (1988). Probabilistic and genetic algorithms in document retrieval. Communications of the ACM, 31(10): 1208-1218. https://doi.org/10.1145/63039.63044

[17] López-Pujalte, C., Guerrero-Bote, V.P., de Moya-Anegón, F. (2003). Order-based fitness functions for genetic algorithms applied to relevance feedback. Journal of the American Society for Information Science and Technology, 54(2): 152-160. https://doi.org/10.1002/asi.10179

[18] Cordón García, O., Moya Anegón, F.D., Zarco Fernández, C. (2000). A GA-P algorithm to automatically formulate extended Boolean queries for a fuzzy information retrieval system. Mathware & Soft Computing, 7(2-3): 309-322.

[19] Salman, Z.A.W. (2023). Text summarizing and clustering using data mining technique. Al-Mustansiriyah Journal of Science, 34(1): 58-64. http://doi.org/10.23851/mjs.v34i1.1195

[20] Abo-Bakr, H., Mohamed, S.A. (2023). Automatic multi-documents text summarization by a large-scale sparse multi-objective optimization algorithm. Complex & Intelligent Systems, 9(4): 4629-4644. https://doi.org/10.1007/s40747-023-00967-y

[21] Yadav, M. (2023). Summarization of text using natural language processing. International Journal of Scientific and Research Publications, 13(4): 68-72. https://doi.org/10.29322/ijsrp.13.04.2023.p13611

[22] Yang, X.S. (2010). A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010), Berlin, Heidelberg: Springer Berlin Heidelberg, Springer, Berlin, Heidelberg, pp. 65-74. https://doi.org/10.1007/978-3-642-12538-6_6

[23] Prakash, S.J., Chetty, M.S.R., Aravapalli, J. (2022). Swarm based optimization for image dehazing from noise filtering perspective. Ingénierie des Systèmes d'Information, 27(4): 653-658. https://doi.org/10.18280/isi.270416

[24] Alguliev, R.M., Aliguliyev, R.M., Mehdiyev, C.A. (2011). Sentence selection for generic document summarization using an adaptive differential evolution algorithm. Swarm and Evolutionary Computation, 1(4): 213-222. https://doi.org/10.1016/j.swevo.2011.06.006

[25] Alguliev, R.M., Aliguliyev, R.M., Hajirahimova, M.S., Mehdiyev, C.A. (2011). MCMR: Maximum coverage and minimum redundant text summarization model. Expert Systems with Applications, 38(12): 14514-14522. https://doi.org/10.1016/j.eswa.2011.05.033