Nature-Inspired Optimization Techniques for Extractive Multi-Document Summarization: A State-of-the-Art Survey

Nidhi A. Suthar[1], Heena K. Patel[2]

Master's Degree in Computer Engineering, Sankalchand Patel University, Visnagar, India[1]
Assistant. Professor, Sankalchand Patel College of Engineering, Visnagar, India[2]

nidhisthr611@gamil.com , hkpce_spce@spu.ac.in

Abstract: The rapid growth of digital textual content has made Automatic Text Summarization (ATS) an essential research area within Natural Language Processing (NLP). Among existing approaches, extractive summarization remains widely used due to its computational efficiency and practical applicability. However, conventional heuristic and statistical methods often face challenges related to redundancy, coherence, and contextual relevance. This review presents a comprehensive survey of metaheuristic-based extractive multi-document summarization techniques, with particular emphasis on optimization algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Teaching–Learning-Based Optimization (TLBO), Firefly Algorithm, and Biogeography-Based Optimization (BBO). The paper systematically reviews summarization classifications, traditional statistical and graph-based methods, machine learning approaches, and recent optimization-driven techniques. Benchmark datasets including DUC 2005 and DUC 2006 are discussed, along with evaluation metrics such as ROUGE and BLEU. The reviewed studies demonstrate that metaheuristic-based approaches effectively enhance summary quality by improving relevance, coherence, and scalability across multilingual and domain-specific scenarios. This survey highlights the potential of metaheuristic optimization as a robust and promising direction for future research in large-scale and real-time text summarization systems.

Keyword: Automatic Text Summarization, Metaheuristics, optimization, Extractive Summarization, NLP

I. INTRODUCTION

The exponential growth of digital content across online platforms has resulted in an unprecedented volume of textual data being generated on a daily basis. Communication channels such as email and instant messaging contribute significantly to this growth, with billions of messages exchanged each day, while social media platforms continue to generate vast amounts of short-form and multimedia-rich textual content. Although knowledge-centric platforms such as Wikipedia produce comparatively fewer contributions, they contain information of high semantic value. Collectively, these trends highlight the rapid accumulation of unstructured textual data in the digital ecosystem.

The continuous expansion of textual resources across diverse domains has made it increasingly difficult for users to efficiently locate, analyze, and comprehend relevant information. Manual summarization of large document collections is both time-consuming and impractical, particularly given the scale and speed at which digital content is produced. Consequently, Automatic Text Summarization (ATS) has emerged as a vital research area within Natural Language Processing (NLP), aiming to generate concise and informative summaries that preserve the essential content of source documents.

ATS techniques are generally classified into extractive, abstractive, and hybrid approaches. Extractive summarization focuses on selecting the most relevant sentences directly from the original text, whereas abstractive summarization attempts to generate new sentences that capture the underlying meaning of the content (1). Hybrid approaches combine the strengths of both methods to enhance summary quality. Among these approaches, extractive summarization remains widely adopted due to its simplicity, interpretability, and computational efficiency, particularly in large-scale and multi-document scenarios (2).

Despite their effectiveness, traditional extractive summarization methods based on heuristics or statistical features often suffer from limitations such as redundancy, lack of coherence, and insufficient contextual coverage (3). To overcome these challenges, optimization-based approaches—especially metaheuristic algorithms—have gained increasing attention in recent years. These methods formulate sentence selection as an optimization problem, enabling the generation of summaries that balance relevance, diversity, and coverage more effectively than conventional techniques. This paper presents a comprehensive review of metaheuristic-driven extractive multi-document text summarization methods and highlights their potential to address the shortcomings of traditional approaches.

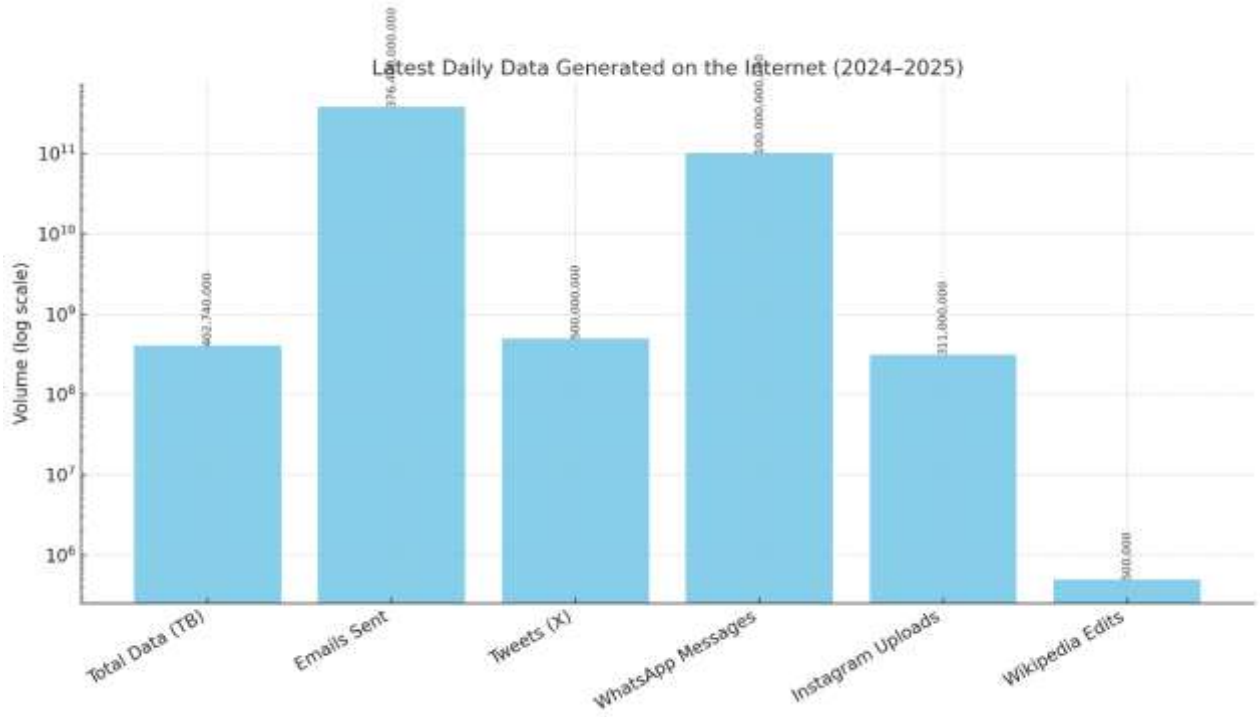

Figure 1. Volume of data created worldwide from 2024 to 2025 [1]

The remainder of this paper is organized as follows. Section 2 presents the classification of Automatic Text Summarization techniques. Section 3 discusses extractive text summarization and existing summarization approaches. Section 4 reviews metaheuristic optimization algorithms applied to extractive summarization. Section 5 describes commonly used evaluation metrics. Finally, Section 6 concludes the paper and highlights future research directions

II. CLASSIFICATION

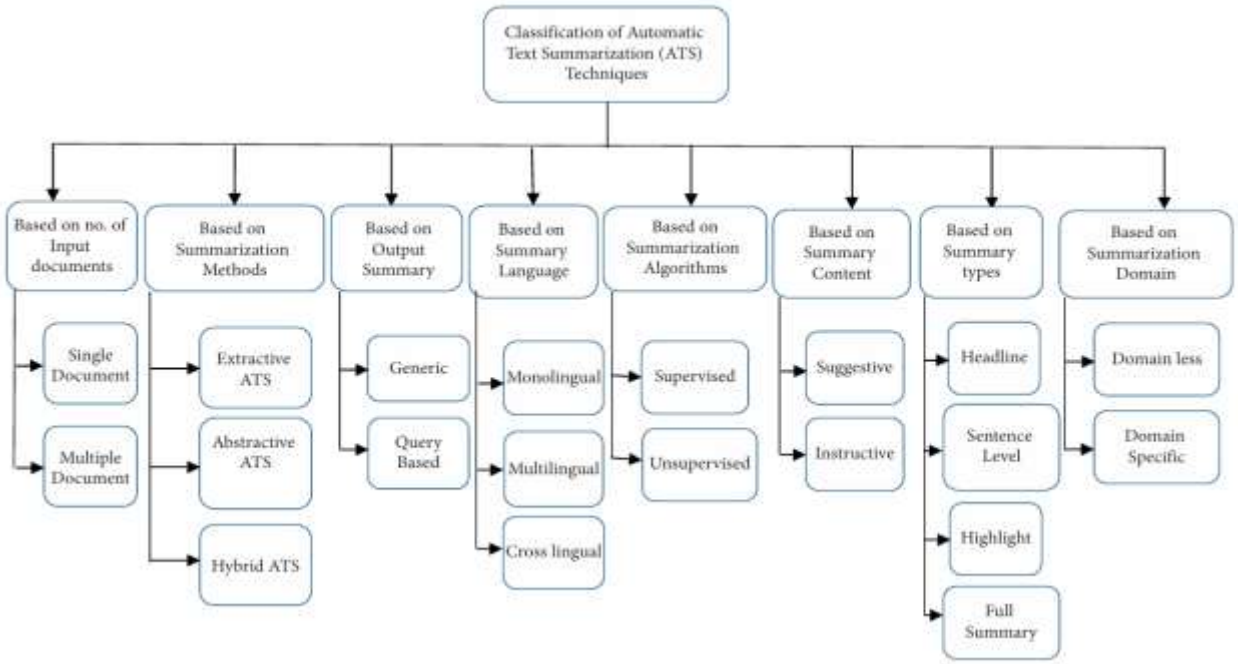

Figure 2. Classification of Automatic Text Summarization [2]

Classification Basis	Categories	Description
Based on Input Size	Single-document, Multi-document	Single-document summarization generates a summary from one document, whereas multi-document summarization produces a summary from a collection of related documents [4].
Based on Summarization Approach	Extractive, Abstractive	Extractive summarization selects important sentences directly from the source text, while abstractive summarization generates new sentences that may not explicitly appear in the original document [4].
Based on Output Summary	Generic, Domain-specific, Query-based	Generic summaries treat all text equally; domain-specific summaries use domain knowledge for improved accuracy; query-based summaries generate content relevant to a user-defined query [5].
Based on Summary Language	Monolingual, Multilingual, Cross-lingual	Monolingual systems produce summaries in the same language as the input; multilingual systems handle multiple languages; cross-lingual systems generate summaries in a different language from the input [6].
Based on Summarization Algorithm	Supervised, Unsupervised	Supervised methods require labeled training data, whereas unsupervised methods operate without human-annotated datasets [6].
Based on Summary Content	Indicative, Informative	Indicative summaries provide a general overview of the document, while informative summaries include all key points from the original text [7].
Based on Summary Type	Headline, Sentence-level, Highlight, Full summary	Headline summaries are very short; sentence-level summaries produce a single sentence; highlight

		summaries use bullet points; full summaries depend on the required compression ratio [8], [9].
Based on Summarization Domain	Domain-agnostic, Domain-specific	Domain-agnostic approaches work across multiple domains, whereas domain-specific approaches are tailored to texts from a particular domain [3].

III. EXTRACTIVE TEXT SUMMARIZATION

An automatic extractive text summarization method involves creating a summary by combining the most important sentences from the given document or set of documents [3]. Compared to abstractive summarization, the extractive approach is simpler and less computationally complex. Since sentences are directly extracted from the source text, the summary retains the original wording, which improves accuracy and reduces the risk of semantic distortion [10]. The extractive text summarization process generally consists of three main stages: pre-processing, processing, and post-processing, as illustrated in Figure 3 [3].

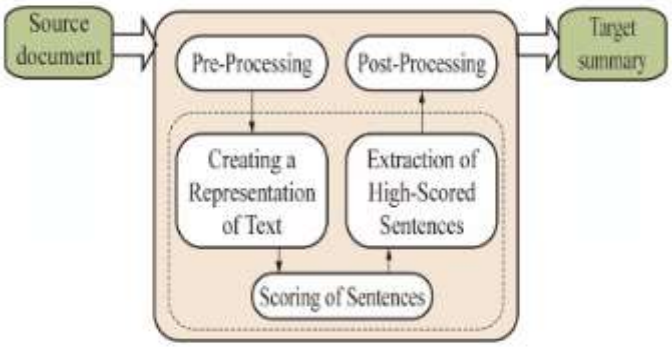

Figure 3: The architecture of an extractive text summarization [4]

Pre-Processing: Pre-processing aims to convert the original text into a structured representation. This stage typically includes: a) Sentence boundary detection, where punctuation marks such as full stops are used to identify sentence endings in English text. b) Stop-word elimination, which removes commonly occurring words that carry little semantic meaning and do not contribute useful information to the summarization task. c) Stemming, which reduces words to their root or base form in order to emphasize their underlying meaning [11].

Processing: The processing stage identifies and computes the features that influence sentence importance. These features are then assigned appropriate weights using a weight-learning approach. Based on the feature–weight formulation, a score is calculated for each sentence. Sentences with the highest scores are selected for inclusion in the final summary [11].

Post-processing: Post-processing involves organizing the selected sentences into a logical and coherent order. This stage also addresses linguistic issues such as redundancy and anaphora resolution to improve the overall readability and coherence of the generated summary.

Challenges: The following are some Challenges of the extractive summary [11]:
The extracted sentences are longer compared to the normal length. As a result, the segments' irrelevant portions are also extracted, taking up space.
Extractive summaries do not capture important or useful information that is often dispersed throughout sentences (barring the summary being long enough to encompass all those sentences).
There can be errors in the presentation of conflicting information.

IV. EXISTING TEXT SUMMARIZATION APPROACHES

In this subsection, a brief description of existing text summarization methods is provided. These approaches can be applied individually or in combination to enhance the functionality of extractive text summarization systems. The selection of a particular method depends on factors such as document size, the level of summarization required, and sentence complexity.

Statistical-Based Methods: Statistical-based methods identify significant terms and phrases from the source text through statistical analysis of various features. In these approaches, terms that are more frequent, appear in advantageous positions, or exhibit higher statistical importance are considered more significant for summarization purposes [11].

Graph-Based Method: Graph-based approaches are unsupervised techniques that rank keywords, sentences, or phrases using graph representations. In this method, documents or sentences are represented as nodes, and edges connect pairs of nodes that share common information or semantic similarity. The importance of each node is determined based on its position and connectivity within the graph structure [12].

Concept-Based Method: Concept-based extractive text summarization focuses on identifying important concepts or ideas present within a document. These concepts are then used to select representative sentences that best convey the main information of the text [3].

Topic-Based Method: The objective of topic-based extractive text summarization is to reduce document length by identifying and emphasizing the major topics or themes present in the text. Sentences associated with dominant topics are selected to form the summary [3].

Semantic-Based Method: Semantic-based approaches consider the semantic relationships between words when computing document or sentence similarity. These methods focus on word meanings and the implicit semantic connections that exist between words, and consequently between sentences and documents, rather than relying solely on surface-level statistical features [13].

Sentence Centrality Based Method: Sentence centrality-based methods identify key and critical sentences within a document or document cluster by determining the importance of words and sentences relative to the overall content. A common technique involves computing the centroid of a document cluster in a vector space and selecting sentences that are closest to this centroid [3].

Machine-Learning Based Method: Machine learning-based approaches formulate text summarization as a supervised classification problem at the sentence level. A trained model classifies sentences as either “summary” or “non-summary” based on learned features derived from annotated training data [3].

Deep-Learning Based Method: Deep learning-based methods utilize advanced neural architectures for sentence representation and summarization. Models such as Long Short-Term Memory networks (LSTMs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models, including BERT and GPT, have been widely adopted for extractive summarization tasks.

Optimization based: Optimization-based approaches treat multi-document text summarization as a multi-objective optimization problem, where multiple criteria such as relevance, coverage, redundancy, and coherence are optimized simultaneously. Various optimization techniques have been proposed to effectively address these objectives in extractive summarization systems [14].

V. LITERATURE

Text summarizing was first attempted in the 1960s and 1970s, mostly with rule-based systems that choose sentences according to pre-established standards such sentence length, keyword frequency, and document position. Scholars started investigating statistical methods for text summarization in the 1990s. In the In text summarization, supervised learning approaches became popular in the 2000s and 2010s. Researchers applied machine learning methods such as decision trees and support vector machines (SVMs) to analyze how sentences are classified as important or not for summarization. More sophisticated and context-sensitive summarization systems are currently made possible using recurrent neural networks (RNNs), transformer architectures such as BERT and GPT, and convolutional neural networks (CNNs). late 1990s and early 2000s, the graph-based text summarization methods took center stage. Techniques such as TextRank, inspired by Google PageRank, modeled documents as phrase graphs with semantic similarity edges. The concepts of TextRank were expanded to document summarization with the introduction of LexRank by Erkan and Radev in 2004. It achieved better summarization performance by measuring the semantic relatedness and significance of sentence vectors using cosine similarity.

Table 2: Comparison of extractive text summarization-based methods

Sr. No.	Title / Study	Technique	Approach	Challenges Addressed	Future Scope
1.	Extractive Text Summarization using Metaheuristic Approach [15]	Genetic Algorithm (GA),	ROUGE-1 = 0.3843, ROUGE-2 = 0.2584.	Processing of large & complex data sets (news corpus, Indian languages), Enhancing precision and recall in the summaries	Scale to more low-resource languages, Real-time applications of summarization for news and social media
2.	Metaheuristic optimization Using Sentence Level Semantics for Extractive Document Summarization [16]	Genetic Algorithm (GA),	ROUGE-1 ≈ 0.55, ROUGE-2 ≈ 0.27	Redundancy in summaries, Balancing relevance and diversity, Preserving semantic meaning other than through statistics	Incorporate with neural embeddings (word2vec, BERT) for more semantically- rich features, Experiment on multilingual datasets
3.	Summarization of scholarly articles using BERT AND BiGRU: deep learning based extractive approach [17]	BERT and BiGRU	ArXiv: ROUGE1 = 46.7, ROUGE2= 19.4, ROUGE-L= 35.4 Pubmed: ROUGE1= 47.0, ROUGE2= 21.3, ROUGE-L= 39.7	BERT for the extraction of information from lengthy academic texts. The suggested method involved incorporating BiGRU on top of BERT to more effectively capture the global context of the papers as well as assimilating BERT to handle lengthy texts.	For long sequences, the learning of GRU is very difficult

4.	Greedy optimization Method for Extractive Summarization of Scientific Articles [18]	Greedy algorithm along with VNS	ArXiv: ROUGE1=0.43, ROUGE2=0.12 Pubmed: ROUGE1 =0.40, ROUGE2=0.13	Two methods were compared to achieve the best ROUGE score: VNS technique and a Greedy algorithm. Despite similar performance, the Greedy approach demonstrated faster completion times.	Greedy algorithms are not suitable for solving complex optimization problems
5.	A Hybrid Text Summarization Approach Using Neural Networks and Metaheuristic Algorithms [19]	Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)	ROUGE-1 = 0.612, ROUGE-2 = 0.417, and ROUGE-L = 0.398.	Pre-processing, Generation of Summary, Feature Extraction	Real-time Summarization, Scalability
6.	Extractive single document summarization using multi-objective modified cat swarm optimization approach: ESDS-MCSO [20]	cat swarm optimization	ROUGE-1 = 0.5324, ROUGE-2 = 0.3126	Redundancy, Coherence, Coverage Balance	Multi-document Summarization, Real-time Applications
7.	Text Summarization Using Swarm-Based and Genetic-Based Methods in Natural Language Processing: A Comparative Review [21]	Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO)	ROUGE-1 score of 0.58	Redundancy, Coherence, Evaluation Metrics	Multi-document Summarization, Real-time Summarization

The diversity of approaches, tactics, and challenges discussed in the current study is reflected in the comparative table of extractive text summarizing methodologies. Genetic Algorithm (GA), BERT-based hybrid models, greedy optimization, and swarm-based approaches such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) were some of the various algorithms that were evaluated. RoUGE-1 and RoUGE-2 scores for genetic algorithms, which are often employed to optimize summary quality and feature selection, were between 0.43 and 0.52 and 0.25 and 0.27, respectively. This reflects consistent but moderate performance for texts that are domain-specific or complex. BERT and hybrid models, in contrast, delivered stronger results for scholarly article summarization, with ROUGE-1 scores reaching 0.61 and ROUGE-L up to 39.7, particularly excelling in maintaining cohesion and coverage with large, multi-topic papers.

Redundancy and coherence are managed efficiently by swarm-based methods such as PSO and ACO, which according to reports attain ROUGE-1 scores of around 0.58 and are keen on evaluation metrics constructed for multi-document summarizing purposes. Greedy optimization methods, especially those augmented by variable neighborhood search (VNS), also performed competitively, though their scalability for highly complex optimization scenarios remains an area for future work. Across all approaches, the table highlights ongoing challenges, including processing lengthy or multilingual documents, minimizing redundancy, presenting nuanced information, and adapting models for broader domains or real-time applications. Researchers suggest that advancing scalability and integration with sophisticated models like BERT, as well as developing efficient solutions for hybrid and multi-document settings, will be pivotal for future growth in this field.

VI. OPTIMIZATION ALGORITHM

6.1 TLBO: Inspired by nature, Teaching-Learning-Based optimization, or TLBO, is an optimization algorithm that tends to imitate the learning and teaching process happening in a class. It is categorized into two steps: the teacher step and the student step.

Teacher phase: It is a mark of an effective teacher to elevate their students' knowledge level to their own level. However, in actuality, this is not feasible, and a teacher's capacity to raise the class mean is limited by the students' abilities [27].

Here, Eq. (9) is employed to reweight the text features' weights with the mean value of the text features and the given teacher solution.

The most frequent sentence in the documents is chosen to be a teacher, and they subsequently work hard to raise the class's general level.

$X_{new} = X_{old} + rand\ (X_{teacher} - T_F.Mean)$ ……eq(1)

where X_{new} and X_{old} stands for the candidate solutions both before and after updating, or the individual's positions before and after learning. The position of the teacher, who is the best member of the population, is represented by $X_{teacher.}$ The mean is the average search agents' level of the population. The mean is the average level of the population of search agents. "rand" is a random number between o and 1, and "TF" is a teaching factor that regulates the variation in the mean value. TF, a heuristic step chosen randomly with an equal probability, is either 2 or 1, where

$TF = round(1+ rand(0,1)\{2-1\})$……eq(2)

Learner phase: Along with learning new information from the instructor, students can increase their knowledge by interacting with others. A student can arbitrarily take up information from a higher-graded other people throughout their mutual learning process. Take into consideration that the randomly chosen answer for each solution X_q is X_q.

After evaluating both of the solutions, X_q and $X_{q'}$, if $X_q < X_{q'}$, then X_q is modified in the manner described below:

$New\_X_{q,dim} = X_{q,dim} + r_z(X_{q,dim} - X_{q',dim})$ ……eq(3)

New X_q stands for the new solution for q^{th}. Otherwise, update the X_q:

$New\_X_{q,dim} = X_{q,dim} + r_z(X_{q',dim} - X_{q,dim})$ ……eq(4)

The X_q will not be updated if $X_q = X_{q'}$

6.2 Binary: Although there are several transfer functions available, the sigmoid function is most frequently employed and produces promising results. Eq. (5) present the sigmoid function [23].

$X_{sig} = 1 / (1 + e^{-X^{(t+1)}})$ ……eq(5)

6.3 Biology Migration Algorithm (BBM): It is motivated by biological phenomenon of many species migrating, including fish, insects, and others. The biological species (particles) in BMA are the population seeking solutions to environmental issues (search space). Migration stage and the update stage are the two stages of the Biology Migration Algorithm (BMA) [23].

Migration phase [23]: At this phase, a species moves its position (from its present location to a new one) based on either of two options: either the best species of the population, as indicated by equations (14) and (15), or its close contenders, as described by equation (16). The neighborhood candidate of each particle is one of the two particles randomly chosen from the population.

$X_i(t+1) = X_i(t) + \lambda * sec(t) * L(t) * | X_{best} – X_i(t) |$ ……eq(6)

where λ is a random vector whose elements are uniformly distributed in [o, 1], $X_i(t)$ and $X_i(t+1)$ are the i^{th} particle positions at iteration t and t+1, L(t) is step size calculated by Eq. (15), and X_{best} is optimal position at the iteration.

$L(t) = 2 – 1.7 ((t-1)/(T-1))$ ……eq(7)

where t is the current iteration and T is the maximum number of iterations.

$X_i(t+1) = X_i(t) + c * (X_j(t) – X_k(t)), i \neq j \neq k$ ……eq(8)

Updating Phase [23]: The updating stage compares the new location, $X_i(t+1)$, which was achieved by finishing the migrating stage with the current location, $X_i(t)$. For searching other search areas, a new position will be utilized randomly if After a number of cycles C, the quality of the current solution cannot be enhanced by the new location $X_i(t+1)$.

6.4 FIREFLY: Firefly is a metaheuristic algorithm that takes inspiration from fireflies' flashing habits [24][28]. It may be employed to summarize text by placing it in a mathematical paradigm.

The algorithm was built using the following three rules:
Since all fireflies are unisex, any brighter firefly can attract the attention of any other firefly.
The objective function is utilized to calculate the brightness of the firefly.
Brightness and attractiveness are proportionally related, and both diminish with an increase in distance, This suggests that a firefly will migrate in the direction of the brighter one, and that it will traverse randomly around the area in the absence of a brighter one.

Firefly Attractiveness: It is clear that the square of the distance, let's say r, from the source, determines the inverse relationship between light intensity and distance. Thus, we can define the attraction variation β with distance r by

$\beta(r) = \beta_0 e^{-\gamma r^2}$……eq(9)

where 0 is the attractiveness at r = 0 and the is Absorption coefficient. Most of the application 0 = 1 can be used.

Firefly Movement: The firefly at X_i will travel toward X_j if it finds the firefly at X_j to be attractive than the one at X_i.

$X_i^{(t+1)} = X_i^t + \beta_0 e^{-\gamma r_{ij}^2} (X_j^t – X_i^t) + \alpha_t \epsilon_i^t$ ……eq(10)

Where $X_i(t+1)$ is new position $X_i(t)$ is old position. The absorption coefficient in firefly optimization is a parameter that controls how quickly the brightness of fireflies decreases with distance. A higher absorption coefficient means that the brightness of fireflies decreases more quickly, which can lead to faster convergence. If is very large the e^{-r^2} term become 0, and If is very small the e^{-r^2} term become 1. The t is being randomization parameter $0 \leq t \leq 1$ and it is a vector of random integers at time t that is taken from a uniform, Gaussian, or other distribution.

VII. Evolution Metrics

It is often evaluated based on the quality of generated summaries using a set of evaluation indicators. These measures are used to determine how effectively the summary conveys the principal concepts contained in the source documents or text.

7.1 ROUGE: ROUGE is an acronym for recall-oriented understudy for gisting evaluation. It compares a set of human-created reference summaries with machine summaries [28]. Some unique ROUGE measures are as follows:

7.1.1 ROUGE-1: The unigram, or ROUGE-1, is employed to measure the divergence between a candidate summary and a reference summary [29]. The n-gram recall between a reference summary and a candidate summary is referred to as ROUGE-N.

Precision (P): As a proportion of all generated summary unigrams, Precision indicates how many overlapping unigrams between a generated summary and a reference summary exist. You can calculate it with the given formula:

P = Number of overlapping unigrams / Total no. of unigrams in the generated summary…….eq(11)

Recall(R): The number of overlapping unigrams in the reference summary and the generated summary over all the unigrams in the reference summary is calculated by recall. It is measured with the following equation:

R = Number of overlapping unigrams / Total number of unigrams in the reference summary ……eq(12)

F1-Score: Precision and recall are merged into a single measure known as the F1 score, which is the harmonic mean of both. It's calculated by the following formula:

F1-Score = (2 * Precision * Recall) / (Precision + Recall) ……eq(13)

Example: The overlap in unigrams between the reference summary and the generated summary is computed by ROUGE-1.
Reference Summary: "Global warming is a pressing issue affecting our planet."
Generated Summary: "The issue of global warming is a concern for our planet."
Overlap Calculation: ROUGE-1 is employed to estimate the overlap of unigrams between the reference summary and the computer summary.
Number of overlapping Unigrams = 5
Total number of unigrams in the generated summary = 11
Total number of unigrams in the reference summary = 9

i. Precision(P):
Precision(P) = Number of overlapping unigrams / Total no. of unigrams in the generated summary
Precis(*P*) = 5/11 ≈ o.4545

ii. Recall(R):
Recall (R) = Number of overlapping unigrams / Total number of unigrams in the reference summary
Recall(R) = 5/9 ≈ o.5556

iii. F1- score:
$F1-Measure$= (2 * Precision * Recall) / (Precision + Recall)
$F1-Measure$= (2 * 0.4545 * 0.5556) / (0.4545 + 0.5556) ≈ o.5ooo

Bigram (two-word) overlap is computed by ROUGE-2, trigram overlap by ROUGE-3, and 4-gram overlap by ROUGE-4. To evaluate the quality of computer summaries or translations, ROUGE-n computes the overlap in n-grams, where an n-gram is an n-word sequence.

7.1.2 ROUGE-L: ROUGE-L(R-L) relies on the length of the longest common sub-sequences between a reference summary and a candidate summary.
Example: A simple example is as follows. ROUGE-L: Let us consider a reference summary and a generated summary of a study on climate change.
Reference Summary: "Global warming is a pressing issue affecting our planet."
Generated Summary: "The issue of global warming is a concern for our plane."
The longest common subsequence is: "global warming is a planet"
Length of the longest common subsequence: 6
Word count of the generated summary = 11
Word count of the reference summary = 9

i. Precision(P):
Precision (P) = Length of the longest common subsequence / Total no. of words in the generated summary
Precision (P)= 6 / 11 ≈ o.5455

ii. Recall(R):
Recall (R) = Number of overlapping unigrams / Total no. of words in the reference summary
$Recall(R)$= 6 / 9 ≈o.6667

iii.F1-score:
$F1 - Measure$ = (2 * Precision * Recall) / (Precision + Recall)
$F1 - Measure$ = (2 * 0.5455 * 0.6667) / (0.5455 + 0.6667) ≈ o.6ooo

7.1.3 ROUGE-S*: ROUGE-S* (R-S*) computes the skip-bigram overlap ratio between a reference summary and a candidate summary.
Example: Here's a simple example. ROUGE-S*: Let us assume that we have a reference summary as well as a generated summary.
Reference Summary: "The cat sat on the mat."
Generated Summary:" The dog sat on the rug."
Assume for the sake of simplicity that skip-bigrams with a maximum skip length of 1 are under consideration.
Total number of skip-bigrams in the reference summary = 4 (The, sat), (cat, on), (sat, the), (on, mat)
Total number of skip-bigrams in the generated summary = 5 (The, sat), (dog, sat), (sat, on), (on, the), (the, rug)
Number of overlapping skip-bigrams = 3 (The, sat), (sat, on), (on, the)

i. Precision(P):
Precision (P) = Number of Overlapping skip bigrams / Total no. of skip bigrams in the generated summary
Precision (P) = 3 /5 =o.6

ii. Recall(R):
Recall (R) = Number of Overlapping skip bigrams / Total number of skip bigrams in the reference summary
$Recall(R)$= 3 / 4 =o.75

iii. F1-score:
$F1 - Measure$ = (2 * Precision * Recall) / (Precision + Recall)
$F1 - Measure$ = (2 * 0.6 * 0.75) / (0.6 + 0.75) ≈ o.6667

7.1.4 ROUGE-SU*: By employing skip bigrams and the unigram as a counting unit, it expands upon ROUGE-S*. The number of skip words is indicated by the "*". For example, ROUGE-SU4 defines that bi-grams can have at most four non-contiguous words between the two bi-gram sentences.
Example: Here is a simple ROUGE-SU* example: Suppose we have generated summary and reference summary of a climate change report.

Reference Summary:
"Global warming is a pressing issue affecting our planet.
Generated Summary:
"The issue of global warming is a concern for our plane."
Let us assume that the ROUGE-SU* implementation is aware of skip-bigrams with a maximum skip value of 2.
Suppose we find the following:
The number of overlapping skip-bigrams and unigrams: 12
The total number of skip-bigrams and unigrams in the generated summary: 25
The total number of unigrams and skip-bigrams of the reference summary: 20

i. Precision(P):

Precision (P) = Number of overlapping unigrams & skip bigrams / Total number of unigrams & skip bigrams in the generated summary
Precision (P) = 12 / 25 = o.48

ii. Recall(R):
Recall (R) = No. of overlapping unigrams & skip bigrams / Total no. of unigrams & skip bigrams in the reference summary
Recall (R) = 12 / 25 = o.6

iii. F1-score:
$F1 - Measure$ = (2 * Precision * Recall) / (Precision + Recall)
$F1 - Measure$ = (2 * 0.48 * 0.6) / (0.48 + 0.6) = o.536

7.2 BLEU [30]: Bilingual evaluation understudy (BLEU), a computer-based evaluation measure, is employed to measure the adequacy of the hypothesis against the reference. For n = 1, 2, 3, and 4, BLEU measures n-gram precision to test the adequacy of word precision and fluency.
Below steps for calculate BLEU [31]:
Precision (P): For a maximum of n, BLEU computes precision for each n-gram, or sequence of n words. To determine the number of n-grams in the candidate translation, tally the number of n-grams in the translation occurring in any one of the reference translations for each size of n-gram.

P_n = Number n-grams in candidate that appear in reference summary / Total no. of n-grams in candidate ……..eq(13)

Brevity Penalty (BP): BLEU penalizes short translations below the references to discourage too-short translations. It is computed by dividing the candidate translation length by the closest reference translation length. There is a penalty if the proposed translation is shorter than the reference; otherwise, there is one.

BP = {1 if candidate length > reference length $e^{1-\text{(reference length / candidate length)}}$ ……..eq(14)

BLEU Score: The geometric mean of each unique n-gram precision, weighted according to its weight, is the BLEU score. Since the weights are usually consistent, every n-gram of accuracy adds the same amount to the final score.

BLEU = BP * exp($\sum^{N}_{n=1} W_n \text{loglog} P_n$)……eq(15)

where W_n is the weight assigned to n-gram precision of size n, and N is the largest n-gram size examined.

VIII. FUTURE SCOPE

To improve how summaries understand context while keeping unnecessary repetition in check, future studies on extractive multi-document summarization should focus on mixed approaches that bring together transformer models like BERT and metaheuristic optimization methods. It's still important to work on making these methods scalable and

less computationally heavy, especially for use in real-time and large-scale situations. Another important direction is to apply these metaheuristic methods to languages with fewer resources and to multiple languages. Also, including summarization goals that are based on user queries and personalized needs could make the system more useful and tailored to individuals. Finally, using more advanced evaluation methods beyond ROUGH, such as those that look at meaning and human feedback, would allow for a better assessment of summary quality.

IX. CONCLUSION

Extractive text summarization based on metaheuristics turns out to be a potent way to deal with the constantly increasing amount of digital data. By leveraging algorithms such as Firefly, Frog Leaping, and Teaching-Learning-Based optimization, these approaches effectively balance relevance, coherence, and coverage while minimizing redundancy. Compared to conventional methods, they demonstrate superior performance on benchmark datasets and hold strong potential for multilingual, domain-specific, and real-time applications. Hence, metaheuristic-driven summarization stands as a promising direction for advancing efficient and intelligent information retrieval in NLP.

REFERENCES

[1] Vilca, G. C. V., & Cabezudo, M. a. S. (2017b). A study of abstractive summarization using semantic representations and discourse level information. In Lecture notes in computer science (pp. 482–490). https://doi.org/10.1007/978-3-319-64206-2_54

[2] El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). Automatic text summarization: A comprehensive survey. Expert Systems With Applications, 165, 113679. https://doi.org/10.1016/j.eswa.2020.113679

[3] Sri, S. H. B., & Dutta, S. R. (2021b). A survey on Automatic text Summarization Techniques. Journal of Physics Conference Series, 2o4o(1), o12o44. https://doi.org/10.1088/1742-6596/2040/1/012044

[4] Patel, V., & Tabrizi, N. (2022). An automatic text summarization: a systematic review. Computación Y Sistemas, 26(3). https://doi.org/10.13053/cys-26-3-4347

[5] Sharma, G., & Sharma, D. (2022). Automatic Text Summarization Methods: A Comprehensive Review. SN Computer Science, 4(1). https://doi.org/10.1007/s42979-022-01446-w

[6] Bhat, I. K., Mohd, M., & Hashmy, R. (2017b). SUMITUP: a Hybrid Single-Document text Summarizer. In Advances in intelligent systems and computing (pp. 619–634). https://doi.org/10.1007/978-981-10-5687-1_56

[7] I. K. Bhat, M. Mohd, and R. Hashmy, "SumItUp: A hybrid single-document text summarizer," Adv. Intell. Syst. Comput., vol. 583, pp. 619–634, 2018, doi: 10.1007/978-981-10-5687-1_56.

[8] F. Dernoncourt, M. Ghassemi, and W. Chang, "A repository of corpora for summarization," in Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC), 2018, pp. 3221–3227.

[9] K. Woodsend and M. Lapata, "Automatic generation of story highlights," in Proc. Annu. Meet. Assoc. Comput. Linguist., Jul. 2010, pp. 565–574.

[10]A. Tandel, B. Modi, P. Gupta, S. Wagle, and S. Khedkar, "Multi-document text summarization – a survey," in Proc. Int. Conf. Data Min. Adv. Comput. (SAPIENCE), 2016, pp. 331–334, doi: 10.1109/SAPIENCE.2016.7684115.

[11]V. Gupta and G. S. Lehal, "A survey of text summarization extractive techniques," J. Emerg. Technol. Web Intell., vol. 2, no. 3, pp. 258–268, 2010, doi: 10.4304/jetwi.2.3.258-268.

[12]A. A. Bichi, R. Samsudin, R. Hassan, L. R. A. Hasan, and A. A. Rogo, "Graph-based extractive text summarization method for Hausa text," PLOS one, vol. 18, no. 5, pp. 1–15, 2023, doi: 10.1371/journal.pone.0285376.

[13]B. Altınel and M. C. Ganiz, "Semantic text classification: A survey of past and recent advances," Inf. Process. Manag., vol. 54, no. 6, pp. 1129–1153, 2018, doi: 10.1016/j.ipm.2018.08.001.

[14]J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "A decomposition-based multi-objective optimization approach for extractive multi-document text summarization," Appl. Soft Comput. J., vol. 91, Jun. 2020,

doi: 10.1016/j.asoc.2020.106231.

[15]D. V. P. Kumar, S. S. Raj, P. Verma, and S. Pal, "Extractive text summarization using meta-heuristic approach," 2022.

[16]P. S. Premjith, A. John, and M. Wilscy, "Metaheuristic optimization using sentence level semantics for extractive document summarization," 2015.

[17]S. Bano, S. Khalid, N. M. Tairan, H. Shah, and H. A. Khattak, "Summarization of scholarly articles using BERT and BiGRU: Deep learning-based extractive approach," Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 9, oct. 2023, doi: 10.1016/j.jksuci.2023.101739.

[18]I. Akhmetov, A. Gelbukh, and R. Mussabayev, "Greedy optimization method for extractive summarization of scientific articles," IEEE Access, vol. 9, pp. 168141–168153, 2021, doi: 10.1109/ACCESS.2021.3136302.

[19]A. M. A. Zeyad and A. Biradar, "A hybrid text summarization approach using neural networks and metaheuristic algorithms," 2023.

[20]A. M. A. Zeyad and A. Biradar, "A hybrid text summarization approach using neural networks and metaheuristic algorithms," 2023.

[21]D. Debnath, R. Das, and P. Pakray, "Extractive single document summarization using multi-objective modified cat swarm optimization approach: ESDS-MCSO," 2001.

[22]A. Abuobeida, "Text summarisation using swarm-based and genetic-based methods in natural language processing: A comparative review," 2025.

[23]D. M. Dunlavy, D. P. o'Leary, J. M. Conroy, and J. D. Schlesinger, "QCS: A system for querying, clustering and summarizing documents," Inf. Process. Manage., vol. 43, no. 6, pp. 1588–1605, 2007.

[24]R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems," CAD Comput. Aided Des., vol. 43, no. 3, pp. 303–315, 2011, doi: 10.1016/j.cad.2010.12.015.

[25]M. Boussalem, S. Aitouche, M. Hamouma, H. Haouassi, and H. Rahab, "BBMA-MDS: Binary biology migration algorithm for multi-document text summarization," Revue d'Intelligence Artificielle, Nov. 2023, doi: 1o.18280/ria.370506.

[26]M. Tomer and M. Kumar, "Multi-document extractive text summarization based on firefly algorithm," J. King Saud Univ. - Comput. Inf. Sci., vol. 34, no. 8, pp. 6057–6065, Sep. 2022, doi: 10.1016/j.jksuci.2021.04.004.

[27]A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm Evol. Comput., vol. 1, no. 1, pp. 32–49, 2011, doi: 10.1016/j.swevo.2011.03.001.

[28]E. Lloret, L. Plaza, and A. Aker, "The challenging task of summary evaluation: An overview," Lang. Resour. Eval., vol. 52, no. 1, pp. 101–148, 2018, doi: 10.1007/s10579-017-9399-2.

[29]N. K. Nagwani and S. Verma, "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm," International Journal of Computer Applications, vol. 17, no. 2, pp. 36–40, Mar. 2011, doi: 10.5120/2190-2778.

[30]N. K. Nagwani and S. Verma, "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm," International Journal of Computer Applications, vol. 17, no. 2, pp. 36–40, Mar. 2011, doi: 10.5120/2190-2778.

[31]N. K. Nagwani and S. Verma, "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm," International Journal of Computer Applications, vol. 17, no. 2, pp. 36–40, Mar. 2011, doi: 10.5120/2190-2778.