J. Integr. Sci. Technol. 2024, 12(6), 834

Journal of Integrated
SCIENCE & TECHNOLOGY

An impact of firefly multi-objective optimization algorithm in the process of text summarization for generation good summaries

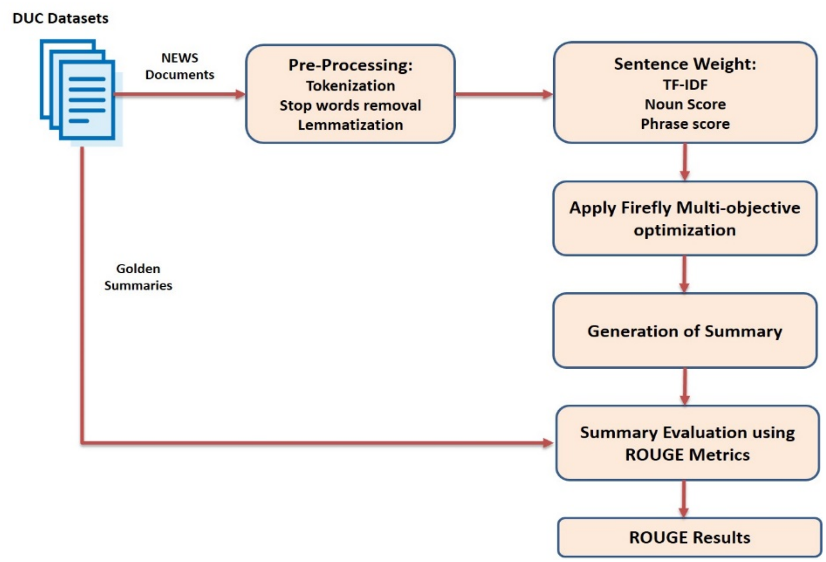
Pravesh Patel*, Paresh Solanki

Faculty of Engineering and Technology, Ganpat University, Gujarat, India

Received on: 18-Dec-2023, Accepted and Published on: 17-Jun-2024

Article

ABSTRACT



The amount of information available on the internet is increasing at a rapid pace. As a result, it is critical that we may quickly and readily get the information we want without having to go through lengthy documentation. Automatic text summarization (ATS) is a technique for producing concise overviews of documents while retaining the most significant information. Everyone wants to finish tasks in the smallest amount of time feasible in today's age of continuously developing technology. The most important lines and topics in the text can be identified, and the amount of text in the summary can be reduced, by using multi-objective optimisation techniques. This makes it easier to make sure the summary is concise and informative while keeping the most important details from the original text. In this research, we integrated effective automated text summarizing strategies for multi-document text summarization based on the firefly multi-optimization algorithm. The proposed algorithm's performance was evaluated using text summarization benchmark datasets from the Document Understanding Conference, namely DUC-2003, DUC-2004, DUC-2005, DUC-2006, and DUC-2007. The ROUGE score is used to assess the generated summaries and also compared with benchmark existing approches of text summarization.

Keywords: Text Summarization, Sentence feature, Multi-objective, firefly optimization, ROUGH score

INTRODUCTION

Text summarization[1] is a subset of Natural Language Processing (NLP) focused on producing concise and informative summaries of text documents.[2] Its primary objective is to extract key information from a longer text, condensing it while retaining the original text's essential meaning and context.[3,4] It is a crucial task in the field of Natural Language Processing (NLP). It involves condensing a longer text into a shorter version, and there are two main approaches to text summarization: extractive and abstractive summarization.[5] The Extractive summarization involves selecting and combining sentences or phrases from the original text to create a summary. It aims to pick out the most important content directly from the source text.[6,7] Abstractive summarization, on the other hand, generates new sentences and phrases that may not be present in the original text but still capture its meaning.[8] This approach requires a deeper understanding of the text, including its context and sentiment.

It is challenging because it demands a comprehensive grasp of the text's content, context, and sentiment. To produce high-quality summaries, NLP models need to identify key information, comprehend the relationships between sentences and ideas, and present the information in a concise and coherent manner. The applications of text summarization are diverse, encompassing news

*Corresponding Author: Pravesh Patel
Email: pravesh.patel@ganpatuniversity.ac.in

Cite as: *J. Integr. Sci. Technol., 2024, 12(6), 834.*
URN:NBN:sciencein.jist.2024.v12.834
DOI: 10.62110/sciencein.jist.2024.v12.834

articles, documents, and legal texts. It's valuable for data retrieval, document management, and textual analysis.[9,10] The primary motivation for text summarization is to alleviate information overload. In today's fast-paced world, people are inundated with vast amounts of information. Summarization helps individuals quickly grasp the essential points and content of a document without having to read it in its entirety.[11] This time-saving process reduces the burden of information consumption in an increasingly overwhelming information landscape.[12]

TEXT SUMMARIZATION CLASSIFICATION

The taxonomy of text summarization is a multifaceted framework that dissects the summarization process into several key dimensions which mention in Figure 1.[13] The method of summary generation distinguishes between extractive and abstractive approaches, with the former selecting key sentences directly from the source text and the latter employing NLP techniques for generating novel summaries.[13,14] The type of input or source classifies summarization as either single document or multi-document, the latter requiring additional considerations like coherence and redundancy. The summary is tailored to a specific query or topic, addressing the user's specific needs.[15] Similar to query-based but focused on summarizing a specific topic within the text. Provides a shorter version of the input text without specific tailoring. Tailored to the preferences and interests of a specific user, often based on their historical interactions or profile.[16] The purpose of summarization can be indicative or informative, depending on whether it merely hints at the topic or provides a more detailed condensed version.[17] Algorithmic approaches involve supervised and unsupervised methods, based on the use of training data. Summarization can vary by language, domain, and user needs, with the potential for query-based, topic-based, general, or personalized summaries.[18,19,20] This taxonomy offers a comprehensive framework to understand and apply text summarization techniques effectively across different contexts. Uses labeled data for training and makes predictions based on it. Aims to uncover hidden patterns and cluster information without prior training data.[20] Summarization input sources can be in different languages, single or multiple languages, or even bilingual. The domain of the text can vary, such as news articles, product reviews, technical papers, and more.[21] This taxonomy provides a comprehensive framework for understanding and categorizing the various dimensions of text summarization, making it easier to analyze, develop, and apply summarization techniques in different contexts. Involves selecting the most informative sentences from the source text and arranging them in order of importance.[22] The summary is essentially an exact copy of the source text.[23,24] Involves using Natural Language Processing (NLP) techniques, such as sentence paraphrasing, to generate a new summary that captures the essence of the text. This method is more complex and time-consuming than extractive summarization. Generates a summary from a single source document. Involves summarizing multiple linked, thematically related documents.[25] It requires considerations such as topic identification, sentence order, coherence, and redundancy removal. Simply indicates the main theme or topic of the text. Provides a condensed version of the source text, giving more detailed information.[26,12]

It is a versatile technique employed in various contexts, such as news articles, product reviews, online forums, language-specific publications, technical and research papers, and other online documents. It is a method used to condense and extract essential information from a given text, making it more accessible and concise. It can be categorized into several types based on the nature of the summary generated. One primary category is General Summarization, where the summary aims to capture the essence of the original content, without being customized for a specific audience or purpose.[27,12] Instead, it provides a broad overview of the key points and content found in the source material.

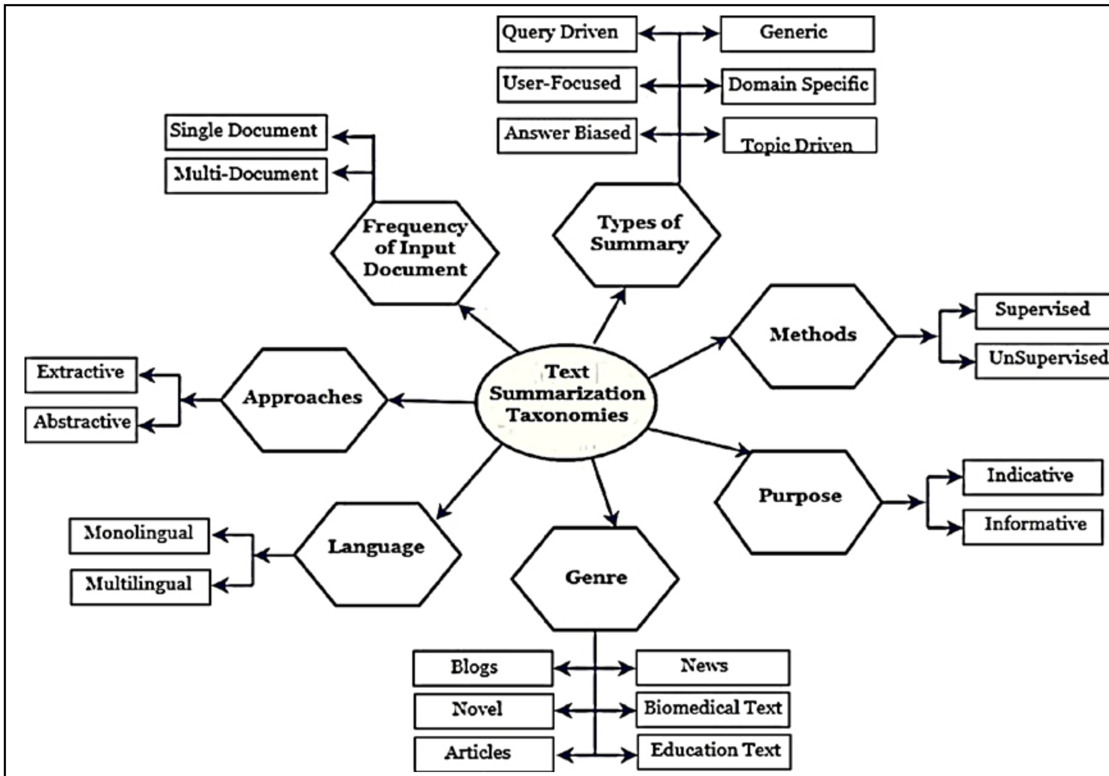


Figure 1. Classification of Text Summarization[13]

Query-based summarization is a method used to generate a summary of a given text or document that is tailored to a specific query or topic[25]. This approach is designed to provide concise and relevant information in response to a particular question or subject of interest. Instead of creating a generic summary of the entire document, query-based summarization focuses on extracting and presenting the most important and contextually relevant facts and details that directly address the query. In essence, this method aims to improve the relevance and specificity of the summary by considering the specific information needs of the user or the context of the query.[28] It can be particularly valuable for information retrieval systems, search engines, and content recommendation algorithms, as it enables them to present users with summaries that directly address their questions or interests. The process of query-based summarization typically involves natural language processing techniques, information retrieval, and semantic analysis to identify and extract the key information related to the query [2029]. This method can be beneficial in various applications, such as search engines, content curation, and question-answering systems, where tailoring the summary to the user's query can enhance the user's experience and access to relevant information.

A meeting summary is a method used to capture and highlight the key actions and decisions made during a meeting. It serves as a valuable tool to help meeting attendees remember the essential topics and outcomes discussed during the meeting. The multi-document summary technique involves creating a concise overview by extracting and combining the most crucial information from multiple text documents. This approach allows for the synthesis of information from various sources to provide a comprehensive and condensed summary of a specific topic. This approach generates a summary that reflects the overall sentiment or emotion expressed in a text[30]. The summary can focus on positive, negative, or neutral sentiments and gives insights into the tone of the text[31]. This approach generates a summary of opinions expressed in a text. The summary highlights the most important opinions and helps readers understand different perspectives[32],[12]. In conclusion, text summarization can be categorized based on the type of summary generated. The choice of summarization method should depend on the specific requirements of the task and the type of text that needs to be summarized. Different situations may call for sentiment summaries to capture the emotional tone of the text or opinion summaries to convey various perspectives.

Related Work

The application of various metaheuristic algorithms in the field of text summarization. These algorithms are used to automatically generate concise and informative summaries from larger text documents. Das and Pal proposed a genetic algorithm-based approach for extractive text summarization[33]. They used a fitness function to evaluate sentence importance and genetic operators like crossover and mutation to generate optimal summaries. Another approach combined genetic algorithms with term frequency-inverse document frequency (TF-IDF) for extractive summarization[34]. Genetic algorithms were used to select the most relevant sentences from the document set. Abed et al. introduced a PSO-based approach for abstractive text summarization[35]. PSO was used to generate optimized word weights for sentence generation, resulting in improved summary quality. Mandal et al. introduced a PSO-based algorithm for multi-document summarization.[33] This algorithm optimized the sentence selection process by considering both the relevance and diversity of the selected sentences. Some researchers applied ACO to the sentence feature extraction stage of text summarization. ACO was used to determine the most relevant sentences based on pheromone trail updates and heuristic information. The FOA was used to select relevant sentences by considering their importance and relationships between sentences.[36] Additional constraints and heuristics were incorporated to improve sentence selection and summarization quality. A modified version of FOA, known as FATS, was tailored specifically for text summarization. Experimental results showed that FATS outperformed traditional methods in terms of summary coherence and relevance. Yang et al. proposed an ensemble approach that combined the firefly optimization algorithm with other techniques for text summarization. HFO integrated genetic algorithms and particle swarm optimization with the firefly algorithm to enhance diversity and convergence in the summarization process, achieving competitive results in summary quality. These works demonstrate the successful application of various metaheuristic algorithms, including genetic algorithms, PSO, ACO, and FOA, in text summarization.[37,38,39] These algorithms help improve summarization performance by enhancing sentence selection, coherence, and relevance in generating concise and informative summaries from multiple documents[40]. The field has also seen the development of specialized algorithms like FATS and HFO to further enhance summarization quality.[41]

Proposed multi document text summarization

A novel framework for automatic extractive text summarization of multiple documents. This framework leverages the Firefly Optimization algorithm shows in Figure 2. The initial step involves extracting textual information from XML documents. These documents may contain a wealth of information that needs to be summarized. Raw text data often requires pre-processing to clean and structure it for further analysis. This step may include tasks such as tokenization, removing stop words, stemming, and handling special characters. To determine the significance of each sentence within the multiple documents, a sentence weighting scheme is applied. Various algorithms, such as TF-IDF, BERT embedding, or other NLP techniques, can be used for this purpose.[42] After assigning importance scores to each sentence, the framework identifies the most relevant and crucial sentences. This step aims to extract key information from the documents while minimizing redundancy. The Firefly Optimization algorithm is employed to assemble the selected important sentences into a coherent and concise summary. Firefly Optimization is an evolutionary algorithm inspired by the flashing behavior of fireflies, and it can be applied to optimize the sentence selection process to create a meaningful summary. For evaluating the effectiveness of our framework, we used the following datasets provided by the National Institute of Standards and Technology (NIST): DUC 2003, DUC 2004, DUC 2005, DUC 2006, DUC 2007.[43] These datasets are valuable resources for research in the field of Natural Language Processing (NLP). NIST has curated news articles from different years, sourced from Newswire and The New York Times. NIST has also furnished golden summaries of these multi-document datasets, making them benchmarks for text summarization research. These datasets serve as the basis for testing and comparing the performance of our summarization framework. In multi-document text summarization framework offers a systematic approach to extract and summarize information from multiple sources, and it is evaluated against NIST-provided datasets to ensure the quality of the summaries generated.[28]

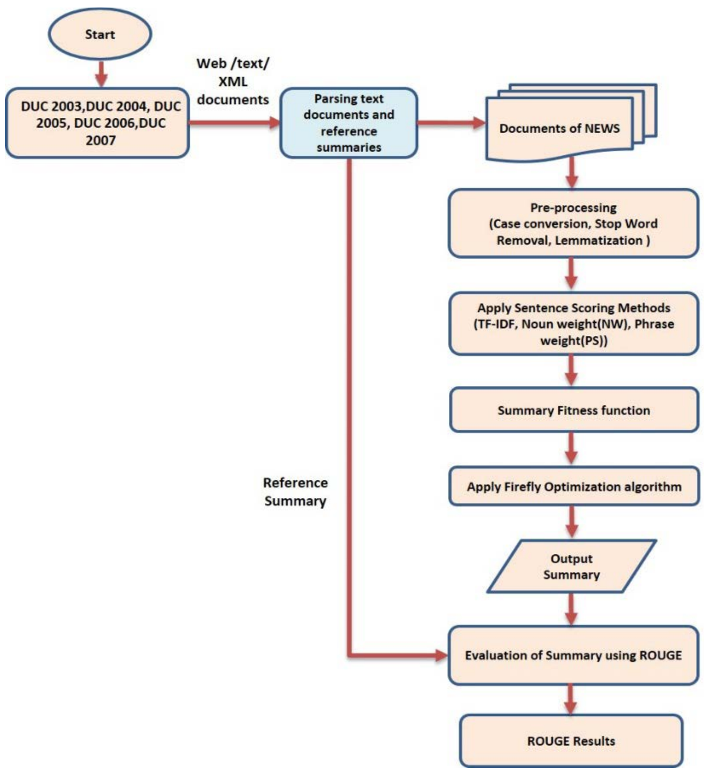


Figure 2. Framework for multi document Text Summarization

This framework holds promise for various applications, including news aggregation, content curation, and information retrieval.

Step 1: Pre - processing

In the initial stage of extractive text summarization, known as pre-processing, several critical tasks are performed to prepare the text data for subsequent summarization techniques. This involves a series of steps, including the conversion of text to a consistent case (typically lowercase) to ensure uniform treatment of words regardless of capitalization. Additionally, irrelevant elements such as hyperlinks and emoticons are removed, while contractions are expanded to their full forms to maintain text integrity. Redundant character repetitions are eliminated, accented characters are replaced with their non-accented counterparts, and spelling mistakes are corrected for accuracy. Common stop words are also removed, and words are lemmatized to their base forms, enhancing the text's suitability for extractive text summarization algorithms. Once this pre-processing is complete, the refined data can be utilized in summarization algorithms, making it ready for the summarization task. Furthermore, the passage alludes to the practice of exporting the data into CSV files, a common approach for further analysis and the application of various algorithmic techniques when working with text data in tasks like summarization and other natural language processing endeavors.

Step 2: Weighting sentences based on their importance

Converting sentences and words into vectors is a fundamental step in numerous natural language processing (NLP) and text analysis tasks, and it plays a pivotal role in extractive text summarization. This process is essential for several reasons. First and foremost, it allows us to translate text into numerical representations, which are the preferred input for machine learning algorithms and statistical models. Text, being inherently non-numeric, needs to be transformed into vectors to be processed and analyzed using mathematical operations. Furthermore, vectorization enables us to capture the semantic and contextual information embedded within the text. This, in turn, empowers algorithms to grasp the relationships between words and sentences and to deduce meaning from the textual data. Consequently, it facilitates various operations that are otherwise impossible with raw text, such as mathematical comparisons (e.g., cosine similarity between vectors) and clustering. These operations are critical for tasks like document retrieval, text classification, and summarization. Several common methods exist for converting text into vectors. One-Hot Encoding, for instance, represents each word or term in a document with a binary vector, where each term is assigned a unique index, and only one element in the vector is set to 1 (indicating the presence of that term) while all others are set to 0. However, it has limitations, particularly with large vocabularies. TF-IDF, on the other hand, assigns weights to terms based on their frequency in a document and rarity across the entire corpus, resulting in a weighted vector representation that reflects term importance. Word embedding, such as Word2Vec, GloVe, and FastText, represent words as dense vectors in a continuous space. These vectors, trained on extensive corpora, capture semantic relationships between words and have become standard in NLP due to their ability to capture meaning and context. Similarly, sentence embedding, generated by techniques like Doc2Vec and Universal Sentence Encoder, represent entire sentences or documents as vectors. This approach is particularly useful for extractive text summarization, as it enables the identification of key sentences that convey the main ideas in the text, facilitating the creation of concise summaries. The choice of vectorization method depends on the specific NLP task, the dataset's size, and the desired level of representation quality. In the context of extractive text summarization, sentence embedding's are often preferred, as they excel at capturing the essential information needed to generate a coherent and concise summary of the source text.

Sentence weights: Assigning relevance to each sentence is a crucial step in extracting important information and creating concise summaries. Sentence weighting involves evaluating the significance of each sentence based on various factors. These factors can include the sentence's length, the frequency of its words, the presence of key terms, and its relevance to a specific topic or query. The goal is to identify and prioritize sentences that are most relevant for summarization. One common technique for determining sentence relevance is by using TF-IDF values.[44] TF (Term Frequency) and IDF (Inverse Document Frequency) are essential statistics used to assess the importance of a term within a specific document and across a collection of documents [45]. By combining TF and IDF, a term's weight can be calculated. This weight reflects the significance of a term within a specific document or a larger text corpus. Sentences containing terms with higher weights are usually given greater importance in the summarization process, as they are likely to convey essential information.

Term Frequency (TF) is a metric that quantifies how often a specific term appears within a document. Typically, TF is computed as the count of occurrences of the term in the document divided by the total number of words in that document.[44]

$$TF_{ij} = {n_{ij}}\Big/{\textstyle\sum_k n_{ij}} \qquad \text{............eq(1)}$$

Where n_{ij} is number of times the word occurs in documents

Inverse Document Frequency (IDF) quantifies how uncommon a term is within a set of documents. It can be computed as the logarithm of the total number of documents in the collection divided by the number of documents containing the term.[44]

$$idf(w) = \log\left(\frac{N}{df_t}\right) \qquad \text{............eq(2)}$$

Where N is total number of documents and df_t is frequency of word in document

The multiplication of the Term Frequency (TF) and Inverse Document Frequency (IDF) values, providing a measure that combines the term's local frequency within a specific document with its global rarity across the entire collection of documents.

$$W_{ij} = TF_{ij} * IDF_i \qquad \text{............eq(3)}$$

$$tf - idf\ score = \frac{w_{ij}}{total\ number\ of\ words\ in\ sentence} \qquad \text{.........eq(4)}$$

Noun weights: The process leverages part-of-speech tagging to identify and prioritize sentences in text based on the number of nouns they contain. It involves tokenization, POS tagging to find nouns, counting nouns in each sentence, and assigning weights. These weighted sentences can then be used for summarization or content prioritization, particularly valuable in applications like news article analysis. The approach offers flexibility for customization, allowing fine-tuning of weight increments and considering noun relevance for more refined results. Overall, it's a powerful tool for automating the extraction of key information from text, greatly aiding natural language processing and content analysis tasks.

Based on phrase score: The process of ranking and selecting phrases from a collection of documents is a highly effective method for distilling essential information. By first identifying frequently used phrases, then narrowing them down based on user-defined criteria, and selecting sentences that encapsulate these key phrases, you can create concise and meaningful summaries or indexes of the content. Assigning scores to phrases and calculating an overall score based on their association with important sentences ensures that the most significant information stands out. This approach is versatile and adaptable to various applications, such as document summarization, keyword extraction, and content identification, making it a potent tool for content analysis and information retrieval, all while allowing customization to meet specific analysis needs.

Step 3: Summary Fitness function

Advanced method designed to enhance the accuracy of generated summaries, closely resembling human-generated counterparts. This function employs three crucial values for evaluating each sentence within a text. Firstly, it leverages TF-IDF (Term Frequency-Inverse Document Frequency), a numerical measure that gauges the significance of words or terms within a document relative to a corpus of documents. In the context of summarization, TF-IDF scores are employed to determine the importance of words within each sentence concerning the entire document. Sentences with higher TF-IDF scores are deemed more vital for summarization. Additionally, Noun Weights (NW) are introduced, assigning values to nouns (and potentially other parts of speech) within each sentence to reflect their relevance. Nouns often carry pivotal information, and sentences with higher noun weights are accorded greater importance in the summarization process. Finally, the summary scoring function incorporates the Phrase Score (PS), evaluating the contextual significance of a sentence within the broader document. This score considers word and phrase relationships and co-occurrences, identifying sentences with a high phrase score as central to the summarization process. By merging these three criteria, the scoring function aims to generate summaries that effectively capture the core information from the source text, mirroring human summarization techniques by emphasizing important words, nouns, and sentence-level relevance.

$$S = \frac{(\alpha * TS-IDF)+(\beta * NW)+(\gamma * PS)}{\alpha+\beta+\gamma} \qquad \text{....eq(5)}$$

In the field of text summarization, the summary function, represented by the parameters α, β, and γ, plays a pivotal role in tailoring the summarization process to meet specific objectives. TF-IDF, a common feature in this context, assists in gauging the significance of individual terms within a document relative to a broader corpus. Alongside this, Newsworthiness (NW) and Position Significance (PS) metrics add layers of evaluation, allowing for the assessment of information importance and the relevance of the sentence's position. By adjusting α, β, and γ, summarization models can be finely tuned to give greater weight to elements like word importance, newsworthiness, or sentence positioning, ultimately enabling the customization of summaries to align with the unique goals and requirements of each summarization task.

Step 4: Firefly optimization based Summary generation

Firefly optimization, inspired by the natural behavior of fireflies, serves as a fascinating tool in solving optimization problems. When applied to the realm of text summarization, this algorithm plays a pivotal role in condensing lengthy texts into concise summaries. The process commences with the definition of an objective function, which quantifies the significance of words or sentences within the text based on various criteria, such as word frequency and keyword importance. Fireflies are then assigned to represent these textual elements, with their brightness indicating their importance score. These digital fireflies move toward brighter counterparts in a bid to optimize their positions, akin to the flow of information. The algorithm continuously updates the brightness of fireflies based on their current positions and their attraction to others. Through an iterative process, it refines their positions until a stopping criterion is met. The final positions of these fireflies denote the selected words or sentences for the summary, revealing the most crucial content, all hinging on the quality of the defined objective function. While firefly optimization offers a unique approach to text summarization, it is essential to acknowledge the effectiveness of the algorithm is heavily reliant on the objective function's ability to capture the essence of textual importance. Various other techniques, including graph-based methods, machine learning, and rule-based approaches, are also prevalent in text summarization, each with its own set of strengths and limitations.

The proposed algorithm for text summarization draws inspiration from the behavior of fireflies to efficiently generate high-quality summaries. In this approach, each firefly is represented as a binary vector, with each element indicating whether a corresponding sentence is included (1) or not included (0) in the summary. The algorithm initiates with a user-defined population of K fireflies, and their "brightness" is determined by a summary fitness function that evaluates the summarization quality. This fitness function incorporates various factors, including TF-IDF values, noun weights, and phrase scores to assess the relevance and coherence of the summary. The brighter fireflies, which represent summaries of higher quality, attract other fireflies, prompting them to adjust their binary vectors to converge toward improved solutions. This iterative process of attraction and movement continues until the fireflies collectively converge on a binary vector that represents the optimal summary. By employing the principles of swarm intelligence, the algorithm leverages the concept of "brightness" to guide the exploration of the solution space, with the ultimate goal of generating a highly effective and coherent summary of the input documents.

Firefly behavior is indeed a remarkable and innovative approach. It successfully combines elements from optimization, natural language processing, and swarm intelligence to offer a unique solution to the challenge of generating high-quality text summaries. The incorporation of a multifaceted fitness function, which considers TF-IDF values, noun weights, and phrase scores, ensures that the algorithm evaluates summaries comprehensively, accounting for both the significance of individual words and the overall coherence of the summary. The concept of attraction and movement within the swarm effectively mirrors principles of swarm intelligence, highlighting the power of collective problem-solving and the emergence of optimal solutions through influence. The iterative optimization process adds further value by allowing the algorithm to gradually adapt and refine summaries, akin to the optimization seen in natural and artificial systems. In conclusion, this approach holds great promise for the field of text summarization and demonstrates how diverse principles can be ingeniously integrated to address complex problems, offering potential applications in various domains where automated summarization is needed.

The Firefly Algorithm for Text Summarization presents an innovative approach that takes inspiration from the intriguing behavior of fireflies to produce concise and informative summaries for multi-document collections. In this unique technique, a population of virtual "fireflies" is initialized, with each firefly corresponding to a sentence extracted from the input documents. The brightness of these virtual fireflies' "light" is determined by a comprehensive function that considers various factors like relevant scores, noun weights, and phrase scores associated with the sentences they represent. Throughout a predefined number of iterative cycles, these virtual fireflies engage in a captivating dance of movement. They continually assess their light intensity in comparison to their peers and make adjustments in their positions, gravitating towards those with stronger illumination. This movement is guided by an attractive coefficient that varies with the distance between fireflies, further enhancing the attraction among closer individuals. In the final ranking stage, the fireflies are ordered based on their light intensities, identifying the most captivating ones. The best firefly, symbolizing the most promising summary, is continually updated to represent the distilled information effectively. The ultimate summary is then crafted by selecting unique, high-intensity fireflies while ensuring it adheres to the desired length or word count constraints. These generated summaries are stored in their respective directories. To assess the quality of these summaries, they undergo evaluation using ROUGE score metrics, which compare them against a reference or "golden" summary. Through this algorithm, the summarization process leverages the fascinating concept of firefly behavior to efficiently distill essential information from multi-document datasets.

EXPERIMENTAL SETUP AND RESULT ANALYSIS

The elucidation of the critical parameters underpinning the algorithm was done with emphasis of their pivotal role in its functioning. The characteristics of the datasets and manual summaries shed light on the nature of the data inputs play a central role in the algorithm's operation. The algorithm outlining was done to leverages the defined parameters. The parameter are mentioned in Table 1 with listing of specific parameters and their values. It is noteworthy that controlled randomness is introduced, a common technique aimed at optimizing convergence.

Table 1 Parameters for firefly based text summarization

Parameter	Values
Population Size	N
Iterations	99
Attractive Coefficient (initial value) β_o	1
Absorption Coefficient (initial value) γ_o	1
Radius for random walk α_o	3
Random walk ε_0	Range (0,1)

Benchmark datasets, named alongside associated years (DUC 2003, DUC 2004, etc.), serve as the testing ground for the algorithm's performance. Detailed dataset characteristics are further encapsulated in Table 2, offering insights into dataset sizes and contents

Table 2 Datasets information

Dataset	No Of cluster	Documents Per Cluster	Golden Summaries	Total Documents
DUC 2007	35	25	4 Human Summaries	875
DUC 2006	50	25	4 Human Summaries	1250
DUC 2005	50	30	4 Human Summaries	1500
DUC 2004	50	10	4 Human Summaries	500
DUC 2003	90	--	4 Human Summaries	--

Finally, the best ROUGE results are presented in Table 3, showcasing the algorithm's achievements in terms of ROUGE scores, with a focus on relevant score, noun weights, and phrase score. Altogether, this section provides a structured and informative account of the algorithm's parameters, data sources, and experimental outcomes, facilitating a comprehensive understanding of its validity and effectiveness.

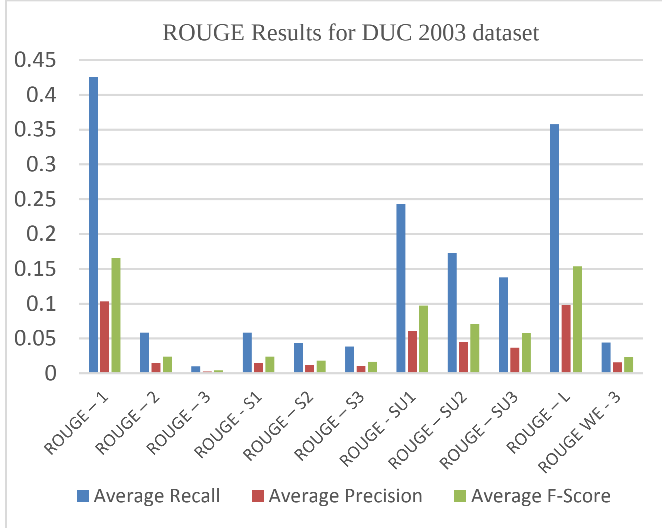
Table 3. Best ROUGE Results

Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.5870	0.1477	0.2357
ROUGE – 2	0.1082	0.0268	0.0429
ROUGE - S1	0.1082	0.0268	0.0429
ROUGE - SU1	0.3606	0.0900	0.1438
ROUGE – SU2	0.2726	0.0674	0.1079
ROUGE – SU3	0.2282	0.0557	0.0894
ROUGE – L	0.5216	0.1501	0.2327

DUC 2003

Table 4. ROUGE Results for DUC 2003 dataset

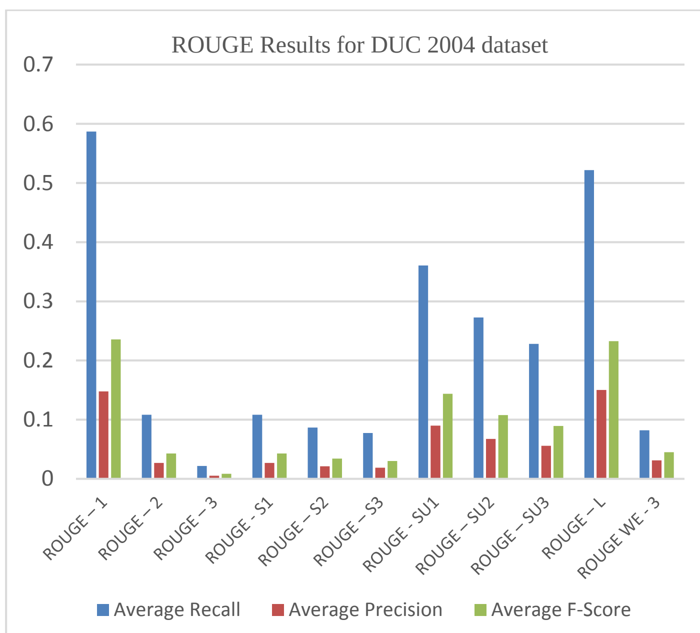
Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.4252	0.1033	0.1658
ROUGE – 2	0.0585	0.0151	0.0240
ROUGE – 3	0.0100	0.0027	0.0042
ROUGE - S1	0.0585	0.0151	0.0240
ROUGE – S2	0.0437	0.0117	0.0184
ROUGE – S3	0.0385	0.0107	0.0167
ROUGE - SU1	0.2434	0.0610	0.0974
ROUGE – SU2	0.1730	0.0448	0.0710
ROUGE – SU3	0.1376	0.0369	0.0581
ROUGE – L	0.3577	0.0981	0.1535
ROUGE WE - 3	0.0442	0.0158	0.0232



DUC 2004

Table 5. ROUGE Results for DUC 2004 dataset

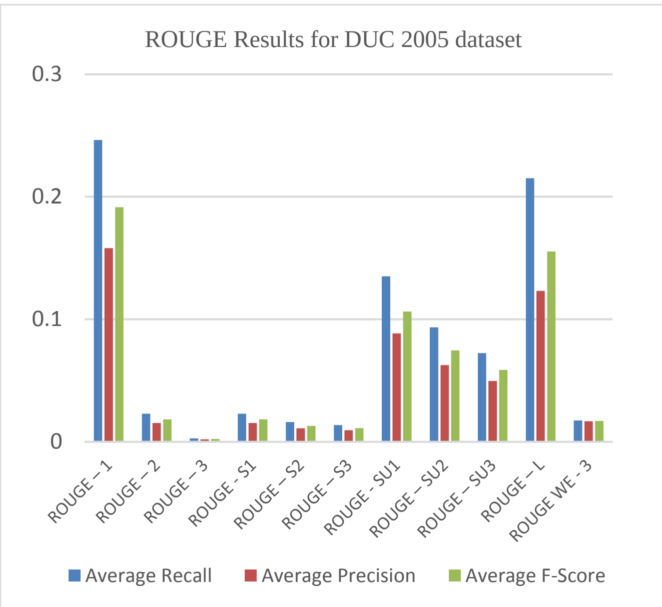
Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.5870	0.1477	0.2357
ROUGE – 2	0.1082	0.0268	0.0429
ROUGE – 3	0.0217	0.0053	0.0085
ROUGE - S1	0.1082	0.0268	0.0429
ROUGE – S2	0.0867	0.0213	0.0341
ROUGE – S3	0.0775	0.0188	0.0302
ROUGE - SU1	0.3606	0.0900	0.1438
ROUGE – SU2	0.2726	0.0674	0.1079
ROUGE – SU3	0.2282	0.0557	0.0894
ROUGE – L	0.5216	0.1501	0.2327
ROUGE WE - 3	0.0821	0.0311	0.0451



DUC 2005

Table 6. ROUGE Results for DUC 2005 dataset

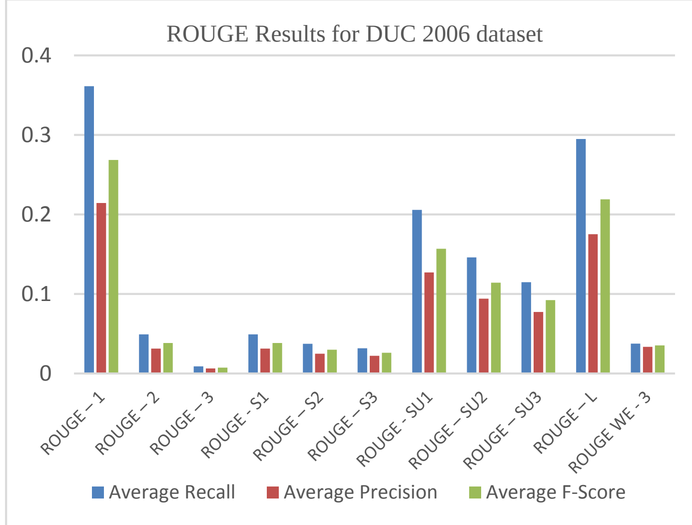
Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.2464	0.1580	0.1915
ROUGE – 2	0.0229	0.0152	0.0182
ROUGE – 3	0.0028	0.0019	0.0023
ROUGE - S1	0.0229	0.0152	0.0182
ROUGE – S2	0.0161	0.0109	0.0130
ROUGE – S3	0.0136	0.0094	0.0111
ROUGE - SU1	0.1350	0.0885	0.1064
ROUGE – SU2	0.0934	0.0626	0.0746
ROUGE – SU3	0.0724	0.0496	0.0586
ROUGE – L	0.2151	0.1232	0.1553
ROUGE WE - 3	0.0174	0.0168	0.0171



DUC 2006

Table 7. ROUGE Results for DUC 2006 dataset

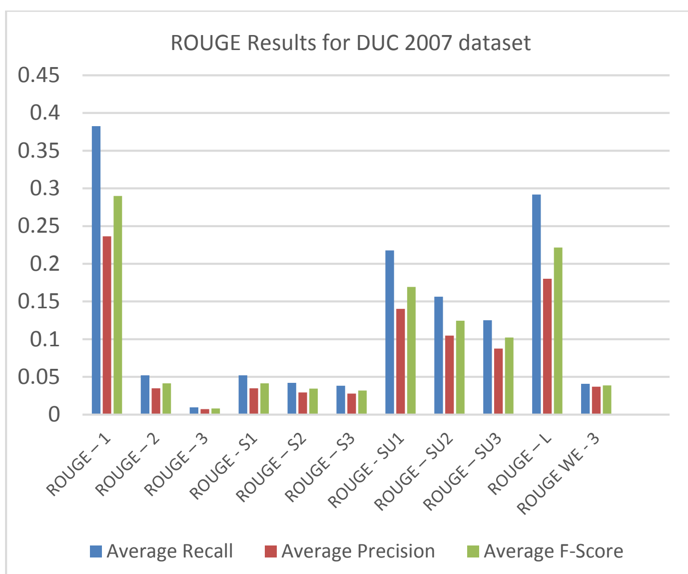
Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.3614	0.2144	0.2685
ROUGE – 2	0.0491	0.0313	0.0382
ROUGE – 3	0.0090	0.0062	0.0074
ROUGE - S1	0.0491	0.0313	0.0382
ROUGE – S2	0.0372	0.0248	0.0298
ROUGE – S3	0.0316	0.0221	0.0260
ROUGE - SU1	0.2058	0.1270	0.1568
ROUGE – SU2	0.1460	0.0940	0.1141
ROUGE – SU3	0.1149	0.0773	0.0922
ROUGE – L	0.2949	0.1751	0.2191
ROUGE WE - 3	0.0374	0.0334	0.0352



DUC 2007

Table 8. ROUGE Results for DUC 2007 dataset

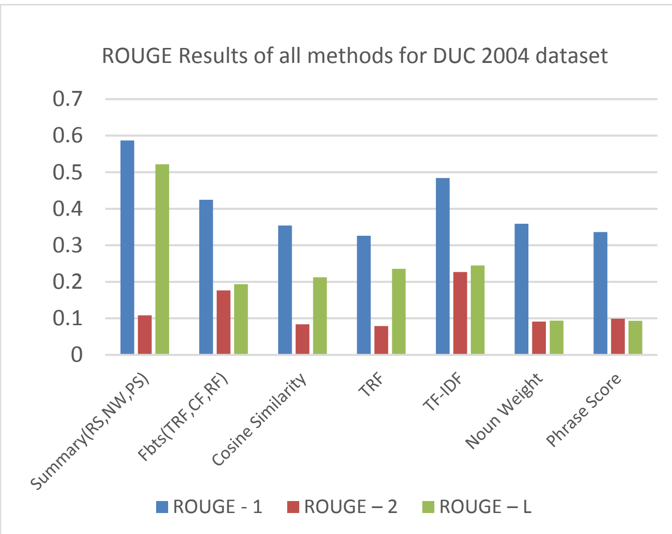
Evaluation Metrics	Average Recall	Average Precision	Average F-Score
ROUGE – 1	0.3824	0.2365	0.2900
ROUGE – 2	0.0521	0.0348	0.0415
ROUGE – 3	0.0097	0.0072	0.0082
ROUGE - S1	0.0521	0.0348	0.0415
ROUGE – S2	0.0421	0.0294	0.0343
ROUGE – S3	0.0382	0.0279	0.0320
ROUGE - SU1	0.2178	0.1401	0.1693
ROUGE – SU2	0.1563	0.1048	0.1245
ROUGE – SU3	0.1251	0.0876	0.1022
ROUGE – L	0.2918	0.1801	0.2216
ROUGE WE - 3	0.0408	0.0370	0.0386



Comparison with other method for the DUC 2004 datasets

Table 9. ROUGE Results of all methods for DUC 2004 dataset

Method	Average Recall	Average Precision	Average F-Score
Summary (RS,NW,PS)	0.5870	0.1082	0.5216
FBTS (TRF,CF,RF)	0.4244	0.1764	0.1934
Cosine Similarity	0.3542	0.0837	0.2122
TRF	0.3260	0.0790	0.2357
TF-IDF	0.4838	0.2269	0.2449
Noun Weight	0.3590	0.0910	0.0937



Conclusion and future work

The analysis provides insights into an ongoing research endeavour aimed at refining text summarization techniques, with a particular focus on the challenges associated with multi-document summarization. The research is currently being evaluated using a News dataset, suggesting a commitment to real-world testing and improvement. The method's adaptability is emphasized, as it employs language-independent features, potentially paving the way for the development of multilingual summarization systems. To enhance summary quality, the approach incorporates WordNet lexical data and semantic aspects, demonstrating a commitment to a deeper understanding of text content. Addressing the challenge of ambiguity in multi-document summarization, the use of tools like lexical chains, morphological analyzers, and statistical-based semantic tools is suggested. Furthermore, bio-inspired optimization algorithms, particularly the firefly optimization algorithm, are harnessed in combination with carefully selected features, such as TF-IDF scores, noun weights, and phrase count, to improve the summarization process. As the research continues, it points towards a promising future that may involve the exploration of novel optimization methods and feature selection techniques to further enhance the quality of extracted summaries.

Acknowledgments

Authors acknowledge the reviewers for their constructive feedback.

Conflict of Interest

There is no conflict of interest as this research was conducted with complete impartiality and without any external influences that could potentially affect the integrity of the findings or interpretation of the results

References

1. P. Janjanam, C.H.P. Reddy. Text Summarization: An Essential Study. In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*; **2019**; pp 1–6.
2. Z. Kerui, H. Haichao, L. Yuxia. Automatic text summarization on social media. *ACM Int. Conf. Proceeding Ser.* **2020**.
3. A.P. Widyassari, A. Affandy, E. Noersasongko, et al. Literature Review of Automatic Text Summarization: Research Trend, Dataset and Method. In *2019 International Conference on Information and Communications Technology (ICOIACT)*; **2019**; pp 491–496.
4. A. Nenkova, K. McKeown. A Survey of Text Summarization Techniques. In *Mining Text Data*; Aggarwal, C. C., Zhai, C., Eds.; Springer US, Boston, MA, **2012**; pp 43–76.
5. M. Gambhir, V. Gupta. Recent automatic text summarization techniques: a survey. *Artif. Intell. Rev.* **2017**, 47 (1), 1–66.
6. S. Buciumas. Reinforcement learning models for abstractive text summarization. *ACMSE 2019 - Proc. 2019 ACM Southeast Conf.* **2019**, 270–271.
7. J.S. Kallimani, K.G. Srinivasa, E. Reddy B. Information extraction by an abstractive text summarization for an Indian regional language. In *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*; **2011**; pp 319–322.
8. A.P. Widyassari, S. Rustad, G.F. Shidik, et al. Review of automatic text summarization techniques & methods. *J. King Saud Univ. - Comput. Inf. Sci.* **2020**.
9. B.T.M. Anh, N.T. My, N.T.T. Trang. Enhanced genetic algorithm for single document extractive summarization. *ACM Int. Conf. Proceeding Ser.* **2019**, 370–376.
10. D. Debnath, R. Das, P. Pakray. Single document text summarization addressed with a cat swarm optimization approach. *Appl. Intell.* **2023**, 53 (10), 12268–12287.
11. P. Verma, H. Om. MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization. *Expert Syst. Appl.* **2019**, 120, 43–56.
12. Z. Xu, J. Zhang, S. Twinandilla, et al. Improving readability through extractive summarization for learners with reading difficulties. *Procedia Comput. Sci.* **2020**, 21 (3), 102536.
13. D. Patel, S. Shah, H. Chhinkaniwala. Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique. *Expert Syst. Appl.* **2019**, 134, 167–177.
14. W. Guan, I. Smetannikov, M. Tianxing. Survey on Automatic Text Summarization and Transformer Models Applicability. *ACM Int. Conf. Proceeding Ser.* **2020**, 176–184.
15. A.R. Mishra, V.K. Panchal, P. Kumar. Extractive Text Summarization - An effective approach to extract information from Text. In *2019 International Conference on contemporary Computing and Informatics (IC3I)*; **2019**; pp 252–255.
16. S. Gupta, S.K. Gupta. Abstractive summarization: An overview of the state of the art. *Expert Syst. Appl.* **2019**, 121, 49–65.
17. Y.K. Meena, D. Gopalani. Analysis of sentence scoring methods for extractive automatic text summarization. *ACM Int. Conf. Proceeding Ser.* **2014**, 11-16-Nove.
18. S. Mandal, G.K. Singh, A. Pal. Single document text summarization technique using optimal combination of cuckoo search algorithm, sentence scoring and sentiment score. *Int. J. Inf. Technol.* **2021**, 13 (5), 1805–1813.
19. M. Joshi, H. Wang, S. McClean. Dense Semantic Graph and Its Application in Single Document Summarisation. In *Emerging Ideas on Information Filtering and Retrieval: DART 2013: Revised and Invited Papers*; Lai, C., Giuliani, A., Semeraro, G., Eds.; Springer International Publishing, Cham, **2018**; pp 55–67.

20. N. Ramanujam, M. Kaliappan. An Automatic Multidocument Text Summarization Approach Based on Naïve Bayesian Classifier Using Timestamp Strategy. *Sci. World J.* **2016**, 2016, 1784827.
21. S. Lamsiyah, A. El Mahdaouy, B. Espinasse, S. El Alaoui Ouatik. An unsupervised method for extractive multi-document summarization based on centroid approach and sentence embeddings. *Expert Syst. Appl.* **2021**, 167, 114152.
22. J. Liu, D.J.D. Hughes, Y. Yang. Unsupervised Extractive Text Summarization with Distance-Augmented Sentence Graphs; Association for Computing Machinery, *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* **2021**, 2313-2317.
23. P. Verma, S. Pal, H. Om. A comparative analysis on Hindi and English extractive text summarization. *ACM Trans. Asian Low-Resource Lang. Inf. Process.* **2019**, 18 (3).
24. K. Al-Sabahi, Z. Zhang, J. Long, K. Alwesabi. An enhanced latent semantic analysis approach for arabic document summarization. *Arab. J. Sci. Eng.* **2018**, 43 (12), 8079–8094.
25. M. Afsharizadeh, H. Ebrahimpour-Komleh, A. Bagheri. Query-oriented text summarization using sentence extraction technique. In *2018 4th International Conference on Web Research (ICWR)*; **2018**; pp 128–132.
26. S. Liu, M.X. Zhou, S. Pan, et al. TIARA: Interactive, topic-based visual text summarization and analysis. *ACM Trans. Intell. Syst. Technol.* **2012**, 3 (2).
27. J. Batista, H. Oliveira, R.D. Lins, et al. A quantitative and qualitative assessment of automatic text summarization systems. *DocEng 2015 - Proc. 2015 ACM Symp. Doc. Eng.* **2015**, 65–68.
28. L. Scanlon, S. Zhang, X. Zhang, M. Sanderson. Evaluation of Cross Domain Text Summarization. *SIGIR 2020 - Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.* **2020**, 1853–1856.
29. L. Marujo, W. Ling, R. Ribeiro, et al. Exploring Events and Distributed Representations of Text in Multi-Document Summarization. *Know.-Based Syst.* **2016**, 94 (C), 33–42.
30. K. Shimada, R. Tadano, T. Endo. Multi-aspects review summarization with objective information. *Procedia - Soc. Behav. Sci.* **2011**, 27, 140–149.
31. P. Gupta, R. Tiwari, N. Robert. Sentiment analysis and text summarization of online reviews: A survey. In *2016 International Conference on Communication and Signal Processing (ICCSP)*; **2016**; pp 241–245.
32. R. Bhargava, Y. Sharma, G. Sharma. ATSSI: Abstractive Text Summarization Using Sentiment Infusion. *Procedia Comput. Sci.* **2016**, 89, 404–411.
33. D. Liu, Y. He, D. Ji, H. Yang. Genetic Algorithm Based Multi-document Summarization. In *PRICAI 2006: Trends in Artificial Intelligence*; Yang, Q., Webb, G., Eds.; Springer Berlin Heidelberg, Berlin, Heidelberg, **2006**; pp 1140–1144.
34. O. Abedinia, N. Amjady, A. Ghasemi. A new metaheuristic algorithm based on shark smell optimization. *Complexity* **2016**, 21 (5), 97–116.
35. H. Asgari, B. Masoumi, O.S. Sheijani. Automatic text summarization based on multi-agent particle swarm optimization. In *2014 Iranian Conference on Intelligent Systems (ICIS)*; **2014**; pp 1–5.
36. R. Saidi, W. Bouaguel, E. Nadia. Hybrid Feature Selection Method Based on the Genetic Algorithm and Pearson Correlation Coefficient. In *Studies in Computational Intelligence*; **2019**; pp 3–24.
37. A.M. Al-Zahrani, H. Mathkour. PSO-based feature selection for arabic text summarization. **2015**, 21, 1454–1469.
38. S. Mandal, G.K. Singh, A. Pal. PSO-Based Text Summarization Approach Using Sentiment Analysis. In *Computing, Communication and Signal Processing*; Iyer, B., Nalbalwar, S. L., Pathak, N. P., Eds.; Springer Singapore, Singapore, **2019**; pp 845–854.
39. S. Gite, S. Patil, D. Dharrao, et al. Textual Feature Extraction Using Ant Colony Optimization for Hate Speech Classification. *Big Data Cogn. Comput.* **2023**, 7 (1).
40. M. Tomer, M. Kumar. Multi-document extractive text summarization based on firefly algorithm. *J. King Saud Univ. - Comput. Inf. Sci.* **2021**.
41. Lucky, A.S. Girsang. Multi-Objective Ant Colony Optimization for Automatic Social Media Comments Summarization. *Int. J. Adv. Comput. Sci. Appl.* **2019**, 10 (3).
42. Y. Liu. Fine-tune BERT for Extractive Summarization. *arXiv preprint*, **2019**, arXiv:1903.10318
43. Document Understanding Conferences https://duc.nist.gov/ (accessed Jun 15, 2022).
44. J.M. Sanchez-Gomez, M.A. Vega-Rodríguez, C.J. Pérez. The impact of term-weighting schemes and similarity measures on extractive multi-document text summarization. *Expert Syst. Appl.* **2021**, 169, 114510.
45. M.A. Ullah, S.M. Marium, S.A. Begum, N.S. Dipa. An algorithm and method for sentiment analysis using the text and emoticon. *ICT Express* **2020**, 6 (4), 357–360.