Received 5 December 2023, accepted 23 December 2023, date of publication 28 December 2023, date of current version 11 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3348075

SURVEY

A Review on Optimization-Based Automatic Text Summarization Approach

MUHAMMAD HAFIZUL H. WAHAB[1], NOR HAFIZA ALI[2], NOR ASILAH WATI ABDUL HAMID[1,2], (Senior Member, IEEE), SHAMALA K. SUBRAMANIAM[1], (Member, IEEE), ROHAYA LATIP[1], (Member, IEEE), AND MOHAMED OTHMAN[1,2], (Senior Member, IEEE)

[1]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia
[2]Institute for Mathematical Research, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

Corresponding author: Nor Asilah Wati Abdul Hamid (asila@upm.edu.my)

This work was supported by the Fundamental Research Grant Scheme awarded by the Ministry of Higher Education, Malaysia, under Grant FRGS/1/2021/ICT08/UPM/02/1.

ABSTRACT The significance of automatic text summarization (ATS) lies in its task of distilling textual information into a condensed yet meaningful structure that preserves the core message of the original content. This summary generated by ATS plays a crucial role in simplifying the processing of textual information, as it captures the primary ideas of the source text while eliminating lengthy and irrelevant textual components. At present, the landscape of ATS is enriched with a multitude of innovative approaches, with a notable focus on optimization-based methods. These optimization-driven ATS techniques have introduced new perspectives, illuminating the field with their heightened accuracy in terms of metrics like ROUGE scores. Notably, their performance closely rivals other cutting-edge approaches, including various methodologies within the realm of machine learning and deep learning. The review presented in this paper delves into recent advancements in extractive ATS, centering mainly on the optimization-based approach. Through this exploration, the paper underscores the gains and trade-offs associated with adopting optimization-based ATS compared to other strategies, specifically with the application of real-time ATS. This review serves as a compass, pointing towards potential future directions that the optimization-based ATS approaches should consider traversing to enhance the field further.

INDEX TERMS Extractive summarization, optimization-based summarization, automatic text summarization, optimization algorithm.

I. INTRODUCTION

Automatic Text Summarization (ATS) is a sophisticated task that involves distilling large amounts of textual information from one or more source documents into a shorter, more concise form. The primary objective is to capture the essence of the original content while preserving its context and meaning. This intricate process is achieved by carefully selecting relevant sentences and eliminating redundant or less significant details. The resulting summary should be coherent and understandable on its own, making it valuable for individuals seeking quick insights without delving into the entirety of the source material.

Scholars such as Widyassari et al. [40], Mahajani et al. [15], and El-Kassas et al. [9] have grouped ATS techniques into distinct categories: abstractive, extractive, and hybrid approaches. These categories reflect the underlying strategies used to create summaries. Abstractive methods aim to generate novel sentences that capture the essence of the original content, even if the exact wording may differ. Extractive methods, on the other hand, directly select sentences or segments from the source text to construct the summary. Hybrid approaches combine elements of both abstractive and extractive techniques to achieve a balance between content preservation and creative expression.

ATS leverages a variety of scientific techniques, including Machine Learning, Deep Learning, Optimization Algorithms, Fuzzy Logic, and Semantic-Based techniques.

The associate editor coordinating the review of this manuscript and approving it for publication was Nikhil Padhi.

Machine learning approaches involve training models on datasets annotated with labels indicating whether a piece of text is a "summary" or "non-summary." However, these methods heavily depend on having access to extensive and accurately labeled data, which can be a limitation. Deep learning techniques, as demonstrated in works by researchers like Yao et al. [41], Nallapati et al. [20], and Warule et al. [39], offer the potential to mimic human reading styles and comprehension. The rise of Large Language Models (LLM) has recently illuminated the landscape of ATS, where LLM has found widespread adoption across various domains, including ATS. The works demonstrated by Tang et al. [51] and Zhang et al. [52] challenge the prevailing sentiment surrounding deep learning-based ATS, showcasing promising summary results. In addition to summarization generation, the work by Gao et al. [59] has also explored the usage of ChatGPT to perform a human-like summary evaluation.

Moreover, newer approaches in LLM, like Zhang et al. [53] and Shi et al. [71], have used ChatGPT and GPT4, respectively, through an iterative prompt method that shows promising results in producing summaries. Lastly, it is noteworthy to highlight that the summaries generated by deep learning-based ATS approach leveraging Large Language Models (LLM) are significantly influenced by the design of the prompts. This influence has been discussed and investigated by Wang et al. [60] and Ma et al. [62].

Optimization-based ATS approaches come with a notable computational cost, as highlighted by El-Kassas et al. [9]. This computational expense demands substantial resources and power, posing challenges to the execution speed and cost-effectiveness of optimization-based ATS approaches. Efforts to mitigate these challenges have led to the parallelization of optimization algorithms, as demonstrated in studies by Sanchez–Gomez et al. [30] and Sanchez-Gomez et al. [31]. While parallelization improves efficiency, achieving real-time application of ATS remains a persistent challenge, as observed in studies including Verma and Om [37], Lovinger et al. [14], Shetty and Kallimani [32], and Srivasta et al. [34]. Recent work by Wahab et al. [63] explicitly addresses critical performance issues in optimization-based ATS, particularly those utilizing an expensive ATS repair operator scheme. The introduction of the decomposition-based multi-objective differential evolution (MODE/D) significantly reduces serial execution time, showcasing advancements beyond prior work by Sanchez-Gomez et al. [31] in generating a 200-word summary from DUC2002. Notably, beyond the focus on the ATS repair operator scheme, this paper aims to comprehensively explore additional performance issues inherent in optimization-based ATS approaches.

The rest of the paper will be organized in the following order: Section II discusses the real-time application of ATS. Next, Section III presents and illustrates the classification of ATS approaches. Section IV highlights ATS approach validation in terms of summary quality and performance. Section V explores optimization-based ATS approaches. Section VI analyses the summary quality and performances of optimization-based ATS approaches. Finally, Section VII concludes the paper and provides the future direction of optimization-based ATS approaches.

II. APPLICATION OF AUTOMATIC TEXT SUMMARIZATION

The relevance of ATS has increased over the years due to the massive amount of textual information generated on the internet, as textual data requires proper summarization techniques to reduce the maintenance effort and necessitates techniques to condense the information into meaningful summaries without losing the original context. Additionally, Mahajani et al. [15] highlighted that developing methods to create digests or summaries from vast textual information has been a forefront area of research, with text summarization being a prime example of such a technique.

ATS also finds applications across various knowledge domains, where using a textual summary instead of the original data can significantly enhance search algorithms. Real-time ATS refers to producing a textual summary within a specified time frame for a given input text size. While the speed of generating a summary is crucial in real-time ATS, it is equally important that the summary's quality meets acceptable standards as outlined in service level agreements (SLAs).

Next, Table 1 outlines the domains where ATS is essential for real-time tasks. Categorizing these domains is crucial because the ATS approach must meet each application's requirements. For instance, an ATS approach utilizing an advanced optimization algorithm that takes several days to summarize multi-document content is impractical for real-time ATS applications. Therefore, ensuring the adequacy of an ATS approach for real-time tasks necessitates a comprehensive performance analysis to verify and validate its reliability. Moreover, Table 1 outlines real-time ATS applications that are not limited to optimization-based ATS but include other scientific-based ATS approaches, such as deep learning-based ATS.

Based on the deliberations presented in Table 1, the significance of real-time ATS is evident across diverse domains, and to be effective in real-time applications, an ATS approach must possess the ability to rapidly generate succinct summaries without compromising precision. Notably, El-Kassas et al. [9], Jani et al. [45], Yadav et al. [46], Goularte et al. [47], and Ma et al. [48] have extensively covered and elucidated domains where real-time ATS applications are not essential. As a result, this discussion will explore the domains left unaddressed by these studies.

III. CLASSIFICATION OF TEXT SUMMARIZATION APPROACHES

In this section, various categorizations of ATS approaches are explored, considering factors such as their capacity to process multiple documents as input, generate novel words not present in the input, adapt to diverse languages, and their

TABLE 1. Real-time application of automatic text summarization.

Domain	Task Description
Media monitoring	In the realm of cybersecurity, the focus is on scraping websites for trends related to security threats and leveraging social media for intelligence-led policing operations, as highlighted by Hassani et al. [73]. The demand for real-time ATS becomes evident in these scenarios. The necessity arises for live textual information to be condensed promptly before analysis, thereby emphasizing the practical application of ATS in media monitoring.
Search marketing and SEO	The domain of Search Engine Optimization (SEO) and search marketing can derive significant advantages from real-time ATS, especially crucial for rapidly scrutinizing search results, making multi-document ATS an essential tool for efficiently sifting through pertinent documents generated from user queries. These advancements' real-time and data-driven characteristics align seamlessly with the evolving landscape of online marketing, emphasizing the need for adaptive strategies to navigate the dynamic environment and enhance overall effectiveness (Jabbar et al.) [74].
Question and answers (Q&A) Bot	A personal digital assistant intervenes in automated customer interactions when customers provide extensive passages for simple instructions. A real-time ATS selectively trims irrelevant details to enhance efficiency before feeding queries into the decision model or prompt on LLMs. A finely tuned prompt design aligns seamlessly with the earlier emphasis on the significance of prompt design for Large Language Models (LLMs) in producing a resilient summary. (Wang et al.) [60] and (Ma et al.) [62]. The significance of real-time ATS is underscored for optimizing both online and archival content. By incorporating swift text summarization techniques, the chatbot produces prompt and concise responses, significantly enhancing the user experience in automated question-and-answer scenarios. Notably, the study by Reddy and Guha [76] explores integrating various summarization techniques in chatbots, aiming to elevate the user's reading experience.
Medical cases	The medical field grapples with information overload due to the digitization of healthcare, continuous updates to online medical repositories, and the escalating availability of biomedical datasets. This inundation poses a challenge for practical data analysis, burdening medical professionals who rely heavily on this data for research and patient consultations. Text-highlighting techniques can capture relevant medical context, alleviating doctors' cognitive load and response time. (Ozyegen et al.) [75]. In telehealth, where medical cases often entail excessive details, a real-time ATS approach would significantly benefit by providing prompt analysis. This streamlined process aligns seamlessly with the overarching goal of improving efficiency and effectiveness in the telehealth domain by facilitating faster decision-making, ultimately enhancing the quality of online medical services
Internal document workflow	Real-time ATS plays a pivotal role in expeditiously classifying newly introduced documents, providing prompt identification. This instantaneous categorization enables seamless integration into the organization's internal archive, streamlining accessibility for individuals requiring organized and tagged documents. The efficiency of real-time ATS accelerates the classification process and enhances the overall accessibility and utilization of archived documents within the organization's framework.
Financial research	The dynamic nature of financial markets, characterized by unpredictable trends influenced by real-time news, demands swift decision-making from financial researchers and analysts. In this context, a real-time ATS approach becomes indispensable, as it consolidates and processes information in real-time from multiple documents across diverse sources. This real-time data consolidation significantly impacts split-second decision-making in financial research, empowering analysts to gain immediate insights into current market trends and make informed decisions crucial for navigating the volatile financial landscape. Notably, Abdaljalil et al. [77] have explored text summarization on finance reports published by British firms on the London Stock Exchange. With the utilization of real-time ATS, these lengthy finance reports can be aggregated and consolidated faster, providing a more efficient means of extracting critical insights for timely decision-making in the financial domain.

algorithmic domain. Herein, delineate the standard classification schemes employed for scientific-based ATS approaches.

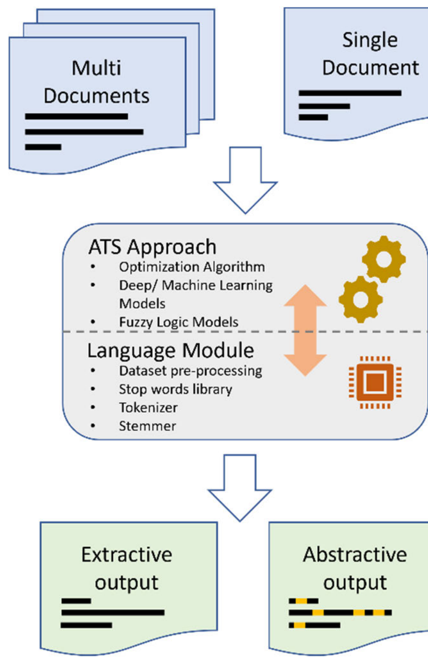

FIGURE 1. The generic ATS system architecture has different input volumes, output novelty, approach domain, and language versatility.

A. GIVEN THE CAPABILITY TO CHANNEL A SPECIFIC VOLUME OF INPUT

In this classification, ATS approaches are categorized based on their capability to process input from a single or multiple document source. When considering the input, an ATS approach categorized as "single-document ATS" can exclusively handle input from a single document source and generate the necessary summary. While this approach is more straightforward than multi-document ATS, its practical applicability in real-world scenarios is limited, given that textual information often originates from multiple documents.

Conversely, an ATS approach categorized as "multi-document ATS" is designed to work with input from more than one document source, and it can also handle a single-document input if needed. However, this versatility comes with challenges, including the need to address redundancy in input from multiple documents and maintain relevance to the main topic of the textual information. Fig. 1. presents the representation of both single-document and multi-document ATS classifications.

Next, Table 3 consolidates the optimization-based ATS approaches with a column labeled "multi-document" (MD) to distinguish the approach specific to single and multi-document summarization. In addition, there are many more recent scientific-based ATS approaches, such as deep learning-based ATS. For instance, the ability of LLM-based automatic text summarization systems to handle both single and multi-document summarization depends on the model's architecture. Some LLMs are designed for single-document tasks, but others, like GPT models based on the Transformer architecture, can be adapted for multi-document summarization. While these models excel at summarizing individual documents, researchers have developed techniques to extend them for multi-document tasks. Multi-document summarization involves considering information from multiple source documents, requiring additional architectural considerations and training process modifications to handle the complexities effectively.

Lastly, Tang et al. [51], Zhang et al. [52], Zhang et al. [53], Guan et al. [54], Bajaj et al. [55], Xu et al. [56], and Joshi et al. [57] are some of the latest works that revolve around the single-document deep learning-based ATS approach whereas Ghadimi and Beigy [58] deals with multi-document deep learning-based ATS.

B. GIVEN THE CAPABILITY TO INTRODUCE NOVELTY IN OUTPUT

Concerning the novelty in generating summary output from a piece of textual information, an ATS approach falls under the abstractive category if it embeds novel words not initially present in the input. Furthermore, these newly generated words are contextually derived from the content of the original input, typically requiring a substantial dataset to generate relevant terms. Conversely, an ATS approach is classified as extractive if it extracts words from the input. In this classification, the ATS approach prioritizes identifying the most critical content within the input. In addition, abstractive text summarization involves generating concise summaries by interpreting and paraphrasing source content using advanced NLP techniques, such as transformer models. It aims to capture essential meaning and can handle diverse writing styles.

In contrast, extractive summarization selects and combines sentences directly from the source text based on certain features like sentence length and term frequency. It prioritizes factual accuracy but may result in less coherent summaries. Both approaches offer unique strengths, with abstractive methods providing human-like summaries and extractive methods preserving the original wording of the source.

The hybrid text summarization approach seamlessly integrates elements from both abstractive and extractive methods to capitalize on their respective strengths. Fig. 1. illustrates the workflow of a hybrid-class ATS approach, showcasing the effective combination of essential features from both categories. An initial extractive step identifies critical sentences or phrases, forming the summary's foundation and applying an abstractive model to enhance coherence. While the hybrid approach aims to balance the factual accuracy of extractive methods and the linguistic quality of abstractive methods, it is essential to note that some ATS approaches in the hybrid class may face challenges due to a lack of clear guidelines on the optimal integration of features from each category. This ambiguity can sometimes result in less precise summaries, highlighting a potential drawback within the hybrid class of ATS methods.

Noteworthy examples of ATS approaches in the hybrid class include the work presented by Sahoo et al. [25], Gehrmann et al. [10], Liu et al. [13], Rudra et al. [24], Alami Merrouni et al. [79], Pei et al. [80], and Ansary [81].

C. GIVEN THE ATS APPROACH'S LANGUAGE VERSATILITY

In addition to classifying ATS approaches based on their summary generation methods and ability to handle different input sizes, another classification criterion relates to their proficiency in handling various languages. The ATS approach's ability to handle various languages leads to a growing interest in language-specific ATS, designed to summarise content in particular languages. Notable examples of this specialization include Arabic language-specific ATS, as demonstrated in the works of Qassem et al. [22] and Al-Radaideh and Bataineh [5]. Besides Arabic, there are also Indonesian language-specific ATS explored by Gunawan et al. [82], Timalsina et al. [83], and Lin et al. [84].

As shown in Fig. 1., language-specific ATS systems require input in a specific language and involve special processing steps on the language-specific module. In this approach, the language module adapts training data representative of the target language and implements language-specific tokenization processes, considering the syntax, grammar rules, and linguistic nuances unique to that language. Fine-tuning the language model on language-specific data and adapting Named Entity Recognition models further contribute to the proficiency of the language-specific ATS technique. Additionally, one of the challenges language-specific ATS faces is the distinct linguistic features and characteristics inherent to each language. Therefore, the language-specific ATS approach should be equipped to handle language-specific ambiguities and idioms. Consequently, an ATS approach tailored to a particular language can incorporate specific linguistic elements and clauses, enhancing the quality of the generated summaries.

D. GIVEN THE CLASSIFICATION OF ATS APPROACH DOMAIN

In this categorization, a systematic classification based on the employed algorithms sheds light on the technical nuances of these approaches. For example, an ATS approach grounded in scientific principles falls under various algorithmic domains, including fuzzy logic, optimization, deep learning, and machine learning. Specifically, ATS approaches that leverage optimization algorithms to formulate summarization as an optimization problem are categorized as optimization-based ATS approaches. Throughout this paper, the focal point of discussion lies in optimization-based ATS, which emphasizes the current landscape of optimization-based ATS. Within this classification, optimization algorithms function as heuristics to transform the words designated for summarization into a format aligning with the parameters and categories of the optimization algorithm. Regardless of the chosen algorithmic domain for ATS, there are discernible advantages and disadvantages, succinctly summarized in Table 3 for further deliberation.

To sum up, ATS approaches can undergo diverse categorizations, encompassing the method they employ to generate summaries, their capacity to handle varying input sizes, their proficiency in handling specific languages, and their classification based on their algorithmic domain.

IV. TEXT SUMMARIZATION APPROACH VALIDATION

To gauge the quality of a summary generated by an ATS approach, it must undergo a standardized evaluation process. Several datasets are available for this purpose, with examples including those from the Document Understanding Conferences (DUC). The DUC dataset comprises extractive and abstractive approach datasets, further differentiating between datasets with sentence segmentation and those with full text without such segmentation. Alongside the DUC dataset, there are other frequently employed datasets like CNN, Daily Mails, and Gigaword; however, recent ATS research predominantly relies on the DUC dataset for evaluation.

Concerning domain-specific datasets, News datasets have consistently been the preferred choice for ATS approaches due to their accessibility. Additionally, datasets from other domains, such as legal, medicine, and finance, are available. Table 2 highlights and lists the datasets available for these mentioned domains.

TABLE 2. Domain-specific dataset.

Domain	Dataset
News	BBC News CNN/Daily Mail Xsum DUC Gigaword
Legal	IN-Abs and IN-Ext from Bhattacharya et al. [85] UK-Abs [88] Greek Case Law Dataset from Koniaris et al. [87] CaseSummarizer Dataset from Polsey et al. [86]
Medicine	PubMed Medical Sum from Michalopoulos et al. [89] MS^2 from DeYoung et al. [90] BioASQ from Nentidis et al. [91]
Finance	UK Annual Report (FNS-News) [77] [92] [93]

A. TEXT SUMMARY QUALITY

In addition to datasets, assessing the quality of the generated summary is a crucial step. The widely adopted tool for this purpose is the ROUGE metric software package, readily available as an open-source tool online. The ROUGE metric software is responsible for systematically evaluating summaries produced by ATS heuristics. This evaluation involves comparing the summary generated by the heuristic with a set of reference summaries created by humans to gauge its quality. Besides ROUGE metrics (Precision, Recall, and F1-score), other metrics are widely used in the literature, such as METEOR (Metric for Evaluation of Translation with Explicit ORdering), BLUE (Bilingual Evaluation Understudy), and CIDER (Consensus-based Image Description Evaluation).

Hence, the quality of a summary generated by an ATS approach is gauged and validated using the ROUGE metrics software tool. Apart from the ROUGE-n score, Sanchez Gomez et al. [29], Sanchez Gomez et al. [30], and Sanchez Gomez et al. [31] propose an alternative measurement method, which includes the Range and Pearson's coefficient of variation (CV). The CV provides another dimension of looking at an ATS approach stability. These measurements are expressed in the general formulas for ROUGE-n (1), Range (2), and CV (3), respectively.

$$\boldsymbol{Rouge-n\,(recall)} = \frac{\boldsymbol{Count_{match}\,(gram_n)}}{\boldsymbol{count\,(gram_n)}} \tag{1}$$

$$\boldsymbol{Range} = \boldsymbol{ROUGE_{best}} - \boldsymbol{ROUGE_{worst}} \tag{2}$$

$$\boldsymbol{CV} = \frac{\boldsymbol{Range}}{\boldsymbol{ROUGE_{average}}} * 100 \tag{3}$$

The Range measurement (2) assesses the variation between the highest and lowest ROUGE scores for a specific topic. A smaller range value signifies less disparity between the topic's best and worst ROUGE metric scores. Conversely, the CV measurement (3) offers a more concise assessment, indicating an ATS approach's consistency across various topics. A smaller CV value suggests that a particular ATS approach exhibits more excellent stability or robustness than other approaches when addressing the same topic. Numerous assessment methods are available when evaluating the quality of summaries generated by ATS approaches. For instance, Steinberger [35] outlines several summary quality evaluation criteria, including grammatical correctness, absence of redundancy, referential clarity, and structural coherence within the realm of text quality evaluation. In terms of content evaluation, various metrics come into play, such as cosine similarity, F-score, recall, unit overlap, and n-gram matching (ROUGE). Notably, works by Sanchez-Gomez et al. [29], Sanchez-Gomez et al. [30], Sanchez-Gomez et al. [31], and Wahab et al. [63] emphasize the importance of non-redundancy as a convergence criterion in conjunction with n-gram matching (ROUGE).

Therefore, it is evident that the quality of summaries produced by ATS approaches can be evaluated using a diverse range of metrics. It is essential to note that an ATS's performance typically refers to the approach's ability to produce summaries according to the mentioned quality evaluation criteria. This performance evaluation does not necessarily pertain to practical aspects like execution time, which will be discussed in Section IV-B.

B. CPU PERFORMANCE MEASUREMENT

In addition to assessing the summary's quality, it is imperative to gauge the performance of an ATS approach. Sanchez-Gomez et al. [30], Sanchez-Gomez et al. [31], and Wahab et al. [63] introduced the use of CPU time or execution time as a measure of the duration required for an ATS approach to generate a summary. Furthermore, they evaluated the parallelization performance of their parallel ATS approach by employing metrics such as the speedup ratio and parallel efficiency. This approach is widely acknowledged and accepted. Consequently, efforts to parallelize an ATS approach should include reporting these performance metrics alongside the summary quality metric, as discussed in Section IV-A.

In addition to assessing the CPU Time of an algorithm used in ATS, an alternative method is proposed by Wahab et al. [63] to provide a more practical measure of an ATS approach's effectiveness in producing a meaningful summary for specific applications. To evaluate an ATS approach based on its capacity to generate a certain number of words within a given time frame, denoted as "n" words in "t" seconds (4) can be used. The WPS (Words Per Second) metric considers the algorithm's processing time, irrespective of its domain (e.g., multi-document, deep learning), and focuses solely on its ability to produce words within a defined time interval measured in seconds.

$$\boldsymbol{WPS} \propto \frac{\boldsymbol{n}}{\boldsymbol{t}} \tag{4}$$

For instance, consider the case of the approach developed by Sanchez-Gomez et al. [31], which requires approximately 2068.70s to generate 200 words for the DUC2002 topic (d070f). This scenario would yield the following computation.

$$\boldsymbol{WPS} = \frac{\mathbf{200}\ \boldsymbol{words}}{\mathbf{2068.70}\ \boldsymbol{seconds}} = \mathbf{0.097}\ \boldsymbol{ws}^{-\mathbf{1}}$$

Finally, apart from utilizing CPU time, speedup, and efficiency as performance metrics, assessing an ATS approach's performance can also include using WPS. This metric provides a more pertinent evaluation, particularly concerning the approach's effectiveness in delivering a summary within a time-constrained context.

V. OPTIMIZATION-BASED AUTOMATIC TEXT SUMMARIZATION APPROACH

This section discusses the landscape of optimization-based ATS approaches found in the current literature. Then, subsequently shed light on the prevalent performance-related concerns, showcasing the existing challenges confronted by these optimization-based ATS methods. Ultimately, the paramount importance of addressing performance issues in optimization-based ATS approaches is underscored, an aspect frequently overlooked in the existing literature.

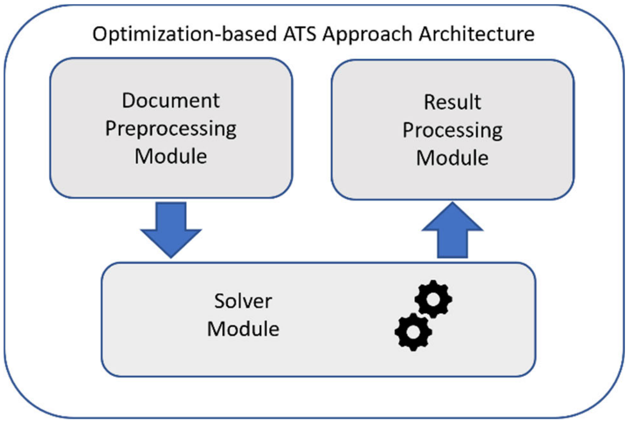

FIGURE 2. The overall program flow of the optimization-based ATS system encompasses document preprocessing, results processing, and solver modules.

Its reliance on optimization algorithms to generate summaries characterizes optimization-based ATS from other scientific-based ATS approaches. Fig. 2. above illustrates how optimization algorithms are typically employed as solver modules within an ATS system. Within the framework of optimization-based ATS, the task of automatic text summarization is viewed as an optimization problem that requires the application of optimization algorithms. Furthermore, ATS often manifests as an NP-complete problem from a computational complexity theory perspective, as Asgari et al. [6] noted when the input size is large enough. Therefore, the performance of an optimization-based ATS approach must be considered before said optimization-based ATS approach could be applicable in many domains, especially the real-time applications of ATS, as discussed previously in Table 1. In the optimization-based ATS approach, the textual input is commonly divided into individual sentences before an ATS problem can be formulated as an optimization problem. Each sentence is then encoded in a binary string, with '0' indicating the absence of the sentence in the final summary and '1' signifying that the sentence should be included in the final summary. Then, the operation of optimization algorithms may vary depending on their specific classes, but they

TABLE 3. Summary of different algorithm domains of ATS approaches with its gains and losses.

Domain	Gains	Losses	Literature (Year)
Optimization	• It produced great summary results regarding ROUGE metrics (summary accuracy against human-generated summary). • It does not require large datasets and handiwork data labeling like Machine Learning and Deep Learning approaches. • Non–platform specific ready, not requiring extensive setup, and thus applicable to more execution domains (i.e., flexible).	• Computational expensive, as demonstrated by the results of Sanchez–Gomez et al. [30], took about ~34h93min to finish execution in serial. • Not suitable for Real-Time Applications (RTA) due to its long execution time. • Performance in producing summary is highly reliance on the optimization algorithm used.	Refer Table 4
Fuzzy logic	• Fuzzy logic is relatively simple and justifiable as compared to other algorithm domains of ATS approaches. • It mimics human nature and is not always binary (true or false).	• A Fuzzy logic framework requires broad testing with equipment, which is not always feasible given the project's constraints. • Requires redundancy removal post-processing the summary input as some sentences might be redundant in the summary produced. (Patel et al.) [21].	Mutlu et al. (2019) [19]; Patel et al. (2019) [21]; Qassem et al. (2019) [22]; Srivastava et al. (2022) [34]; Agarwal et al. (2022) [50]
Deep learning	• The network can be trained based on the human reader's style. The feature set can be modified according to the user's requirements (Moratanch and Chitrakala) [18]. • Widely accessible with the current emergence of Large Language Models (LLMs). • Generate a highly accurate summary in both abstractive and extractive ATS. • It produced great summary results regarding ROUGE metrics (summary accuracy against human-generated summary). • Suitable for abstractive ATS approach.	• It requires human efforts to build the training data manually. • Neural networks are slow in the training and testing phases. Defining how the network yields a decision is difficult. (Moratanch and Chitrakala) [18]. • The prompt design is an important aspect that influences the outcome of the summary. (Wang et al.) [60] and (Ma et al.) [62].	Cheng & Lapata (2016) [8]; Kobayashi et al. (2015) [12]; Nallapati et al. (2017) [20]; Warule et al. (2019) [39]; Yao et al. (2018) [41]; Yousefi-Azar et al. (2017) [42]; Wang et al. (2018) [38]; Wazery et al. (2022) [49]; Tang et al. (2023) [51]; Zhang et al. (2023) [52]; Zhang et al. (2023) [53]; Guan and Tianxing (2020) [54]; Bajaj et al. (2021) [55]; Xu et al. (2022) [56]; Joshi et al. (2023) [57]; Ghadimi & Beigy (2023) [58]; Gao et al. (2023) [59]; Wang et al. (2023) [60]; Kieuvongngam et al. (2020)[61]; Ma at al. (2023) [62]; Shi et al. (2023) [71]; Goyal et al. (2022) [72]; Bharathi et al. (2023) [78]
Machine learning	• Relatively simple regression models can achieve better results than other classifiers (Gambhir & Gupta) [44].	• An extensive training data set is required to improve sentence selection in the final summary (Moratanch and Chitrakala) [18]. • Manually created extractive summaries in the original training documents must be labeled as either a "summary" or a "non-summary" in a large quantity (Moratanch and Chitrakala) [18].	John & Wilscy (2013) [11]; Alguliyev et al. (2019) [4]; Shetty & Kallimani, (2017) [32]

typically adhere to common steps such as the population initialization and the optimization parameters initialization. Optimization algorithm parameters are initially set, including factors like objective functions, mutation rate, crossover rate, maximum iteration, and error tolerance. Subsequently, optimization begins with the creation of initial populations, followed by ongoing population mutation, identifying the best solution, and considering the defined constraints and objective functions. As depicted in Fig. 2., the solution generated through optimization undergoes additional processing in the result processing module. The optimization-based ATS approach's generated summary is crafted at this stage based on the solution or multiple candidate solutions from the optimization algorithm and the provided textual input.

Furthermore, optimization-based ATS approaches are characterized by their substantial computational demands. This aspect is not unique to optimization-based approaches but is shared with machine learning and deep learning methods. When employing neural network frameworks in machine learning and deep learning ATS, extensive computational power is required, particularly for training neural networks on large datasets. Hence, optimization-based ATS, machine learning, and deep learning ATS inherently entail significant computational demands.

In the context of optimization-based ATS, the work conducted by Sanchez-Gomez et al. [29] employs the Artificial Bee Colony (ABC) algorithm to generate summaries. In this study, they recognize that the nature of ATS aligns well with multi-objective optimization techniques. This alignment arises because optimization-based ATS commonly seeks to maximize content coverage while minimizing redundant information simultaneously. Consequently, the multi-objective optimization technique is better suited for optimizing these dual criteria than the single-objective optimization technique. Sanchez-Gomez et al. [29] emphasize that utilizing a single-objective optimization technique can introduce subjective biases that significantly affect ATS solution quality. However, a simple decomposition technique could be introduced to address this by applying non-biased weight dynamically to drive the optimization toward the Pareto front.

Verma and Om [37] also endeavor to address the extraction-based approach for multi-document ATS. Their work employs the Shark Smell Optimization (SSO) metaheuristic as a means to tackle ATS challenges. In contrast to the approach presented by Sanchez-Gomez et al. [29], this work introduces a supervised model for sentence extraction and an unsupervised model for generating single documents from multiple source documents. Furthermore, the study highlights three crucial optimization criteria: context coverage, text unit non-redundancy, and relevancy. These criteria align with those that Sanchez-Gomez et al. [29] proposed, underscoring their significance in crafting effective summaries. The heuristic introduced by Verma and Om [37] prioritizes maximizing coverage and relevancy while minimizing redundancy, a strategy denoted as maximum coverage and relevancy with minimal redundancy (MCRMR). Moreover, the works presented by Saleh et al. [28], Umam et al. [36], Benjumea and León [7], Qiang et al. [23], Alguliyev et al. [4], and Sanchez-Gomez et al. [29] all focus on optimizing ATS for context coverage and redundancy reduction as primary optimization criteria. Therefore, it is identified that these criteria are crucial in yielding good-quality summaries in optimization-based ATS.

Furthermore, the exploration of optimization-based ATS is extended by Alguliyev et al. [4]. Their research proposes a two-stage sentence selection model incorporating clustering and optimization techniques. Additionally, the study introduces an adaptive differential evolution (DE) algorithm as a mutation strategy. Alguliyev et al. [4] emphasize the significance of the mutation operator, which alters a solution by adding or removing sentences in the candidate summary. More mutations are often necessary to yield higher-quality summaries, making the mutation operator crucial for achieving good ROUGE metrics evaluations. Their results are compared to those of Saleh et al. [28], representing the only multi-objective approach available in the literature at the time of their work. To date, other multi-objective approaches have been proposed by Sanchez-Gomez et al. [29], Wahab et al. [63], and Abo-Bakr and Mohamed [64] for multi-document ATS. It is worth noting that Alguliyev et al. [4] acknowledge that the analysis of CPU time for their proposed heuristic remains an open research question, indicating potential issues related to program execution time. However, the execution time is not disclosed as it lies outside the primary focus of their work.

Next, optimization-based ATS approaches pose performance-related challenges, and these issues have not received significant attention within the current research landscape. For instance, the work of Sanchez-Gomez et al. [30] underscores the criticality of addressing execution time concerns associated with their proposed heuristic. To tackle this issue, they employ OpenMP to parallelize multi-objective Artificial Bee Colony (MOABC) specifically for solving multi-document ATS. It is worth noting that the serial execution time of their approach is quite substantial, approximately 221485.10 seconds, equivalent to about 61 hours and 52 minutes, for the topic of d061j. This observation aligns with the findings in the work by Alguliev et al. [1], which also highlights performance issues related to the Differential Evolution (DE) algorithm.

Additionally, Sanchez-Gomez et al. [31] introduce an updated MOABC approach, incorporating a decomposition-based technique. The proposed enhanced heuristic is a decomposition-based multi-objective Artificial Bee Colony (MOABC/D) and has been parallelized using OpenMP to leverage multicore processing capabilities. Furthermore, Sanchez-Gomez et al. [31] achieved improved execution times for the serial version of MOABC/D, reaching an efficiency of 84.45% when executed on sixty-four threads. Nevertheless, the improvement demonstrated here does not remedy the performance issues at the core.

Delving into the performance challenges highlighted in earlier studies by Sanchez-Gomez et al. [29], Sanchez-Gomez et al. [30], Sanchez-Gomez et al. [31], and Saleh et al. [28], Wahab et al. [63] has introduced an improved repair operator scheme explicitly designed to address the inefficiencies associated with expensive repair operator schemes. In this enhancement, Wahab et al. [63] introduce MODE/D, explicitly targeting a performance issue in optimization-based ATS arising from the utilization of overly complex swarm intelligence optimization and the use of a resource-intensive ATS repair operator. Although the repair operation is executed based on the chance of similarity threshold on the solutions, it had consumed a significant portion of the optimization time in previous works Sanchez-Gomez et al. [29], Sanchez-Gomez et al. [30], and Sanchez-Gomez et al. [31]. The enhanced ATS operator scheme proposed by Wahab et al. [63] employs a lightweight sentence scoring approach to reduce recalculations and minimize the creation of intermediate frequency tables and document mean vectors for candidate solutions. However, the decomposition-based method of the weighted sum is employed by MODE/D, and it has affected the stability of the said approach. The CV values obtained by MODE/D are significantly higher than MOABC and MOABC/D.

Moreover, apart from the works presented by Sanchez-Gomez et al. [30] and Sanchez-Gomez et al. [31], Zamuda and Lloret [43] explore the parallelization of text summarization as a parallel task. In their study, they opt for a Grid Computing environment to parallelize ATS tasks, enabling multiple optimization processes to run concurrently. Their work demonstrates that conducting more function evaluations proves advantageous in terms of parallelism. Additionally, both Zamuda and Lloret [43] and Alguliyev et al. [4] employ self-adaptive differential evolution (DE) in their proposed heuristics. However, DE in Zamuda and Lloret [43] work operates concurrently without dependencies based on modeled data-driven summarization. Notably, Zamuda and Lloret [43] place a significant focus on recall metrics rather than performance metrics, unlike the approach taken by Sanchez-Gomez et al. [30] and Sanchez-Gomez et al. [31]. They highlight that recall values can be obtained from various ROUGE metrics, including ROUGE-1, ROUGE-2, ROUGE-L, or ROUGE-SU4, which represent distinct types of n-grams, indicating contiguous sequences of n items within a textual unit. Furthermore, they emphasize that higher recall values contribute to summary accuracy in terms of content coverage, and their model surpasses the powerful COMPENDIUM system in the Natural Language Processing (NLP) field.

In addition, Mojrian and Mirroshandel [17] draw attention to the adverse impact of fitness function formulation on the performance of evolutionary algorithms. Their work underscores the importance of aligning the enhancement of an evolutionary algorithm's fitness function with its performance evaluation. The performance of such algorithms ultimately determines their suitability for real-time ATS applications. Furthermore, Rudra et al. [24] emphasize the challenges posed by processing real-time information during emergencies, particularly on microblogging sites like "Twitter." They propose a method claimed to be time and memory-efficient, which allows them to outperform the baseline in terms of quality, event coverage (content coverage), effectiveness, and utility in disaster scenarios. Regrettably, their work does not emphasize performance metrics such as CPU time analysis, although it does highlight the recall score.

Moreover, the recent trends in optimization-based ATS approaches favor incorporating swarm intelligence optimization techniques. This trend is evident in the works of Tomer and Kumar [68], Debnath et al. [65], Aote et al. [66], and Vaissnave and Deepalakshmi [69], where the Firefly algorithm, Cat swarm optimization, Binary PSO, and Glowworm swarm optimization (GSO) have been adopted, respectively. Despite these optimization-based ATS methods showcasing notable improvements in summary quality, these works have ignored a just as important dimension: the performance of the newly introduced optimization-based ATS approaches. It is worth reiterating that the performance of an optimization-based ATS approach will allow it to be applicable in many domains, especially the real-time application of ATS that yearns for accuracy at a brisker pace.

In addition, the emergence of LLMs has breathed a new life and substantially impacted optimization algorithms. They contribute by excelling in representation learning, enabling effective transfer learning for optimization tasks with limited labeled data, assisting in problem formulation and encoding through natural language understanding, automating hyperparameter tuning, and enhancing algorithmic performance through integration and guidance. LLMs are also employed for problem-specific language modeling, aiding in domain-specific optimization nuances. Additionally, researchers explore their use in meta-optimization tasks, where LLMs learn to adapt optimization strategies across diverse tasks. In the field of ATS, LLMs are currently being used to generate extractive and abstractive summaries in many domains and have yielded promising results. The instances of these related works are listed in Table 3, primarily classified under deep learning-based ATS. While offering significant advantages, applying LLMs to optimization algorithms remains an evolving field, with ongoing research seeking innovative ways to leverage these models effectively.

In conclusion, optimization-based ATS approaches must consider algorithmic performance alongside the production of high-quality summary results. These approaches inherently demand significant computational resources. While parallelization efforts can help, addressing the fundamental aspects of ATS algorithm performance is essential. For example, these performance difficulties can be tackled by employing less complex optimization algorithms or optimizing the core algorithm more efficiently. Simplifying the algorithmic steps, improving the mutation operator, enhancing objective functions, and refining the repair and selection operators are all viable strategies to create a less computationally intensive yet intelligent optimization-bassed ATS approach suitable for real-time application. Lastly, the preceding delineation can be summarized in Table 4, where these optimization-based ATS approaches are further juxtaposed with their optimization algorithm classification, multi-objectivity usage, support for multi-document domains, and parallelization efforts.

TABLE 4. List of optimization-based ATS approaches.

Author (Year)	OA	MO	MD	MtC
Alguliev et al. (2011) [2]	Adaptive DE	-	✓	-
Mendoza et al. (2014) [16]	CHC	-	-	-
Benjumea and León (2015) [7]	Genetic Clustering	-	-	-
Alguliyev et al. (2015) [3]	DE	-	-	-
Saleh et al. (2015) [28]	NGSA – II (GA)	✓	✓	-
Qiang et al. (2016) [23]	Closed pattern mining	-	-	-
Alguliyev et al. (2019) [4]	Adaptive DE	-	-	-
Verma and Om (2019) [37]	SSO	-	-	-
Sanchez-Gomez et al. (2018) [29]	MOABC	✓	✓	-
Sanchez-Gomez et al. (2019) [30]	MOABC	✓	✓	✓
Saini et al. (2019a) [26]	Self-organized DE	✓	-	-
Saini et al. (2019b) [27]	Adaptive composite DE	✓	✓	-
Sanchez-Gomez et al. (2020) [31]	Decomposition–based MOABC (MOABC/D)	✓	✓	✓
Srivastava et al. (2021) [33]	DSO	-	-	-
Morijan & Mirroshandel (2021)	Modified quantum-inspired genetic algorithm	-	✓	-
Sanchez-Gomez et al. (2022) [67]	Multi-objective shuffled frog-leaping algorithm (MOSFLA)	✓	✓	-
Tomer and Kumar (2022) [68]	Swarm intelligence (Firefly algorithm) (FbTS)	✓	✓	-
Abo-Bakr and Mohamed (2023) [64]	Large-scale sparse multi-objective algorithm (LSSA)	✓	✓	-
Wahab et al. (2023) [63]	MODE/D	✓	✓	-
Debnath et al. (2023) [65]	Swarm intelligence (Cat swarm Optimization)	-	-	-
Aote et al. (2023) [66]	Binary PSO	-	✓	-
Vaissnave and Deepalakshmi (2023) [69]	Glowworm swarm optimization (GSO)	-	-	-
Thirumoorthy and Britto (2023) [70]	Social mimic optimization algorithm	-	-	-

OA = Optimization Algorithm, MO = Multi-objective, MD = Multi-Document, MtC = Parallel Multicore.

VI. COMPARATIVE PERFORMANCE REVIEW OF OPTIMIZATION-BASED ATS

In this section, the review focuses on the summary quality performance of optimization-based ATS. The scope of the reviews is defined as follows to examine the results of the works listed in Table 4:

- ATS approach that uses the DUC2002 dataset as input for their optimization-based ATS approach.
- ATS approach that employs performance measurement (summary quality) using ROUGE – n (ROUGE – 2).
- ATS approach that addresses multiple documents.

The reviews of optimization-based ATS approaches within the defined scope can then be discussed systematically and tabulated in Table 5.

TABLE 5. List of optimization-based ATS approaches in scope of review.

Author (Year)	OA	MO	MD	MtC
Alguliev et al. (2011) [2]	Adaptive DE	-	✓	-
Saleh et al. (2015) [28]	NGSA – II (GA)	✓	✓	-
Sanchez-Gomez et al. (2018) [29]	MOABC	✓	✓	-
Sanchez-Gomez et al. (2019) [30]	MOABC	✓	✓	✓
Saini et al. (2019b) [27]	Adaptive Composite DE	✓	✓	-
Sanchez-Gomez et al. (2020) [31]	Decomposition–based MOABC (MOABC/D)	✓	✓	✓
Tomer and Kumar (2022) [68]	Swarm intelligence (Firefly algorithm) (FbTS)	✓	✓	-
Abo-Bakr and Mohamed (2023) [64]	Large-scale sparse multi-objective algorithm (LSSA)	✓	✓	-
Wahab et al. (2023) [63]	MODE/D	✓	✓	-

OA = Optimization Algorithm, MO = Multi-objective, MD = Multi-Document, MtC = Parallel Multicore.

A. ROUGE-2 SCORES OF OPTIMIZATION-BASED ATS APPROACH

The remainder of this section is structured as follows: first, the ROUGE – 2 score of optimization-based ATS is discussed, followed by the average ROUGE – 2 score of the optimization-based ATS, and then the CV of these approaches is discussed further. Finally, the comparative re--view examines factors such as the number of fitness function effects on the best ROUGE – 2 score achieved by optimization-based ATS.

Based on Table 6 and Fig. 3., the optimization-based ATS approaches of Adaptive DE, NGSA – II, MOABC, CDE (Model – III), MOABC/D, LSSA (criterion parameter of $\mu = 7$), and MODE/D tabuluted. These approaches are solving for the same DUC2002 dataset, spanning topics from d061j to d070f. Notably, MODE/D outperforms the other optimization-based ATS approaches with an average ROUGE-2 score of 0.348, MOABC/D with 0.342, and LSSA with 0.335. In contrast, CDE, MOABC, NGSA – II, and Adaptive DE achieved average ROUGE-2 scores of 0.275, 0.312, 0.263, and 0.238, respectively, indicating slightly inferior performance. The primary distinction among these

TABLE 6. Overall Rouge-2 score of optimization-based ATS approaches for each Duc2002 topic.

	Topic	d061j	d062j	d063j	d064j	d065j	d066j	d067f	d068f	d069f	d070f	Avg
Adaptive DE *Alguliev et al. (2011) [2]*	Avg	0.266	0.188	0.245	0.194	0.144	0.201	0.239	0.491	0.184	0.224	**0.238**
	CV	109.02	146.28	84.9	144.33	145.14	127.86	98.33	78.21	90.22	116.07	*114.03*
NGSA-II *Saleh et al. (2015) [28]*	Avg	0.306	0.2	0.275	0.233	0.182	0.181	0.26	0.496	0.232	0.262	**0.263**
	CV	85.95	211	101.45	152.79	114.29	135.36	114.62	56.65	103.02	82.03	*115.72*
MOABC *Sanchez-Gomez et al. (2018) [29]*	Avg	0.365	0.342	0.272	0.308	0.198	0.29	0.356	0.444	0.24	0.305	**0.312**
	CV	25.43	6.60	1.84	2.83	13.32	6.54	1.39	18.83	3.15	0.76	*8.07*
CDE Model-III *Saini et al. (2019b) [27]*	Avg	0.301	0.313	0.217	0.266	0.214	0.265	0.33	0.327	0.23	0.294	**0.275**
	CV	91.36	95.85	93.09	93.23	96.73	92.45	93.64	97.25	86.52	88.44	*92.86*
MOABC/D *Sanchez-Gomez et al. (2020) [31]*	Avg	0.472	0.466	0.329	0.352	0.231	0.329	0.353	0.396	0.238	0.254	**0.342**
	CV	4.70	5.74	15.93	13.97	13.30	7.05	8.34	9.30	15.54	7.26	*10.11*
LSSA ($\mu = 7$) *Abo-Bakr and Mohamed (2023) [64]*	Avg	0.418	0.462	0.345	0.437	0.149	0.246	0.271	0.514	0.214	0.297	**0.335**
	CV	-	-	-	-	-	-	-	-	-	-	-
MODE/D *Wahab et al. (2023) [63]*	Avg	0.537	0.358	0.427	0.311	0.267	0.369	0.316	0.337	0.297	0.261	**0.348**
	CV	28.68	0.84	12.88	63.99	93.63	111.11	92.41	43.62	24.24	46.36	*51.78*

heuristics, particularly MOABC/D and MODE/D, lies in implementing the multi-objective optimization algorithm with a decomposition approach.

It is essential to highlight that Adaptive DE is the only approach that does not employ a multi-objective optimization algorithm. During the publication of this work, multi-objective optimization algorithms were still an emerging field of optimization algorithms, likely contributing to Adaptive DE's relatively weaker performance than its counterparts. Moreover, as Section IV-A discussed, CV indicates an ATS approach's stability or robustness when addressing different topics. A smaller CV value suggests that a particular ATS approach maintains greater stability when addressing the same topic compared to other approaches. In alignment with the observations from Table 6 and Fig. 3., it becomes evident that both MOABC and MOABC/D exhibit notably low CV values, indicating their stability in addressing the DUC2002 dataset. On the other hand, CDE, NGSA – II, Adaptive-DE, and MODE/D display comparatively higher CV values, suggesting slightly less stability in their performance with the DUC2002 dataset.

Nevertheless, it is essential to note that the CV for MOABC/D is slightly higher compared to MOABC. This increment could be due to the modifications, reworkings, and refactorings undertaken during the development of MOABC/D, potentially introducing a minor degree of instability. However, MOABC/D still achieves an impressive average ROUGE-2 score of 0.342, surpassing other heuristics except MODE/D. Furthermore, while MODE/D stands out as the top-performing optimization-based ATS approach based on average ROUGE-2 scores, it encounters challenges when dealing with significant CV values in comparison to MOABC and MOABC/D. The increase is due to the implementation of a less complex enhanced ATS repair operator scheme that operates based on a simple weighted-sum approach as a decomposition method in MODE/D. While acknowledging its compromised nature aimed at substantially reducing serial execution time, it has demonstrated significantly greater CV values than MOABC/D and MOABC [63].

Furthermore, Fig. 3. illustrates that all approaches faced challenges when addressing the topic d069j, resulting in suboptimal performance compared to the topic d068f, where most heuristics performed well, except for CDE (Model-III) and MODE/D. It is worth highlighting that among the presented approaches, MOABC/D and MODE/D stand out as the optimization-based ATS approaches employing a decomposition-based technique. In this approach, the initially intricate Pareto front is decomposed into several

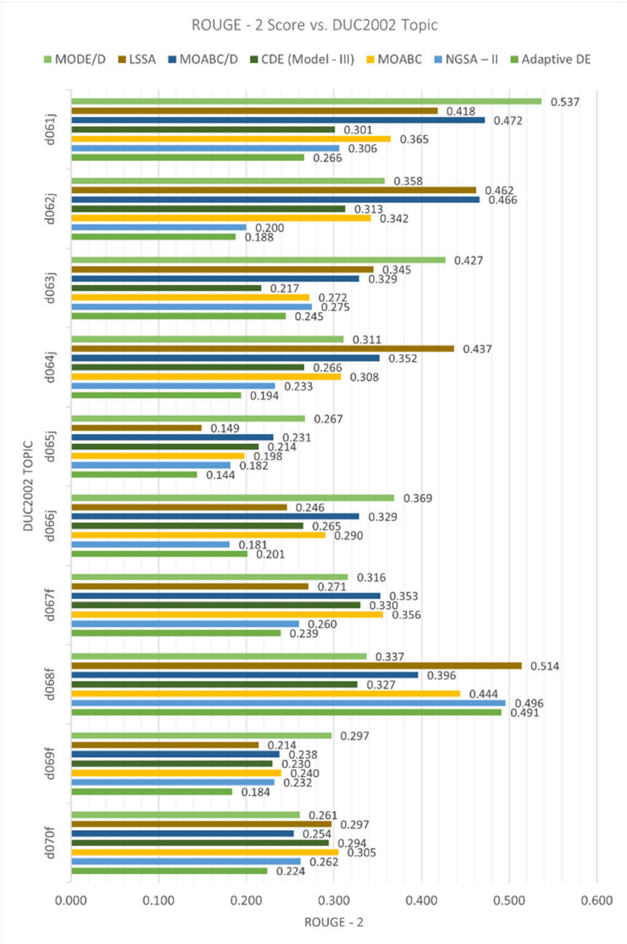

FIGURE 3. ROUGE-2 Score of optimization-based ATS approaches against DUC2002 topic.

subproblems, aiming to find a set of non-dominated solutions based on the information exchanged from neighboring subproblems. Consequently, the Pareto front is further exploited and explored by the optimization's gradient. Next, both MODE/D and MOABC/D results demonstrate the effectiveness of this decomposition approach compared to their counterparts. This comparison between MOABC and MOABC/D reveals a substantial increase in the average ROUGE – 2 scores, rising from 0.312 to 0.342, representing an approximately 9.61% improvement.

Next, Fig. 4. provides numerical indicators for the ATS approaches in brackets in the following format: Approach (Number of Fitness Function), e.g., MOABC/D (2). The number represents of the fitness functions integrated into each optimization-based ATS approach. Notably, it becomes evident that the approaches utilizing two fitness functions tend to outperform their counterparts, exemplified by MOABC and MOABC/D. Surprisingly, CDE (Model – III), despite employing the highest number of fitness functions, falls short in comparison to MOABC and MOABC/D. In addition, the top four approaches of MODE/D, MOABC/D, MOABC, and LSSA all employ the same objective functions: content

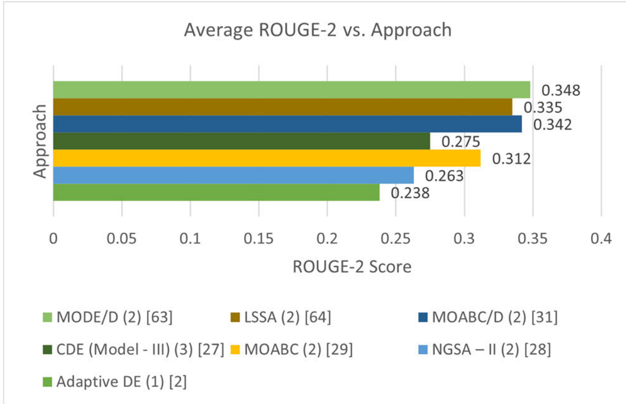

FIGURE 4. Average ROUGE-2 score of Optimization-based ATS Approaches.

coverage and similarity reduction (or redundancy reduction). The average ROUGE-2 scores obtained by the approaches suggest that further exploring these objective functions could still be an open research question.

Consequently, it can be inferred that the number of fitness functions incorporated does not consistently correlate with a higher average ROUGE score. In reality, an increased quantity of fitness functions often results in performance degradation for the optimization algorithm. The performance degradation occurs mainly because fitness functions represent one of the most computationally intensive aspects within the optimization algorithm. Having more than two or an excessive number of fitness functions will directly hamper the algorithm's performance as it significantly impacts the evaluation of solution scores.

B. PERFORMANCE REPORTS OF OPTIMIZATION-BASED ATS APPROACH

In this section, the studies listed in Table 7 consider algorithm performance as one of their performance metrics. They have reported the serial and parallel execution time for their optimization-based ATS approaches in generating a 200-word summary from the DUC2002 dataset. As previously mentioned, multi-document ATS is a computationally intensive task. Therefore, reporting its performance is crucial for its relevance and applicability in real-time ATS applications, as discussed in Section II. The previous section explored the influence of the number of fitness functions on the performance of optimization-based ATS. However, several other factors can affect the performance of optimization-based ATS, including:

- Number of fitness functions used (as well as the complexity of fitness function).
- The number of evaluation criteria (e.g., coverage, redundancy, relevancy) could further increase the fitness function's complexity.
- The complexity of the optimization algorithm could directly impact the optimization-based ATS approach.

- Inefficient data structures and recalculation of the same vector (i.e., non-cached static vector).
- Inefficient memory distribution of the given optimization problem (i.e., insufficient knowledge of machine underlying memory architectures).
- Repairing multiple candidate solutions.
- Extensive ATS repair operator scheme.
- Expensive sentence-scoring method for ATS repair operator.

With that said, numerous profiler tools are available for identifying hotspots and bottlenecks in optimization programs. Additionally, various methods to address slow serial programs, including the introduction of parallelization of the serial program, have been adopted by Sanchez-Gomez et al. [30] and Sanchez–Gomez et al. [31].

TABLE 7. List of optimization-based ATS approaches that report execution time.

Author (Year)	OA	OA Class	N	MO	MD	MtC
Sanchez-Gomez et al. (2018) [29]	MOABC	Swarm Intelligence	2	✓	✓	-
Sanchez-Gomez et al. (2019) [30]	MOABC	Swarm Intelligence	2	✓	✓	✓
Sanchez-Gomez et al. (2020) [31]	Decomposition–based MOABC (MOABC/D)	Swarm Intelligence	2	✓	✓	✓
Wahab et al. (2023) [63]	MODE/D	Differential Evolution	2	✓	✓	-

OA = Optimization Algorithm, N = Number of fitness functions used, MO = Multi-objective, MD = Multi-Document, MtC = Parallel Multicore.

Next, according to Table 8, the ATS approach presented by Sanchez–Gomez et al. [29] initially had a super lengthy average execution time of 33 hours. However, they recognized this issue and tried to address it in their subsequent work Sanchez-Gomez et al. [30]. In this later work, they parallelized the base algorithm of MOABC for solving extractive multi-document ATS, significantly reducing the average execution time from 30 hours to just 0.97 hours. Furthermore, Sanchez–Gomez et al. [31] further enhanced the base algorithm introduced by Sanchez – Gomez et al. [29], improving average ROUGE – 2 scores. They also have parallelized MOABC/D and compared it to their earlier work, Sanchez–Gomez et al. [29]. Consequently, the parallel work on MODE/D by Sanchez–Gomez et al. [31] achieved longer execution times, completing in 1.87 hours compared to the previously improved version, which took 0.97 hours. Nevertheless, the parallelization work done by Sanchez-Gomez et al. [30] and Sanchez-Gomez et al. [31] still struggles in terms of execution time, as demonstrated in Table 8.

TABLE 8. Execution time (ET) in seconds of serial MOABC, parallel MOABC, parallel MOABC/D, serial MODE/D from Sanchez-Gomez et al. (2018) [29], Sanchez-Gomez et al. (2019) [30], Sanchez-Gomez et al. (2020) [31], and Wahab et al. (2023) [63].

Topic	MO ABC *Serial*	Parallel MOABC *64 Threads*	MOABC /D *Serial*	Parallel MOABC /D *64 Threads*	MODE/ D *Serial*
d061j	221 485.10	6 223.95	197 058.27	3 536.18	21.15
d062j	83 056.88	2 323.98	73 496.95	1 408.37	8.81
d063j	-	-	475 206.18	8 982.26	173.30
d064j	-	-	270 498.76	4 832.18	18.91
d065j	-	-	775 763.26	14 220.60	26.08
d066j	-	-	295 616.36	5 589.73	21.09
d067f	86 756.31	2 656.30	78 333.87	1 479.17	9.61
d068f	88 537.94	2 616.69	78 924.33	1 476.08	6.50
d069f	-	-	1 286 430.37	23 688.97	37.37
d070f	125 739.33	3 658.22	114 875.14	2 068.70	10.43
Average	**121 115.11 ~33.64h**	**3 495.83 ~0.97h**	**364 620.35 ~101.28 h**	**6 728.22 ~1.87h**	**33.33**

Table 9 shows the performance of the optimization-based ATS approaches in a tabular manner where the execution time of these approaches is stated and transposed to hours for MOABC and MOABC/D. In addition, the WPS values are calculated alongside its average execution time. Next, a precise observation emerges, indicating that the execution time of MODE/D far precedes that of MOABC and MOABC/D, even in their parallel variants. This massive serial execution time reduction obtained by Wahab et al. [63] is with the integral technique of introducing an enhanced ATS repair operator scheme and by foundation using a less complex optimization algorithm, MODE. In addition, it is notable that the MODE/D approach repairs only a single candidate solution that surfaced from the optimization process and employs a lighter sentence-scoring scheme in their ATS repair operator. On average, the time taken by Parallel MOABC and Parallel MOABC/D to generate a 200-word summary from the DUC2002 dataset stands at approximately 0.97 hours and 1.87 hours, respectively. In retrospect, the long execution time of these optimization-based ATS approaches directly contributes to their inapplicability in real-time ATS applications. In terms of Execution Time (ET), the serial iteration of MODE/D exhibits superior performance compared to both Parallel MOABC and Parallel MOABC/D. Based on Table 9, it is observed that MODE/D yields exactly six words per second, whereas Parallel MOABC/D, MOABC/D Serial, MOABC Parallel, and

MOABC serial yield 1.65×10^{-3} wps, 57.21×10^{-3}wps, 548.52×10^{-6} wps, and 29.73×10^{-3} wps respectively. Therefore, in terms of WPS, only MODE/D yields a substantial WPS as compared to MOABC and MOABC/D. The usage of expensive ATS operator schemes with costly sentence scoring methods has directly attributed to the substantial execution times of these optimization-based ATS approaches.

TABLE 9. Word per second (WPS) of MOABC (Serial), parallel MOABC, MOABC/D (Serial), MOABC/D (Parallel), and MODE/D (Serial) in generating a 200-word long summary.

	MO ABC *Serial*	Parallel MOABC *64 Threads*	MOABC /D *Serial*	Parallel MOABC /D *64 Threads*	MODE/ D *Serial*
Average ET	121 115.11 ~33.64h	3 495.83 ~0.97h	364 620.35 ~101.28 h	6 728.22 ~1.87h	33.33s
WPS	**1.65×10^{-3}**	**57.21×10^{-3}**	**548.52×10^{-6}**	**29.73×10^{-3}**	**6.00×10^{0}**

In summary, optimization-based approaches in ATS face a multitude of potential influencing factors. Beyond the considerations previously discussed regarding their impact on performance, it is crucial to recognize the diverse development paths of individual ATS approaches. These factors may not uniformly affect every optimization-based ATS method, and there could be additional, unexplored elements contributing to performance variations. Emphasizing the need for performance metrics becomes paramount for the viability of new optimization-based ATS approaches, ensuring their relevance and applicability in real-time ATS applications.

VII. CONCLUSION

As highlighted in Section II, there is a significant demand for real-time Automatic Text Summarization (ATS) applications. However, despite this critical need, many recently introduced ATS approaches have overlooked the dimension of performance measurements. Consequently, while these approaches may produce excellent results in terms of summary quality, their suitability for real-time applications remains uncertain. Furthermore, Section III presented a refined classification of ATS, considering factors such as the ability to handle single or multiple documents, generate novel summaries, support multi-lingual documents, and the specific algorithmic domain of the ATS approach. In addition, Section III-D condensed the ATS approach domains, including optimization, fuzzy logic, deep learning, and machine learning, in a tabular manner. In Table 3, these scientific-based ATS approaches are juxtaposed with their gains and losses, enriching the current overview of the landscape of ATS approaches that are currently being approached in many techniques.

Notably, the deep learning-based approach in ATS has garnered considerable attention, particularly with the rise of Large Language Models (LLMs). Significantly, the deep learning-based approach in Automatic Text Summarization (ATS) has garnered considerable attention, particularly with the rise of Large Language Models (LLMs). Notably, Yang et al. [94] have explored the application of ChatGPT as a text summarizer, challenging the conventional use of ROUGE metrics as the sole indicator for evaluating ChatGPT's summaries and advocating for improved evaluation metrics. Furthermore, research by Wang et al. [60] and Ma et al. [62] underscores the importance of prompt design in enhancing the resilience of summaries generated by LLM deep learning-based ATS approaches. In essence, while deep learning-based ATS methods exhibit significant promise, there remains a need for comprehensive studies, particularly on widely used datasets, to assess their robustness in generating summaries across both extractive and abstractive ATS domains.

Next, in Section IV, this review study focused on validating ATS approaches, highlighting both the quality of the produced summaries and the potential performance measurements that can be reported for an ATS approach. Additionally, a more meaningful measurement approach for ATS that was introduced by Wahab et al. [63] is reiterated and calculated for the current optimization-based ATS approaches, providing guidelines for assessing the performance of an ATS approach. Section V centered on optimization-based ATS approaches, emphasizing their limitations from a technical perspective. Within this section, various performance issues associated with optimization-based ATS were also highlighted. In Section V, the argument was made for future optimization-based ATS research endeavors to incorporate performance considerations alongside ongoing efforts to enhance summary quality outcomes.

In Section VI, an analysis was conducted on both the quality of summaries, as indicated by ROUGE-2 scores, and the performance of optimization-based ATS approaches. It was observed that none of these approaches have been effectively applied to real-time applications, but one approach using MODE/D by Wahab et al. [63] achieves 33s in serial execution. However, there is a lot more work that could be applied to further speed up the presented approach, such as using parallelization and enhancing the core algorithm. This limitation arises from suboptimal reported performance and the high computational demands associated with most of these approaches. A comparative review was conducted on optimization-based ATS approaches, specifically those addressing multi-document ATS through multi-objective optimization algorithms.

Additionally, the impact of the number of fitness functions on the complexity of optimization-based ATS was discussed, highlighting how this could directly affect approach performance and providing a comparative assessment of its influence on ROUGE metric scores. From the deliberation, it is concluded that the ideal fitness function number is not more than two. Moreover, it is highlighted that the fitness function of content coverage and redundancy reduction or similarity reduction yields the best average ROUGE-2 scores, as demonstrated by Sanchez-Gomez et al. [31] and Wahab et al. [63] and Abo-Bakr and Mohamed [64]. Next, based on the optimization algorithm technique, the conclusion drawn from this analysis is that the decomposition-based approach employed by Sanchez-Gomez et al. [31] and Wahab et al. [63] yielded the best results in terms of summary quality, as indicated by ROUGE metrics. Furthermore, it is highlighted that performance issues rooted in Sanchez-Gomez et al. [31] have been addressed by Wahab et al. [63]. They have highlighted that the usage of an expensive ATS repair operator scheme along with a complex optimization algorithm has undeniably led the previous approach to inhibit a performance issue. Ultimately, what makes it worse is that the MOABC/D and MOABC are repairing for more than one candidate solution, and in the end, only a single good summary is evaluated. Naturally, the multi-objective optimization algorithm produces a set of non-dominated solutions. However, only a single best summary is required. (Abo-Bakr and Mohamed) [64]. Therefore, the act of repairing multiple candidate solutions is counterproductive in the context of general summary evaluation [63]. While the parallel MOABC/D algorithm performed better than the parallel MOABC algorithm when run on a multicore system with 64 threads, it falls short of being suitable for real-time ATS applications due to its extended execution time.

In conclusion, a suggestion was put forth in Section V to adopt a more straightforward approach when dealing with multi-objective ATS by utilizing a less intricate optimization algorithm. This strategic shift aims to enhance ATS performance, particularly for real-time applications. Furthermore, any optimization-based ATS approach should meticulously report its performance metrics to make it suitable for real-time ATS applications, allowing users to establish reasonable expectations regarding waiting times and instilling confidence in the newly introduced approach. Additionally, considering using a decomposition-based optimization technique to identify effective Pareto optimal solutions is crucial for such approaches. Lastly, an ATS approach must holistically consider overall algorithm performance, not solely focusing on summary quality, as neglecting this performance dimension will render it unsuitable for real-time applications.

REFERENCES

[1] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1675–1689, Apr. 2013.

[2] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "PSum-SaDE: A modifiedp-median problem and self-adaptive differential evolution algorithm for text summarization," *Appl. Comput. Intell. Soft Comput.*, vol. 2011, pp. 1–13, Jan. 2011.

[3] R. M. Alguliyev, R. M. Aliguliyev, and N. R. Isazade, "An unsupervised approach to generating generic summaries of documents," *Appl. Soft Comput.*, vol. 34, pp. 236–250, Sep. 2015.

[4] R. M. Alguliyev, R. M. Aliguliyev, N. R. Isazade, A. Abdi, and N. Idris, "COSUM: Text summarization based on clustering and optimization," *Expert Syst.*, vol. 36, no. 1, p. e12340, Feb. 2019, doi: 10.1111/exsy.12340.

[5] Q. A. Al-Radaideh and D. Q. Bataineh, "A hybrid approach for Arabic text summarization using domain knowledge and genetic algorithms," *Cognit. Comput.*, vol. 10, no. 4, pp. 651–669, Aug. 2018, doi: 10.1007/s12559-018-9547-z.

[6] H. Asgari, B. Masoumi, and O. S. Sheijani, "Automatic text summarization based on multi-agent particle swarm optimization," in *Proc. Iranian Conf. Intell. Syst. (ICIS)*, Feb. 2014, pp. 1–5.

[7] S. S. Benjumea and E. León, "Genetic clustering algorithm for extractive text summarization," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 949–956.

[8] J. Cheng and M. Lapata, "Neural summarization by extracting sentences and words," presented at the 54th Annu. Meeting Assoc. Comput. Linguistics, Berlin, Germany, 2016.

[9] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113679.

[10] S. Gehrmann, Y. Deng, and A. Rush, "Bottom-up abstractive summarization," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Association for Computational Linguistics, 2018, pp. 4098–4109.

[11] A. John and M. Wilscy, "Random forest classifier based multi-document summarization system," presented at the IEEE Recent Adv. Intell. Comput. Syst. (RAICS), Dec. 2013.

[12] H. Kobayashi, M. Noguchi, and T. Yatsuka, "Summarization based on embedding distributions," presented at the Conf. Empirical Methods Natural Lang. Process., Lisbon, Portugal, 2015.

[13] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser, and N. Shazeer, "Generating Wikipedia by summarizing long sequences," presented at the ICLR 2018. Accessed: Apr. 26, 2020. [Online]. Available: https://arxiv.org/pdf/1801.10198.pdf

[14] J. Lovinger, I. Valova, and C. Clough, "Gist: General integrated summarization of text and reviews," *Soft Comput.*, vol. 23, no. 5, pp. 1589–1601, Mar. 2019.

[15] A. Mahajani, V. Pandya, I. Maria, and D. Sharma, "A comprehensive survey on extractive and abstractive techniques for text summarization," in *Ambient Communications and Computer Systems*, 2019, pp. 339–351.

[16] M. Mendoza, C. Cobos, E. León, M. Lozano, F. Rodríguez, and E. Herrera-Viedma, "A new memetic algorithm for multi-document summarization based on CHC algorithm and greedy search," in *Proc. Mex. Int. Conf. Artif. Intell.* Cham, Switzerland: Springer, Nov. 2014, pp. 125–138.

[17] M. Mojrian and S. A. Mirroshandel, "A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA," *Expert Syst. Appl.*, vol. 171, Jun. 2021, Art. no. 114555.

[18] N. Moratanch and S. Chitrakala, "A survey on extractive text summarization," in *Proc. Int. Conf. Comput., Commun. Signal Process. (ICCCSP)*, Jan. 2017, pp. 1–6.

[19] B. Mutlu, E. A. Sezer, and M. A. Akcayol, "Multi-document extractive text summarization: A comparative assessment on features," *Knowl.-Based Syst.*, vol. 183, Nov. 2019, Art. no. 104848, doi: 10.1016/j.knosys.2019.07.019.

[20] R. Nallapati, F. Zhai, and B. Zhou, "SummaRuNNer: A of recurrent neural network based sequence model for extractive summarization documents," presented at the 31st AAAI Conf. Artif. Intell. (AAAI), San Francisco, CA, USA, 2017.

[21] D. Patel, S. Shah, and H. Chhinkaniwala, "Fuzzy logic based multi document summarization with improved sentence scoring and redundancy removal technique," *Expert Syst. Appl.*, vol. 134, pp. 167–177, Nov. 2019, doi: 10.1016/j.eswa.2019.05.045.

[22] L. A. Qassem, D. Wang, H. Barada, A. Al-Rubaie, and N. Almoosa, "Automatic Arabic text summarization based on fuzzy logic," presented at the 3rd Int. Conf. Natural Lang. Speech Process., Trento, Italy, 2019.

[23] J.-P. Qiang, P. Chen, W. Ding, F. Xie, and X. Wu, "Multi-document summarization using closed patterns," *Knowl.-Based Syst.*, vol. 99, pp. 28–38, May 2016.

[24] K. Rudra, P. Goyal, N. Ganguly, M. Imran, and P. Mitra, "Summarizing situational tweets in crisis scenarios: An extractive-abstractive approach," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 5, pp. 981–993, Oct. 2019, doi: 10.1109/TCSS.2019.2937899.

[25] D. Sahoo, A. Bhoi, and R. C. Balabantaray, "Hybrid approach to abstractive summarization," *Proc. Comput. Sci.*, vol. 132, pp. 1228–1237, Jan. 2018, doi: 10.1016/j.procs.2018.05.038.

[26] N. Saini, S. Saha, A. Jangra, and P. Bhattacharyya, "Extractive single document summarization using multi-objective optimization: Exploring self-organized differential evolution, grey wolf optimizer and water cycle algorithm," *Knowl.-Based Syst.*, vol. 164, pp. 45–67, Jan. 2019.

[27] N. Saini, S. Saha, A. Kumar, and P. Bhattacharyya, "Multi-document summarization using adaptive composite differential evolution," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, Dec. 2019, pp. 670–678.

[28] H. H. Saleh, N. J. Kadhim, and A. A. Bara, "A genetic based optimization model for extractive multi-document text summarization," *Iraqi J. Sci.*, vol. 56, no. 2B, pp. 1489–1498, 2015.
[29] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Extractive multi-document text summarization using a multi-objective artificial bee colony optimization approach," *Knowl.-Based Syst.*, vol. 159, pp. 1–8, Nov. 2018.
[30] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "Parallelizing a multi-objective optimization approach for extractive multi-document text summarization," *J. Parallel Distrib. Comput.*, vol. 134, pp. 166–179, Dec. 2019.
[31] 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.*, vol. 91, Jun. 2020, Art. no. 106231.
[32] K. Shetty and J. S. Kallimani, "Automatic extractive text summarization using K-means clustering," in *Proc. Int. Conf. Electr., Electron., Commun., Comput., Optim. Techn. (ICEECCOT)*, Dec. 2017, pp. 1–9.
[33] A. K. Srivastava, D. Pandey, and A. Agarwal, "Extractive multi-document text summarization using dolphin swarm optimization approach," *Multimedia Tools Appl.*, vol. 80, pp. 1–18, Jan. 2021.
[34] A. K. Srivastava, D. Pandey, and A. Agarwal, "Redundancy and coverage aware enriched dragonfly-FL single document summarization," *Lang. Resour. Eval.*, vol. 56, no. 4, pp. 1195–1227, Dec. 2022.
[35] J. Steinberger, "Evaluation measures for text summarization," *Comput. Inform.*, vol. 28, no. 2, pp. 251–275, 2009.
[36] K. Umam, F. W. Putro, G. Q. O. Pratamasunu, A. Z. Arifin, and D. Purwitasari, "Coverage, diversity, and coherence optimization for multi-document summarization," *Jurnal Ilmu Komputer dan Informasi*, vol. 8, no. 1, pp. 1–10, 2015.
[37] P. Verma and H. Om, "MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization," *Expert Syst. Appl.*, vol. 120, pp. 43–56, Apr. 2019.
[38] L. Wang, J. Yao, Y. Tao, L. Zhong, W. Liu, and Q. Du, "A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization," 2018, *arXiv:1805.03616*.
[39] P. D. Warule, S. D. Sawarkar, and A. Gulati, "Text summarization using adaptive neuro-fuzzy inference system," in *Computing and Network Sustainability*. Singapore: Springer, 2019, pp. 315–324.
[40] A. P. Widyassari, S. Rustad, G. F. Shidik, E. Noersasongko, A. Syukur, and A. Affandy, "Review of automatic text summarization techniques & methods," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1029–1046, 2020.
[41] K. Yao, L. Zhang, T. Luo, and Y. Wu, "Deep reinforcement learning for extractive document summarization," *Neurocomputing*, vol. 284, pp. 52–62, Apr. 2018, doi: 10.1016/j.neucom.2018.01.020.
[42] M. Yousefi-Azar and L. Hamey, "Text summarization using unsupervised deep learning," *Expert Syst. With Appl.*, vol. 68, pp. 93–105, Feb. 2017, doi: 10.1016/j.eswa.2016.10.017.
[43] A. Zamuda and E. Lloret, "Optimizing data-driven models for summarization as parallel tasks," *J. Comput. Sci.*, vol. 42, Apr. 2020, Art. no. 101101.
[44] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: A survey," *Artif. Intell. Rev.*, vol. 47, no. 1, pp. 1–66, Jan. 2017.
[45] D. Jani, N. Patel, H. Yadav, S. Suthar, and S. Patel, "A concise review on automatic text summarization," in *Proc. ICCIDM*, 2021, pp. 523–536.
[46] D. Yadav, J. Desai, and A. Kumar Yadav, "Automatic text summarization methods: A comprehensive review," 2022, *arXiv:2204.01849*.
[47] F. B. Goularte, S. M. Nassar, R. Fileto, and H. Saggion, "A text summarization method based on fuzzy rules and applicable to automated assessment," *Expert Syst. Appl.*, vol. 115, pp. 264–275, Jan. 2019.
[48] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multi-document summarization via deep learning techniques: A survey," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, May 2023.
[49] Y. M. Wazery, M. E. Saleh, A. Alharbi, and A. A. Ali, "Abstractive Arabic text summarization based on deep learning," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–14, Jan. 2022.
[50] R. Agarwal and N. Chatterjee, "Query-focused multi-document text summarization using fuzzy inference," *J. Intell. Fuzzy Syst.*, vol. 42, no. 5, pp. 4641–4652, Mar. 2022.
[51] L. Tang, Z. Sun, B. Idnay, J. G. Nestor, A. Soroush, P. A. Elias, Z. Xu, Y. Ding, G. Durrett, J. F. Rousseau, C. Weng, and Y. Peng, "Evaluating large language models on medical evidence summarization," *npj Digit. Med.*, vol. 6, no. 1, p. 158, Aug. 2023.
[52] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, "Benchmarking large language models for news summarization," 2023, *arXiv:2301.13848*.
[53] H. Zhang, X. Liu, and J. Zhang, "SummIt: Iterative text summarization via ChatGPT," 2023, *arXiv:2305.14835*.
[54] W. Guan, I. Smetannikov, and M. Tianxing, "Survey on automatic text summarization and transformer models applicability," in *Proc. Int. Conf. Control, Robot. Intell. Syst.*, Oct. 2020, pp. 176–184.
[55] A. Bajaj, P. Dangati, K. Krishna, P. A. Kumar, R. Uppaal, B. Windsor, E. Brenner, D. Dotterrer, R. Das, and A. McCallum, "Long document summarization in a low resource setting using pretrained language models," 2021, *arXiv:2103.00751*.
[56] S. Xu, X. Zhang, Y. Wu, and F. Wei, "Sequence level contrastive learning for text summarization," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 10, pp. 11556–11565.
[57] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "DeepSumm: Exploiting topic models and sequence to sequence networks for extractive text summarization," *Expert Syst. Appl.*, vol. 211, Jan. 2023, Art. no. 118442.
[58] A. Ghadimi and H. Beigy, "SGCSumm: An extractive multi-document summarization method based on pre-trained language model, submodularity, and graph convolutional neural networks," *Expert Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119308.
[59] M. Gao, J. Ruan, R. Sun, X. Yin, S. Yang, and X. Wan, "Human-like summarization evaluation with ChatGPT," 2023, *arXiv:2304.02554*.
[60] J. Wang, Y. Liang, F. Meng, Z. Sun, H. Shi, Z. Li, J. Xu, J. Qu, and J. Zhou, "Is ChatGPT a good NLG evaluator? A preliminary study," 2023, *arXiv:2303.04048*.
[61] V. Kieuvongngam, B. Tan, and Y. Niu, "Automatic text summarization of COVID-19 medical research articles using BERT and GPT-2," 2020, *arXiv:2006.01997*.
[62] C. Ma, Z. Wu, J. Wang, S. Xu, Y. Wei, Z. Liu, X. Jiang, L. Guo, X. Cai, S. Zhang, T. Zhang, D. Zhu, D. Shen, T. Liu, and X. Li, "ImpressionGPT: An iterative optimizing framework for radiology report summarization with ChatGPT," 2023, *arXiv:2304.08448*.
[63] M. H. H. Wahab, N. A. W. A. Hamid, S. Subramaniam, R. Latip, and M. Othman, "Decomposition–based multi-objective differential evolution for extractive multi-document automatic text summarization," *Appl. Soft Comput.*, vol. 151, Jan. 2023, Art. no. 110994.
[64] H. Abo-Bakr and S. A. Mohamed, "Automatic multi-documents text summarization by a large-scale sparse multi-objective optimization algorithm," *Complex Intell. Syst.*, vol. 9, no. 4, pp. 4629–4644, Aug. 2023.
[65] D. Debnath, R. Das, and P. Pakray, "Single document text summarization addressed with a cat swarm optimization approach," *Int. J. Speech Technol.*, vol. 53, no. 10, pp. 12268–12287, May 2023.
[66] S. S. Aote, A. Pimpalshende, A. Potnurwar, and S. Lohi, "Binary particle swarm optimization with an improved genetic algorithm to solve multi-document text summarization problem of Hindi documents," *Eng. Appl. Artif. Intell.*, vol. 117, Jan. 2023, Art. no. 105575.
[67] J. M. Sanchez-Gomez, M. A. Vega-Rodríguez, and C. J. Pérez, "A multi-objective memetic algorithm for query-oriented text summarization: Medicine texts as a case study," *Expert Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116769.
[68] 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.
[69] V. Vaissnave and P. Deepalakshmi, "Modeling of automated glowworm swarm optimization based deep learning model for legal text summarization," *Multimedia Tools Appl.*, vol. 82, no. 11, pp. 17175–17194, May 2023.
[70] K. Thirumoorthy and J. J. J. Britto, "A hybrid approach for text summarization using social mimic optimization algorithm," *Iranian J. Sci. Technol., Trans. Electr. Eng.*, vol. 47, no. 2, pp. 677–693, Jun. 2023.
[71] Y. Shi, P. Ren, J. Wang, B. Han, T. ValizadehAslani, F. Agbavor, Y. Zhang, M. Hu, L. Zhao, and H. Liang, "Leveraging GPT-4 for food effect summarization to enhance product-specific guidance development via iterative prompting," *J. Biomed. Informat.*, vol. 148, Dec. 2023, Art. no. 104533.
[72] T. Goyal, J. J. Li, and G. Durrett, "News summarization and evaluation in the era of GPT-3," 2022, *arXiv:2209.12356*.
[73] H. Hassani, C. Beneki, S. Unger, M. T. Mazinani, and M. R. Yeganegi, "Text mining in big data analytics," *Big Data Cognit. Comput.*, vol. 4, no. 1, p. 1, Jan. 2020.

[74] A. Jabbar, P. Akhtar, and S. Dani, "Real-time big data processing for instantaneous marketing decisions: A problematization approach," *Ind. Marketing Manage.*, vol. 90, pp. 558–569, Oct. 2020.
[75] O. Ozyegen, D. Kabe, and M. Cevik, "Word-level text highlighting of medical texts for telehealth services," *Artif. Intell. Med.*, vol. 127, May 2022, Art. no. 102284.
[76] K. M. Reddy and R. Guha, "Automatic text summarization for conversational chatbot," in *Proc. IEEE 8th Int. Conf. Converg. Technol. (ICT)*, Apr. 2023, pp. 1–7.
[77] S. Abdaljalil and H. Bouamor, "An exploration of automatic text summarization of financial reports," in *Proc. 3rd Workshop Financial Technol. Natural Lang. Process.*, 2021, pp. 1–7.
[78] G. B. Mohan, R. P. Kumar, S. Parathasarathy, S. Aravind, K. B. Hanish, and G. Pavithria, "Text summarization for big data analytics: A comprehensive review of GPT 2 and BERT approaches," in *Data Analytics for Internet of Things Infrastructure*, 2023, pp. 247–264.
[79] Z. A. Merrouni, B. Frikh, and B. Ouhbi, "EXABSUM: A new text summarization approach for generating extractive and abstractive summaries," *J. Big Data*, vol. 10, no. 1, p. 163, Oct. 2023.
[80] J. Pei, R. Hantach, S. B. Abbès, and P. Calvez, "Towards hybrid model for automatic text summarization," in *Proc. 19th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2020, pp. 987–993.
[81] M. S. Ansary, "A hybrid approach for automatic extractive summarization," in *Proc. Int. Conf. Inf. Commun. Technol. Sustain. Develop. (ICICTSD)*, Feb. 2021, pp. 11–15.
[82] D. Gunawan, A. Pasaribu, R. F. Rahmat, and R. Budiarto, "Automatic text summarization for Indonesian language using TextTeaser," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 190, no. 1, Apr. 2017, Art. no. 012048.
[83] B. Timalsina, N. Paudel, and T. B. Shahi, "Attention based recurrent neural network for nepali text summarization," *J. Inst. Sci. Technol.*, vol. 27, no. 1, pp. 141–148, Jun. 2022.
[84] N. Lin, J. Li, and S. Jiang, "A simple but effective method for Indonesian automatic text summarisation," *Connection Sci.*, vol. 34, no. 1, pp. 29–43, Dec. 2022.
[85] P. Bhattacharya, K. Hiware, S. Rajgaria, N. Pochhi, K. Ghosh, and S. Ghosh, "A comparative study of summarization algorithms applied to legal case judgments," in *Proc. 41st Eur. Conf. IR Res. (ECIR)*. Cologne, Germany: Springer, Apr. 2019, pp. 413–428.
[86] S. Polsley, P. Jhunjhunwala, and R. Huang, "Casesummarizer: A system for automated summarization of legal texts," in *Proc. 26th Int. Conf. Comput. Linguistics, Syst. Demonstrations (COLING)*, Dec. 2016, pp. 258–262.
[87] M. Koniaris, D. Galanis, E. Giannini, and P. Tsanakas, "Evaluation of automatic legal text summarization techniques for Greek case law," *Information*, vol. 14, no. 4, p. 250, Apr. 2023, doi: 10.3390/info14040250.
[88] Supreme Court of the United Kingdom. (2021). *UK-Abs: Abstractive Summaries of UK Supreme Court Case Documents (UKSC 2021/0001; Version V1). [Data Set]. Supreme Court of the United Kingdom.* Accessed: Dec. 3, 2023. [Online]. Available: https://www.supremecourt.uk/decided-cases/
[89] G. Michalopoulos, K. Williams, G. Singh, and T. Lin, "MedicalSum: A guided clinical abstractive summarization model for generating medical reports from patient-doctor conversations," in *Proc. Findings Assoc. Comput. Linguistics, EMNLP*, 2022, pp. 4741–4749.
[90] J. DeYoung, I. Beltagy, M. van Zuylen, B. Kuehl, and L. Lu Wang, "MS2: Multi-document summarization of medical studies," 2021, *arXiv:2104.06486*.
[91] A. Nentidis, G. Katsimpras, A. Krithara, S. L. López, E. Farré-Maduell, L. Gasco, M. Krallinger, and G. Paliouras, "Overview of BioASQ 2023: The eleventh BioASQ challenge on large-scale biomedical semantic indexing and question answering," in *Proc. Int. Conf. Cross-Lang. Eval. Forum Eur. Lang.* Cham, Switzerland: Springer, Sep. 2023, pp. 227–250.
[92] N. Zmandar, M. El-Haj, P. Rayson, M. Litvak, G. Giannakopoulos, and N. Pittaras, "The financial narrative summarisation shared task FNS 2021," in *Proc. 3rd Financial Narrative Process. Workshop*, 2021, pp. 120–125.
[93] U. Khanna, S. Ghodratnama, and A. Beheshti, "Transformer-based models for long document summarisation in financial domain," in *Proc. 4th Financial Narrative Process. Workshop@ LREC2022*, Jun. 2022, pp. 73–78.
[94] X. Yang, Y. Li, X. Zhang, H. Chen, and W. Cheng, "Exploring the limits of ChatGPT for query or aspect-based text summarization," 2023, *arXiv:2302.08081*.

MUHAMMAD HAFIZUL H. WAHAB received the B.Sc. degree (Hons.) in computer networks from Universiti Putra Malaysia (UPM), where he is currently pursuing the Ph.D. degree, with a focus on parallel and distributed computing and GPU programming. This demonstrates his commitment to advancing the field of computer science through cutting-edge research. In addition to his academic pursuits, he is actively involved in the industry as a Mobile Engineer at FinTech industry. His professional role at FinTech showcases his ability to apply his skills and knowledge in real-world settings, further enhancing his profile as a well-rounded professional in the field of computer science. He is an accomplished individual with a strong academic background and a deep passion for research. His dedication to networking excellence is further exemplified by his Huawei Certified Network Associate (HCNA—Routing and Switching) Certification, showcasing his expertise in the field. His academic achievements are complemented by his recognition as a recipient of the prestigious "Biasiswa Yang di-Pertuan Agong (BYPA)" Scholarship, awarded by the Yang di-Pertuan Agong, Malaysia, highlighting his exceptional academic and leadership qualities.

NOR HAFIZA ALI received the B.Sc. degree in computer networks from Universiti Putra Malaysia (UPM), in 2019, where she is currently pursuing the master's degree in parallel and distributed computing with the Institute for Mathematical Research (INSPEM). She has a combination of academic expertise in computer networking and parallel/distributed computing, complemented by practical experience as a Mobile Engineer in the industry. This dual focus on education and industry involvement suggests a well-rounded approach to her career in the field of information technology.

NOR ASILAH WATI ABDUL HAMID (Senior Member, IEEE) received the Ph.D. degree in computer science from The University of Adelaide, in 2008. She was a Visiting Scholar with the High-Performance Computing Laboratory, The George Washington University, USA, from 2013 to 2015. She is currently the Laboratory Head of the Laboratory of Computational Sciences and Mathematical Physics, Institute for Mathematical Research, Universiti Putra Malaysia, where she is also an Associate Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology. She has authored or coauthored more than 80 journals and conference proceedings in her area and her work was funded by the government and some by industry. Her research interests include parallel and distributed high-performance computing, cloud computing, and data-intensive computing. She received the CUDA Teaching Centre Award from NVIDIA, in 2015, and established the CUDA Laboratory, Faculty of Computer Science and Information Technology. She also serves as the reviewer for various renowned journals and conference proceedings.

SHAMALA K. SUBRAMANIAM (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in computer science from Universiti Putra Malaysia (UPM), in 1996, 1999, and 2002, respectively. She is currently a Professor with the Department of Communication Technology and Network, Faculty of Computer Science and Information Technology, UPM. Her research interests include computer networks, simulation and modeling, and scheduling and real-time systems.

ROHAYA LATIP (Member, IEEE) received the bachelor's degree in computer science from Universiti Teknologi Malaysia, in 1999, and the M.Sc. degree in distributed systems and the Ph.D. degree in distributed database from Universiti Putra Malaysia (UPM). From 2011 to 2012, she was the Head of the HPC Section, UPM. She was a Consultant for the Campus Grid Project and the Wireless for Hostel in Campus UPM Project. She is currently an Associate Professor with the Faculty of Computer Science and Information Technology, UPM. She is also the Head of the Department of Communication Technology and Networks and a Co-Researcher with the Institute for Mathematics Research (INSPEM). Her research interests include big data, cloud and grid computing, network management, and distributed databases.

MOHAMED OTHMAN (Senior Member, IEEE) received the Ph.D. degree (Hons.) from the National University of Malaysia. He was the Deputy Director of the Information Development and Communication Centre, where he was in charge of the UMPNet Network Campus, uSport Wireless Communication Project, and the UPM Data Centre. He is currently a Professor of computer science with the Department of Communication Technology and Networks, Universiti Putra Malaysia, where he is also an Associate Researcher and a Coordinator of high-speed machines with the Laboratory of Computational Science and Mathematical Physics, Institute for Mathematical Science Research. He has led six Malaysian, one Japanese, one South Korean, and three U.S. patents. He has published more than 300 international scientific journal articles. His research interests include computer networks, parallel and distributed computing, high-speed interconnection networks, network design and management (network security and wireless and traffic monitoring), consensus in the IoT, and mathematical models in scientific computing.