Performance analysis of large language models in text summarization: challenges and comparisons

1st Aritra Ghosh
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
Aritra.Ghosh2021@iem.edu.in

2nd Subhojit Ghosh
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
Subhojit.Ghosh2021@iem.edu.in

3rd Anupam Mondal
Department of CSE
Institute of Engineering & Management
UEM, Kolkata, India
anupam.mondal@iem.edu.in

***Abstract*- The purpose of this paper is to evaluate the efficiency of large language models (LLMs) in text summarization tasks in comparison to conventional methods. It provides background on summarization techniques, introduces LLM architectures like transformers, and discusses evaluation metrics like Recall-Oriented Understudy for Gisting Evaluation (ROUGE), BERTScore, and the emerging G-Eval framework that uses LLMs for self-evaluation. The challenges of conventional extractive (e.g. frequency-based, graph-based) and abstractive (e.g. template-based, sequence-to-sequence) summarization methods are outlined. A literature review covers recent research analyzing the capabilities, evolution, and assessment of LLMs. The paper then delves into the working mechanisms of LLMs, including processes like word embedding, self-attention, and autoregressive text generation. It compares the configurations of popular LLMs like GPT-4, GPT-3, and BERT, highlighting aspects such as model size, learning rates, optimizers, and maximum context length. Finally, the abstract summarizes the key points about understanding LLM architectures and evaluating their performance on summarization tasks, setting the stage for further exploration of this rapidly evolving field.**

Keywords*- Large Language Models (LLMs), Text Summarization, Transformer Architecture, Evaluation Metrics, ROUGE, BERTScore, G-Eval, GPT-4, GPT-3, BERT

I. Introduction

Summarization involves distilling the essential elements from a larger text while preserving its main ideas and concepts. This technique is vital across various fields. In journalism, articles are often condensed to emphasize key points of an event, enabling busy readers to stay updated. Researchers benefit from summaries by quickly grasping significant findings and methods of pertinent studies, facilitating efficient review of existing literature. In the corporate world, lengthy reports are transformed into concise summaries for executives who need quick insights before making important decisions. Social media platforms use summarization algorithms to provide users with brief updates that encapsulate the essence of conversations. Text condensation uses various approaches to distil large texts. Fuzzy logic handles vague information with fuzzy set theory, while concept-driven methods summarize key ideas based on relationships. Latent-semantic approaches like Latent Dirichlet allocation reveal semantic patterns, and machine learning algorithms such as Decision Trees, Support Vector Machines, and Naive Bayes use labelled datasets for summarization. Neural networks, including transformers and recurrent neural networks, generate summaries by learning text representations, and Conditional Random Fields (CRFs) treat summarization as a sequence labelling task to extract key phrases. Tree-based strategies analyze discourse structure, template-driven approaches use predefined templates, and rule-based methods rely on manually constructed rules. The ontology approach leverages knowledge bases to rank ideas, multimodal semantic techniques integrate various representations, information item methods prioritize significant units, and semantic graph-based approaches illustrate relationships between entities for summarization.

In this paper, we discuss the scope of previous research on this topic in Section 2, Literature Review. Popular and effective conventional approaches are covered in Section 3. Section 4 compares and highlights the advantages of large language models (LLMs) over traditional methods. In Section 5, we delve into the architecture of LLMs. The performance and evaluation metrics of LLMs are analyzed in Section 6. Section 7 compares LLMs with popular summarization models such as GPT, BERT, and Pegasus. Section 8 addresses the challenges faced by LLMs. In Section 9, we discuss the operational challenges and practical implications of using LLMs. Finally, Section 10 concludes the paper with a summary of our findings and potential directions for future research.

II. Literature Review

Summarization can be broadly categorized into two types: abstractive and extractive. In the extractive method, key sentences or phrases are selected from the original text, while in the abstractive method, new sentences are generated to convey essential

TABLE I
SUMMARY OF LITERATURE REVIEW ON LLM RESEARCH

LLM Papers	Scope	Key Findings	Methodology and Approach
Huang et al. (2022) [17]	Reasoning in LLMs	This aims to deliver a critical evaluation of Large Language Model (LLM) capabilities, explore methods for enhancing and assessing reasoning skills, and derive conclusions from prior research while suggesting future directions	By analysing and reviewing reasoning power of LLMs
Zhao et al. (2023) [13]	Evolution and impact of LLMs	Examine how LMs have evolved historically, taking into account the use of pre-trained language models such as GPT. Consider the special powers of LMs, their advantages and disadvantages as development resources, and the important contributions they have made to the fields of AI and NLP research.	Examining the development and effects of LLMs
Fan et al. (2023) [18]	Bibliometric review of LLM research	provides a thorough summary of research from 2017 to 2023, highlighting developments, trends, and discoveries about its dynamic nature, evolution, and significance of LLM research across a range of areas	study of more than 5,000 LLM publications using bibliometrics
Chang et al. (2023) [19]	Assessment of LLMs	Examine the approaches used to evaluate LLM programmes, paying particular attention to the issues of what, where, and how to perform evaluations. You should also identify any potential hazards and future challenge sets	An overview and critique of LLM assessment methods

information. The exponential growth of Large Language Models (LLMs) marks a significant advancement in AI, with extensive research examining their capabilities. Scholars have investigated LLMs' developments, applications, and transformative potential in text production, understanding, and reasoning abilities. Recent work by [21] provides a thorough analysis of LLMs' reasoning abilities, exploring strategies for improvement and evaluation standards. Another comprehensive review by [20] charts the development of LLMs, from the earliest models to the latest billion-parameter pre-trained language models (PLMs), highlighting contributions like ChatGPT. The study covers four main aspects: capacity appraisal, adaptability tuning, usage, and pre-training. [22] conducted a bibliometric review of over 5,000 papers on LLM research from 2017 to 2023, analyzing trends and advancements in algorithms and applications across various fields such as engineering, humanities, medicine, and social sciences. [23] examines LLM evaluation in academia and industry, emphasizing the importance of assessing societal and task-level impacts. The study uses three crucial factors to analyze LLM assessment methods: what, where, and how to evaluate [32]. It addresses reasoning, natural language processing, ethics, and education, while also looking at benchmarks and assessment techniques. Traditional methods relied on fine-tuning small datasets, but the advent of LLMs has shifted attention to their zero-shot generation capabilities, as highlighted in Table 1.

III. CONVENTIONAL APPROACHES

Understanding the architecture, training and evaluation of LLMs necessitates a foundational knowledge of prior conventional text summarization methods. These methods are broadly categorized into extractive and abstractive summarization techniques, each employing distinct strategies to condense text [37]. The most important parts of a text are found and chosen using a variety of criteria in extractive text summarizing techniques. Sentences with frequently occurring words are given priority by frequency-based approaches like Term Frequency-Inverse Document Frequency (TF-IDF) [**?**] occurring yet uncommon words, indicating their potential importance within the text. Graph-based approaches like TextRank and LexRank construct a graph with sentences as nodes, using sentence similarity as edges to determine the centrality of sentences, which reflects their significance in the summary. Clustering methods group sentences based on similarity measures, selecting representative phrases from each cluster to ensure a comprehensive summary that covers various topics from the original text [38]. Furthermore, to optimize the summarization process, machine learning algorithms like Naive Bayes, Support Vector Machines, and Decision Trees are [11] trained on features like sentence position, length, and phrase frequency. On the other hand, abstractive summarization methods involve creating new sentences that contain the most important [10] details from the original text. Template-based techniques apply predefined templates to structure summaries, utilizing key phrases identified through frequency analysis like TF-IDF. Semantic parsing transforms [9] text into semantic representations [15] such as semantic networks or logical forms, which are then used to generate summaries through natural language generation techniques erkan2004. Sequence-to-sequence (Seq2Seq) models, often incorporating recurrent neural networks, decode the input text into a condensed form to produce coherent summaries. These models have been instrumental [13] in advancing neural approaches to text summarization. Each of these methods has contributed to the field by offering unique approaches to reducing texts to their most informative components, paving the way for the development and enhancement of more sophisticated models like large language models (LLMs). [14]

Figure 1 illustrates the complete workflow for text summarization, starting from data acquisition and text pre-processing to employing language models (statistical, embedding, pre-trained) and summarization techniques (extractive and abstractive), followed by evaluation using methods like overlap, similarity, and LLM assessments. It highlights the tools, methods, and models involved at each stage of the process.

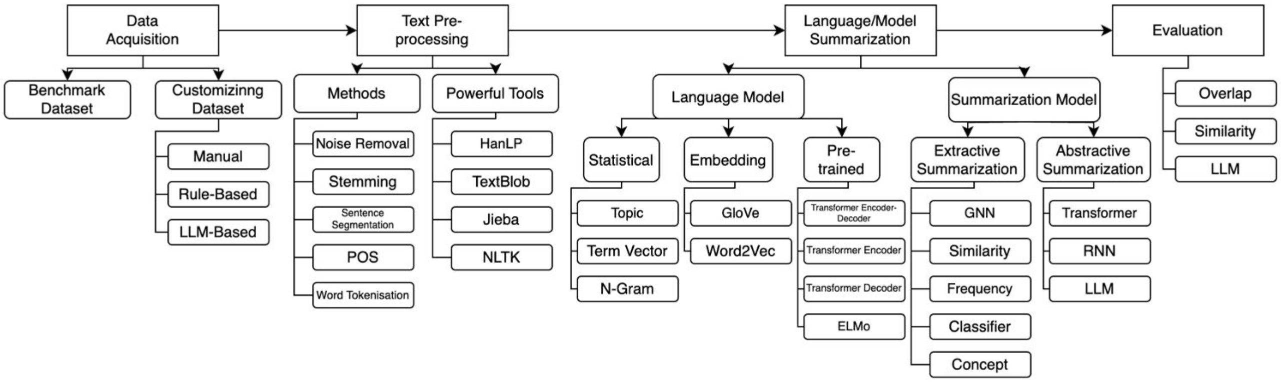


Fig. 1. Approaches and methodologies at various stages of summarization task

A. Challenges in Conventional Methods

Conventional text summarization methods, both extractive and abstractive, encounter several challenges that can impede their effectiveness. Extractive approaches often overemphasize frequently occurring terms, neglecting nuanced or critically relevant aspects of the text [10]. These methods typically lack a deep understanding of context, as frequency-based techniques may overlook important sentences containing less common terms [19]. Additionally, extractive methods such as graph-based approaches involve significant computational resources, particularly with larger documents, which can make these methods computationally intensive [8]. They treat sentences independently, leading to summaries that lack coherence and logical flow, and rely heavily on the quality of features selected, potentially resulting in suboptimal summaries if the feature selection is poor [17].

On the abstractive front, the challenges include limited flexibility due to reliance on predefined templates, which may not capture all pertinent information, leading to incomplete summaries [11]. The process of transforming text into semantic representations is both complex and resource-intensive, often requiring significant computational power and data, particularly when training Seq2Seq models that involve recurrent neural networks [14]. Early Seq2Seq models frequently struggled with producing grammatically correct and coherent summaries and maintaining factual accuracy was also a challenge [16].

Both methods also face issues of maintenance and adaptability; rules and models must be continually updated to accommodate new types of content and linguistic shifts, which can be both time-consuming and labour-intensive [18]. These challenges highlight the limitations of traditional text summarization techniques and underscore the need for more adaptive, context-aware methods in processing complex textual data.

IV. Advantages of LLM over Traditional ML approaches

Large language models, especially those utilizing transformer architectures, offer significant advantages over traditional machine learning approaches for text summarization. One of the primary benefits is their advanced contextual understanding, enabled by deep neural networks and self-attention mechanisms. These features allow LLMs to generate more coherent and contextually relevant summaries than traditional models, which often rely on simpler algorithmic approaches and lack the ability to grasp the subtleties of complex textual data [5]; [3]. Moreover, LLMs excel in few-shot learning, where they can adapt to new tasks with minimal training data. This is a stark contrast to traditional models that typically require extensive labelled datasets to achieve comparable performance. The few-shot learning capability of LLMs significantly reduces the need for large-scale data annotation, which can be time-consuming and resource-intensive. [2].

In terms of scalability, LLMs are inherently designed to manage large volumes of data and complex patterns efficiently. This makes them particularly well-suited for summarizing extensive documents or integrating content from multiple sources, tasks that may overwhelm traditional machine learning approaches due to their computational and algorithmic limitations [4]. Lastly, LLMs possess superior linguistic understanding due to their training on diverse and extensive language datasets. This allows them to better interpret nuances in language, including idioms, colloquialisms, and complex grammatical structures, which traditional models often misinterpret or oversimplify. As a result, LLMs can provide more accurate and nuanced summaries, enhancing the quality of information extraction in various applications [3].

V. Understanding LLM Architecture

The class of AI models referred to as LLMs was made expressly to generate human language and understanding. The fields of creating content, communicating, article composition, research, health and infotainment have greatly advanced the field of artificial intelligence and have been applied in a variety of contexts, such as education. [24]; [25]. An LLM is one kind of artificial intelligence AI which is made to mimic also produce human-like text, drawing information from their training corpus using computer programs based on specific algorithms. These models are developed with machine learning methods, particularly NLP (natural language processing).

Figure 2 shows the architecture of a transformer model, specifically illustrating the flow of input tokens through embedding layers, multiple attention blocks (highlighting the self-attention mechanism in one block), feed-forward layers, and ultimately to the classifier layer for output generation. It visualizes how attention mechanisms process and weigh input tokens to capture contextual relationships.

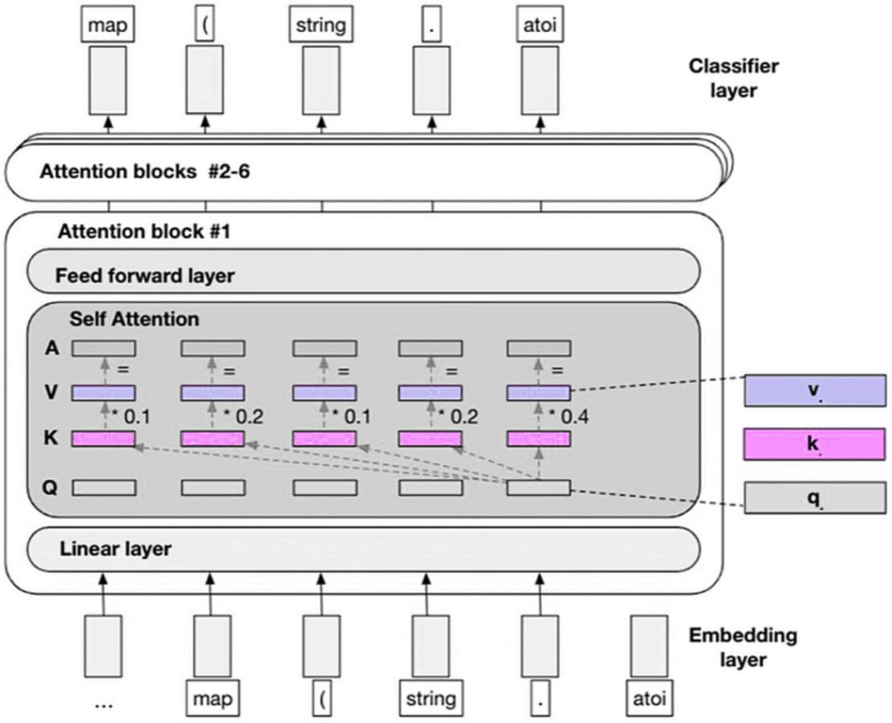


Fig. 2. Architecture of an LLM model

A. Working of a LLM

Large Language Models LLMs are the modern AI's cornerstone; they analyse also comprehend human language well by using a sophisticated transformer architecture. This section outlines the essential elements of the transformer architecture that are necessary for LLM operation.

1) Word Embedding: The initial step in the LLM's processing pipeline involves converting words into high-dimensional vectors. This transformation facilitates the representation of semantic similarity within the vector space, where semantically similar words are positioned closer together. This embedding is achieved through training on extensive text corpora, allowing prediction by the model, the word occurrence based on their contextual usage [6].

2) Positional Encoding: Following word embedding, positional encoding is applied to imbue the sequence of words with order information. Unlike embeddings that capture semantic meaning, positional encodings are crucial for maintaining sequence integrity, informing the model of the position of words within the sentence, which is vital for tasks requiring a sense of order such as translation and summarization. [5].

3) Transformers & Attention Mechanism: The LLM architecture at the heart, lies the transformer structure, characterized by self-attention mechanisms that assign weights to each word in a sequence (See Fig. 3), signifying the weightage of each word relative to others in the sequence [33]. This architecture component allows LLMs to process words in parallel and capture relationships between words at different positions within the text. Feed-forward neural networks subsequently process these weighted inputs to refine the learning of complex patterns [5].

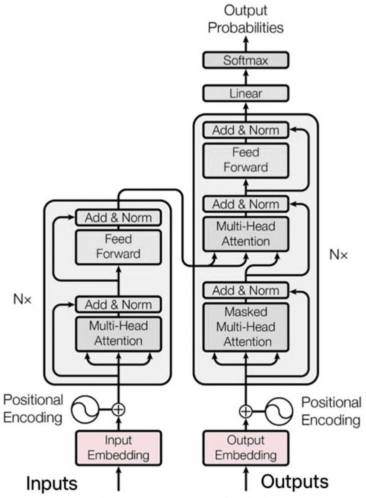


Fig. 3. Architecture of the Transformer Model

4) Text Generation: After going through these phases of processing the input data, LLMs generate text using an autoregressive technique in which the order of previously generated words is used to forecast each new word. This method generates coherent and contextually relevant outputs as text, underpinning the model's ability to produce fluent and accurate language constructs [7].

VI. EVALUATING A LLM TEXT SUMMARIZATION TASK

Traditionally, text summaries have been assessed using model-based scorers such as Recall-Oriented Understudy for Gisting Evaluation (ROGUE) and BERTScore. As previously said, these measures are helpful, but they frequently concentrate on superficial characteristics such as semantic similarity and word overlap. The types of scores used for evaluation and their correlation have been shown in Fig. 4.

1) Word Overlap Metrics: Evaluation measures such as Recall-Oriented Understudy for Gisting Evaluation (ROUGE) frequently measure the overlap in words or phrases between a reference summary and a generated summary to determine how well the summary is written. There is a greater chance of a higher overlap and higher scores if both summaries are of a comparable length. [29].

$$\text{ROUGE-N} = \frac{\sum_{\text{Reference}\in\text{References}} \sum_{\text{N-gram}\in\text{Reference}} \min(\text{Count}_{\text{hypothesis}}(\text{N-gram}), \text{Count}_{\text{reference}}(\text{N-gram}))}{\sum_{\text{Reference}\in\text{References}} \sum_{\text{N-gram}\in\text{Reference}} \text{Count}_{\text{reference}}(\text{N-gram})} \tag{1}$$

2) The BLEU Bilingual Evaluation Understudy: One measure of how well a generated sentence matches a reference sentence is the BLEU score. By adding a penalty factor to account for instances in which the generated sentence is shorter than the reference, it computes a score based on modified n-gram precision. The following represents the formula for BLEU:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2}$$

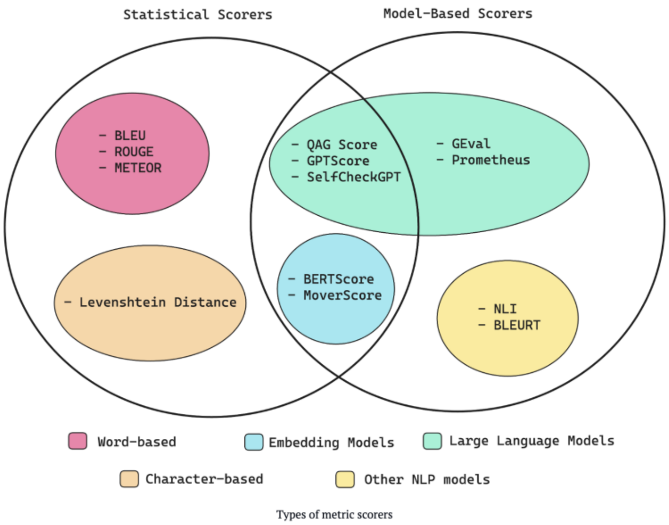


Fig. 4. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

3) Cross-Entropy Loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \tag{3}$$

These metrics struggle especially when the original text is composed of concatenated text chunks, which is often the case for a retrieval augmented generation (RAG) summarization use case. This is because they often need to effectively assess summaries for disjointed information within the combined text chunks.

A. Emerging Trends in LLM Evaluation: The G-Eval Framework

A recent trend among large language models involves using models like GPT-4, to evaluate themselves or other LLMs. One such framework is G-Eval, which leverages LLMs for evaluation purposes. G-Eval is a two-part process: the first part involves generating evaluation steps, and the second part uses these steps to produce a final score [30]. To illustrate how G-Eval works, let's consider a concrete example:

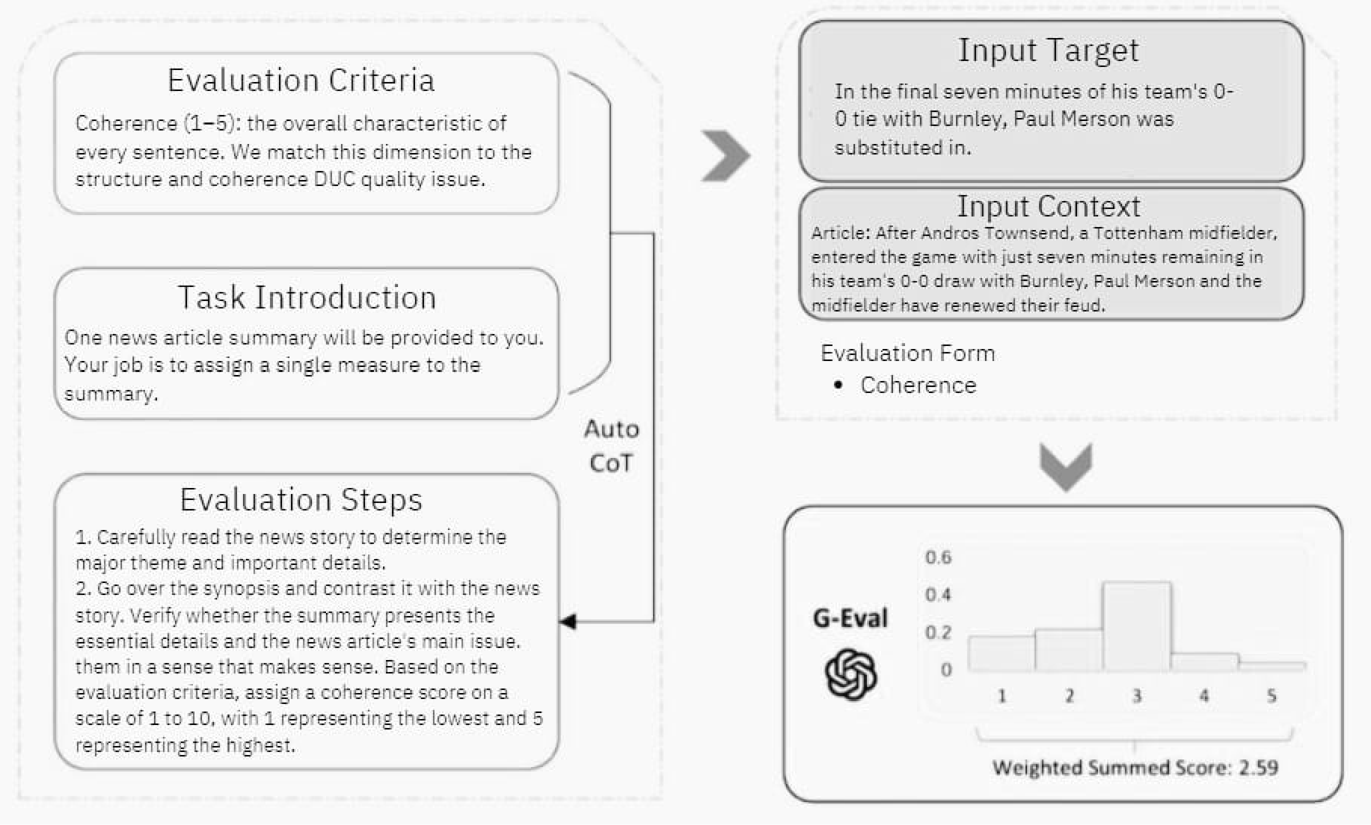
1) Generating Evaluation Steps: To introduce an evaluation task to GPT-4, you can start by clearly stating the task, such as "Rate this summary from 1 to 5 based on relevancy." Next, define the evaluation criteria to ensure clarity and consistency in the assessment. For example, you might specify, "Relevancy will be based on the collective quality of all sentences." This structured approach helps in setting clear expectations and provides a framework for GPT-4 to perform the evaluation accurately.

2) Using Evaluation Steps to Output a Final Score: To perform a comprehensive evaluation with GPT-4, start by concatenating the input text, the generated evaluation steps, the context, and the actual output. Then, ask GPT-4 to generate a score between 1 and 5, where 5 indicates higher relevancy. For a more nuanced evaluation, consider taking the probabilities of the output tokens from the LLM and normalizing the score by calculating their weighted summation. This step involves accessing the raw model outputs, not just the final generated text, providing a more fine-grained assessment that better reflects the quality of the outputs [31].

This process, particularly the optional third step, adds complexity because obtaining the probability of the output tokens typically requires access to the model's raw outputs. However, this approach offers a more detailed and accurate scoring mechanism.

Figure 5 helps visualize the described evaluation process.

The introduction of frameworks like G-Eval represents a significant advancement in the evaluation of LLMs, enabling more sophisticated and precise assessments of their outputs. By leveraging the capabilities of models like GPT-4, G-Eval not only enhances the evaluation process but also sets a precedent for using LLMs in self-assessment and the evaluation of other models. This innovation is relevant in the context of summarization, where the quality and relevancy of summaries are critical.



Evaluation Criteria

Coherence (1–5): the overall characteristic of every sentence. We match this dimension to the structure and coherence DUC quality issue.

Task Introduction

One news article summary will be provided to you. Your job is to assign a single measure to the summary.

Evaluation Steps

1. Carefully read the news story to determine the major theme and important details.
2. Go over the synopsis and contrast it with the news story. Verify whether the summary presents the essential details and the news article's main issue. them in a sense that makes sense. Based on the evaluation criteria, assign a coherence score on a scale of 1 to 10, with 1 representing the lowest and 5 representing the highest.

Input Target

In the final seven minutes of his team's 0-0 tie with Burnley, Paul Merson was substituted in.

Input Context

Article: After Andros Townsend, a Tottenham midfielder, entered the game with just seven minutes remaining in his team's 0-0 draw with Burnley, Paul Merson and the midfielder have renewed their feud.

Evaluation Form

- Coherence

Fig. 5. The overall process of G-EVAL involves first providing the large language model (LLM) with the Task Introduction and Evaluation Criteria, instructing it to generate a Chain of Thought (CoT) outlining comprehensive Evaluation Steps. Then, following a form-filling paradigm, the natural language generation (NLG) outputs are evaluated using the prompt and the resulting CoT. The final score is calculated by summing up all the output scores, weighted by their respective probabilities.

VII. Comparisons of Readily available LLM on summarization

Table II below shows a comparative summary of the configuration details and optimization settings used in different large language models (LLMs). The comparison encompasses models such as GPT-4, GPT-3, BERT, and others, highlighting their respective model sizes, learning rates, activation functions, and other crucial parameters. Understanding these configurations is vital for assessing the performance and capabilities of different LLMs in tasks such as text summarization and other natural language processing applications [39].

TABLE II
Configuration Details and Optimization Settings of Various LLMs

Model	**LR**	**Size**	**BS**	**CG**	**AF**	**Bias**	**NAH**	**SHS**	**NL**	**MCLDT**	**Optimizer**	**Dropout**
GPT-1	1×10^{-4}	125M	16-64K	AR	GeLU	Yes	12	768	12	512	Adam	0.1
GPT-2	1×10^{-4}	1.5B	16-64K	AR	GeLU	Yes	24	1280	48	1024	Adam	0.1
GPT-3	6×10^{-5}	175B	32K-3200K	CD	GeLU	Yes	96	12288	96	2048	Adam	-
GPT-4	3×10^{-5}	1.75T	320K-3000K	CD	GeLU	Yes	120-150	120	70	32768	Adam	-
PanGU-A	2×10^{-5}	207B	32K-3200K	CD	GeLU	Yes	96	24	24	2048	32764	-
BERT	1×10^{-5}	340M	16-64K	-	ReLU	Yes	16	1024	24	512	AdamW	0.1
BARD	-	340M	64K	-	ReLU	Yes	24	768	118	512	AdamW	-
LLaMA-2	1.5×10^{-4}	70B	400K	CD	SwiGLU	No	64	8192	80	AdamW	4096	-
Jurassic-1	6×10^{-5}	178B	32-3200K	CD	GeLU	Yes	96	13824	105	2048	AdaFactor	-
PalM	1×10^{-2}	540B	1000K-4000K	CD	SwiGLU	No	48	28432	96	2048	Adam	0.1
LaMDA	-	137B	256K	CD	GeLU	-	128	8192	Adam	60	-	-
T5	1×10^{-2}	11B	64K	ED	ReLU	No	128	1024	80	512	AdamW	0.1

The explanation highlights several aspects of large language models (LLMs). Model size and layers are crucial, as GPT-4, with its 1.8 trillion parameters, offers greater capacity for understanding and generating text compared to earlier models like GPT-1 and BERT, which have far fewer parameters but have laid the groundwork for modern LLMs. Learning rates and optimizers vary, with models like PanGU-α and BLOOM using a rate of 6×10^{-5}, and Adam and AdamW being common optimizers due to their effectiveness in training deep learning models. Batch sizes significantly impact training stability and speed, and dropout rates are specified for some models, indicating strategies to prevent overfitting. The number of attention heads (NAH) and the size of hidden states (SHS) provide insight into the models' architectures, influencing their ability to capture complex patterns in data [40]. Maximum Context Length During Training (MCLDT) highlights the models' capacity to handle long-range dependencies in text, with GPT-4 supporting the longest context length at 32,768 tokens. Overall, this detailed overview helps in understanding the strengths and potential applications of these leading LLMs in text summarization.

In summary, this table provides a comprehensive overview of the diverse architectures and configurations of leading LLMs, aiding in the understanding of their strengths and potential applications in text summarization.

A. Dataset used

Table III presents detailed characteristics of the datasets used in the study, providing a comprehensive overview of the training, testing, and validation splits for each dataset (CNN, DailyMail, NYT, and XSum). It includes the average document length in terms of words and sentences, the average summary length in terms of words and sentences, and the percentage of novel bi-grams in the gold summary. These metrics are crucial for understanding the diversity and complexity of the datasets, which in turn influence the performance and evaluation of the large language models (LLMs) in text summarization tasks. The CNN and DailyMail datasets, for instance, show a balance in average document and summary lengths, while the XSum dataset stands out with a notably high percentage of novel bi-grams, highlighting its unique challenge for summarization models [41].

TABLE III
CHARACTERISTICS OF DATASETS USED FOR MODEL TRAINING AND EVALUATION

Datasets	docs (train/test/val)	avg. doc length	avg. summary length	% of novel bi-grams
		words sentences	words sentences	in gold summary
CNN	90,266/1,220/1,093	760.50 / 33.98	45.70 / 3.59	52.90
DailyMail	196,961/12,148/10,397	653.33 / 29.33	54.65 / 3.86	52.16
NYT	96,834/4,000/3,452	800.04 / 35.55	45.54 / 2.44	54.70
XSum	204,045/11,332/11,334	431.07	-	83.31

VIII. OPERATIONAL CHALLENGES FOR LLMS

Large Language Models (LLMs) have revolutionized machine learning with their human-like text generation capabilities. Despite their rapid development, these models face significant challenges impacting their effectiveness and broader application. LLMs require vast datasets, raising concerns about the quality and biases that could propagate harmful or inaccurate information. The process of breaking text into tokens, known as tokenization, is crucial but sensitive, with minor variations potentially altering output meaning and leading to adversarial attacks. Training LLMs demands significant computational resources and energy, contributing to sustainability concerns [27]. Fine-tuning these models for specific tasks is resource-intensive and costly, requiring extensive human labor [26]. Extensive training often results in slower inference speed, limiting their use in applications that require quick responses. LLMs are also limited by the number of tokens they can consider, posing challenges in maintaining coherence over long documents or conversations [35]. Inherent biases in training data can lead to discriminatory or harmful content, necessitating ethical deployment. The reliance on historical data means LLMs may lack updated information, which is problematic in scenarios requiring up-to-date knowledge. Traditional metrics may not fully capture LLM performance and can be manipulated, making it essential to develop robust evaluation methodologies [36]. Finally, as language evolves, static benchmarks may not adequately reflect this dynamism, requiring evaluation frameworks to adapt to context and language use to remain relevant.

IX. DISCUSSION

Large Language Models (LLMs) have significantly improved text summarization. Models like GPT-3 Davinci, Pegasus, and BART have shown exceptional performance, as measured by metrics such as BERT and ROUGE scores. These models benefit from fine-tuning and architectural optimizations. While effective, LLMs face challenges in real-world applications, including handling diverse data and maintaining consistency. Issues of bias, ethics, and resource requirements also persist. Future research should focus on improving efficiency, scalability, and accessibility of these models. Addressing practical and ethical concerns, such as interpretability and energy consumption, is crucial. Continued refinement of LLMs has the potential to revolutionize information processing and consumption in the digital age, benefiting society at large.

X. Conslusion

The current research provides estimates of the effectiveness of large language models in text summarization against traditional approaches. It turns out that LLM—especially GPT-3 Davinci, Pegasus, and BART—are very superior over the conventional approaches. Table IV. These models all performed well on almost all metrics—GPT-3 Davinci tends to dominate most of these, followed by a really close second place from Pegasus and BART. This is powered by state-of-the-art neural architectures with large and diverse datasets.

While the potentials opened by LLMs go beyond summary generation into machine translation and text creation, there exist several challenges to be overcome. Some of these include overcoming bias using models, computational efficiency, and the ethical points raised against fully automated text generation. This paper judges that it is the performance of these LLMs that marks the growth pertinent to a new milestone in text summarization technology and sets a stage for future developments in this field.

TABLE IV
Performance Metrics of LLMs on Text Summarization

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**	**BERT Score**
GPT-3 Davinci	0.272	0.096	0.255	0.868
Pegasus	0.236	0.060	0.213	0.851
BART	0.224	0.052	0.197	0.838

References

[1] Greenwade, G.D.: The Comprehensive TeX Archive Network (CTAN). TUGBoat **14**(3), 342–351 (1993)

[2] Brown, T.B., et al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)

[3] Devlin, J., et al.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (2019)

[4] Kaplan, J., et al.: Scaling laws for neural language models. arXiv preprint arXiv:2001.08361 (2020)

[5] Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)

[6] Mikolov, T., et al.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

[7] Radford, A., et al.: Language Models are Unsupervised Multitask Learners. OpenAI Blog (2019)

[8] Erkan, G., Radev, D.R.: LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research **22**, 457–479 (2004)

[9] Joachims, T.: Text categorization with support vector machines: Learning with many relevant features. In: European Conference on Machine Learning (1998)

[10] Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004)

[11] Reiter, E., Dale, R.: Building natural language generation systems. Cambridge University Press (2000)

[12] Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, Inc. (1983)

[13] Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining (2000)

[14] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to Sequence Learning with Neural Networks. In: Advances in Neural Information Processing Systems **27** (2014)

[15] Wong, K.-F., Mooney, R.J.: Learning for semantic parsing with statistical machine translation. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (2007)

[16] Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473 (2014)

[17] Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research **3**(Jan), 993-1022 (2003)

[18] Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artificial Intelligence Review 47(1), 1-66 (2017)

[19] Luhn, H.P.: The automatic creation of literature abstracts. IBM Journal of research and development **2**(2), 159-165 (1958)

[20] Zhao, W.X., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)

[21] Huang, J., Chang, K.C.-C.: Towards reasoning in large language models: A survey. arXiv preprint arXiv:2212.10403 (2022)

[22] Fan, L., et al.: A bibliometric review of large language models research from 2017 to 2023. arXiv preprint arXiv:2304.02020 (2023)

[23] Chang, Y., et al.: A survey on evaluation of large language models. arXiv preprint arXiv:2307.03109 (2023)

[24] Kasneci, E., et al.: ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences **103**, 102274 (2023)

[25] Hadi, M.U., et al.: A survey on large language models: Applications, challenges, limitations, and practical usage. TechRxiv (2023)

[26] Kardum, M.: Rudolf Carnap – The Grandfather of Artificial Neural Networks: The Influence of Carnap's Philosophy on Walter Pitts. In: Guide to Deep Learning Basics: Logical, Historical and Philosophical Perspectives, pp. 55–66 (2020)

[27] Leech, G.: Corpora and theories of linguistic performance. In: Svartvik, J. (ed.) Directions in Corpus Linguistics, pp. 105–122 (1992)

[28] Liddy, E.D.: Natural language processing. Annual Review of Information Science and Technology (2001)

[29] Cronin, B.: Annual Review of Information Science and Technology (2004)

[30] Hain, D.S., et al.: A text-embedding-based approach to measuring patent-to-patent technological similarity. Technological Forecasting and Social Change 177, 121559 (2022)

[31] Curto, G., et al.: Are AI systems biased against the poor? A machine learning analysis using word2vec and glove embeddings. AI & Society **37**, 1–16 (2022)

[32] Azunre, P.: Transfer Learning for Natural Language Processing. Simon and Schuster (2021)

[33] Shi, Y., Larson, M., Jonker, C.M.: Recurrent neural network language model adaptation with curriculum learning. Computer Speech & Language **33**(1), 136–154 (2015)

[34] Mikolov, T., Zweig, G.: Context dependent recurrent neural network language model. In: 2012 IEEE Spoken Language Technology Workshop (SLT), pp. 234–239. IEEE (2012)

[35] Kovacevic, A., Keco, D.: Bidirectional LSTM networks for abstractive text summarization. In: Advanced Technologies, Systems, and Applications VI: Proceedings of the International Symposium on Innovative and Interdisciplinary Applications of Advanced Technologies (IAT) 2021, pp. 281–293. Springer (2022)

[36] Fahad, N.M., et al.: SkinNet-8: An efficient CNN architecture for classifying skin cancer on an imbalanced dataset. In: 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–6. IEEE (2023)

[37] Dey, Monalisa, et al. ”SETA-Extractive to Abstractive Summarization with a Similarity-Based Attentional Encoder-Decoder Model.” ECTI Transactions on Computer and Information Technology (ECTI-CIT) 18.3 (2024): 319-328.

[38] Mondal, Anupam, et al. ”An Automatic Summarization System to Understand the Impact of COVID-19 on Education.” Applications of Machine intelligence in Engineering. CRC Press, 2022. 379-386.

[39] Dey, Monalisa, Anupam Mondal, and Dipankar Das. ”JUNLP at IJCNLP-2017 Task 3: A Rank Prediction Model for Review Opinion Diversification.” Proceedings of the IJCNLP 2017, Shared Tasks. 2017.

[40] Dey, Monalisa, Anupam Mondal, and Dipankar Das. ”NTCIR-12 MOBILECLICK: Sense-based Ranking and Summarization of English Queries.” Ntcir. 2016.

[41] Ghosh, Bavrabi, Aritra Ghosh, Subhojit Ghosh, and Anupam Mondal. ”An Analytical Study of Text Summarization Techniques” Proceedings of the IEMTRONICS 2024