SANDIA REPORT
SAND2024-11549
Printed September 2024

A Summary of Advances in Document Summarization from 2023-2024

Hamilton Link
Adam Hooker

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@osti.gov
Online ordering:	http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.gov
Online order:	https://classic.ntis.gov/help/order-methods/

ABSTRACT

In computer science, Document Summarization is the task of condensing some quantity of text and related content through automated means. The resulting summary is expected to emphasize key information from the original text in the service of some purpose. In this document, we review recent literature in text summarization. The "hybrid" extractive-abstractive approaches training transformers and graph neural networks to prioritize and incorporate content into generated text continue to be explored, with some efforts continuing to test such methods on domain-specific and non-English text. Some of the latest efforts have also sought to enable users to adjust summaries with queries or other structure and benefit from new chat and instructible large language models. While summaries are evaluated qualitatively on relevance and salience, in addition to brevity and faithfulness to the original text, many traditional metrics are poorly correlated to human judgements and the field continues to seek good quantitative metrics for evaluating competing approaches; this related task has also begun to test reinforcement-learning style agentic LLM-based solutions. We see opportunities for leveraging causal analysis, psychology, and agent-based solutions to create more trustworthy, natural, and interactive content summarization systems.

ACKNOWLEDGEMENTS

We would like to thank Alex Ezazi and Eyas Taifour for their contributions under the Stanford AI Professional Program to a prior examination of automatic text summarization that informed this review.

CONTENTS

This page left blank

1. GENERAL PROBLEM

Operative Guiding Question

The operative question at hand, motivating this review, is what progress is being made in natural language processing that would further the goal of developing a capable research assistant. This literature survey notes progress specifically in summarization; other examinations might address progress in open-domain scientific Q&A, education and mentoring, experiment design, etc. in the pursuit of "human uplift".

Document Summarization

Document summarization is a multi-faceted problem, and this is reflected in a body of literature that addresses diverse combinations of these facets with varying success. Simplistically, the goal is to take some amount of text and produce a shorter piece of text, but this glosses over the complexity of the task: when summarizing some amount of content, human authors must consider the purpose to which they are writing, the audience, the level of detail that must be preserved, and many other nuances. Researchers have studied successful human summary strategies and have identified some basic rules [1]:

- Include important ideas.
- Delete trivia.
- Delete repeated ideas.
- Collapse lists.
- Choose or create a topic sentence.

From a humanistic viewpoint this translates into many subproblems:

- Collecting relevant source material (text).
- Collecting user constraints on the summary such as style, focus, length etc.
- Identifying the relationships between portions of the source (ranging from individual words to full documents).
- Scoring or extracting the important portions of source (from entities to sections).
- Explicitly representing the assertions being made in the source.
- Generating text for a summary given some subset of the source text, user guidance, portion relationships, indicators of importance, and extracted information.
- Organizing, styling, and checking the generated summary.
- Providing access to or references to portions of the source document that support the summary.
- Tailoring these steps based on specific requirements of the domain such as in law or medicine.

Some of this complexity is reflected in the diversity of solutions. Early summarization research (e.g. [2]) scored sentences, potentially adjusted those scores using relationships, and directly quoted the highest-scoring passages as the final summary. More recent work (e.g. [3]) has developed deep neural networks that consume source material and directly generate a final summary.

2. SUMMARY OF SURVEYED 2023-2024 PUBLICATIONS

Building upon a previous literature survey in this space, the authors scraped the internet for relevant publications appearing in 2023 and 2024. The abstracts were scanned to get an overall sense of the direction of the field before reviewing the details of new architectures in the selected papers presented here. For a summary of previously reviewed literature, see Appendix A; for a brief scan of pre-transformers strategies, see Appendix B.

The following subsections are a suggested framework for the field of text modeling document summarization. We have organized the literature based on the authors' extension of prior methods (verbatim extraction, abstract restatement, and hybrids of these), expansion of the process (as responding to a guiding question, or as a multi-step process), or use of multi-agent and reinforcement learning strategies (developing the task for autonomous reward-driven agents).

2.1. Content-Focused

Extractive, abstractive, and mixed summarization, where the task is defined as putting a document in and getting a summary out. – This is well heeled territory; what's new in 2023/2024 as far as interesting new approaches go? As the traditional framing of the summarization task, and as a useful basic research assistant function when the user does not yet have focused questions, it is valuable to track progress in this field.

Extractive document summarization involves developing systems that extract key text segments and restructure them to form a representative summary of the original document. The basis by which these text segments are identified and selected are primary areas the field is focused on improving.

2.1.1. Preserve Context Information for Extract-Generate Long-Input Summarization Framework [4]

The authors present an Extract-generate algorithm [5], [6], [7], [8], [9] for long-input summarization. Algorithms in this family use an *extractor* to identify the most important parts of the input and a *generator* to compile these parts into a summary. However, the issue with these frameworks is that there is a loss of context as data propagates through the system. This loss of context occurs when the local context information surrounding an extracted snippet is not transferred to the generator, leading to potential inaccuracies in the output summarization. Longer inputs are particularly susceptible to context loss. To address this, the authors propose a *Context-Aware Extract-Generate* framework (CAEG) for long-input summarization. CAEG identifies sentences with an importance score and pulls in additional context, feeding both the important sentences and the context prompts to the generator.

Context prompts are generated by interpreting the attention mechanisms within a transformer-based model, specifically using the attention rollout method. This method quantifies the flow of information through self-attention layers by recursively multiplying attention weight matrices across layers, treating the model as a directed acyclic graph. The attention rollout scores [10] of the [CLS] token is used to identify the most informative text spans, with a sliding window strategy applied to extract spans with the highest average scores. These high-scoring spans, termed context prompts, are then used to enhance the generation process by providing local context information.

Before passing the local information to the generator, global context information is collected by identifying the most related snippets to an extracted snippet using cosine similarity between snippet representations in the extractor, and adopting context prompts from these related snippets as the global context prompts. These local and global context prompts, along with the extracted snippets, are then fed into the generator to enhance the summarization process.

2.1.2. Unsupervised Extractive Summarization with Learnable Length Control Strategies [11]

This paper presents an unsupervised extractive summarization model, SimSiam, which leverages a transformer-based Siamese network [4] and introduces a differentiable length-control mechanism. Length control summarization has primarily been researched in abstractive summarization tasks and is relatively unexplored in extractive summarization. Unlike traditional importance-scoring methods, which are non-differentiable and may rely on positional assumptions, the proposed model can be trained end-to-end. This is achieved through a bidirectional prediction objective introduced during training.

A bidirectional prediction objective is a mechanism where the model is trained to predict the representation of the original document from the selected summary sentences and to reconstruct the selected summary sentences from the representation of the original document.

The overall objective function for the summarization model combines the extractive summarization loss, which maximizes the relevance of selected sentences within length constraints, and the contrastive learning loss, which enhances the robustness and distinctiveness of sentence representations. These losses are weighted by hyperparameters to balance their contributions, guiding the model to select key sentences for summarization.

The authors address the challenge of length control in extractive summarization by modeling it as a 0-1 knapsack problem [12], [13], [14] and introducing a knapsack transformer to approximate the dynamic programming solver. The knapsack network is trained on 6 million simulated samples reflecting empirical sentence lengths and scores, using Poisson, gamma, and uniform distributions, and learns to maximize sentence importance scores within length constraints through a balanced loss function. The knapsack transformer, combined with Gumbel-soft top-K sampling [15], and the straight-through trick [16], enables the integration of length control into the end-to-end training process.

[12], [13], [14], [15], [16] Experimental results on datasets such as CNNDM, NYT, and CNewSum show the proposed method significantly outperforms (R1, R2, RL) centrality-based methods using the same encoder. Human validation based on relevance, coherence, and consistency was also used to support the effectiveness of the model and indicated that it often surpassed models like BertSum in terms of relevance and consistency.

2.1.3. Compressed Heterogeneous Graph for Abstractive Multi-Document Summarization [17]

The authors propose HGSum for abstractive multi-document summarization (MDS). HGSum extends the encoder-decoder architecture to incorporate compressed heterogeneous graphs consisting of document, sentence, and word nodes. HGSum comprises four main components: a text encoder, a graph encoder, a graph compressor, and a text decoder, with both the text encoder and decoder initialized using PRIMERA [3] weights. While the text encoder leverages the sparse attention mechanism of Longformer (LED) [18] and the text decoder functions as a standard transformer-based decoder, the authors' primary contributions are the enhancements made to the graph encoder and graph compressor.

We note that HGSum builds on previous work [19] and, while it addresses the objective of abstract text summarization, it is part of a larger body of work on Heterogeneous Graph Attention Networks [20], [21], [22], [23], [24], [25].

To address the limitations of standard graph neural networks [26], [27], [28] in handling heterogeneous graphs, the authors introduce a multi-channel graph attention network (MGAT) to encode these graph types. MGAT aggregates embeddings of different channels (i.e., edge types) for each node. The graph compressor computes attention scores for all sentence nodes, removes the lowest scores, and masks the remaining nodes based on their attention scores. The model is trained with two objectives: maximizing the likelihood of generating the ground-truth summary and maximizing the graph similarity between the compressed graph encoding and the ground-truth summary graph encoding. For the first objective, cross-entropy loss is minimized over the ground-truth and generated summaries. For the second objective, cosine similarity of the average node embeddings is computed between the compressed graph and the ground-truth summary graph.

HGSum is evaluated against other recent abstractive MDS models, including PLM-based (PEGASUS [29], LED, PRIMERA) and graph-based (MGSum [30], GraphSum [31]) models. Using benchmark datasets such as MULTI-NEWS [32], WCEP-100 [33], and ARXIV [34], the evaluation metrics include ROUGE-1, ROUGE-2, and ROUGE-L scores. Results indicate that HGSum generally performs well compared to other models. The code and experimental setups are available at [35].

2.1.4. *Efficient Memory-Enhanced Transformer for Long-Document Summarization in Low-Resource Regimes [36]*

The proposed model, EMMA, addresses memory efficiency in long document summarization (LDS) with fixed memory requirements during training and inference. EMMA's primary contributions include applying cross-memory attention to chunks and integrating recurrent short- and long-term memory structures within an encoder-decoder architecture to enhance memory performance.

For LDS, EMMA processes chunks through a chunk-level summarization and concatenates results for a document-level summary. It incorporates recurrent layer-level memory to store past information. The model extends the classical BART encoder by adding cross-memory attention after the residual connection following self-attention. This cross-memory attention comprises a layer-level memory and a secondary attention block, forming a single memory matrix.

While cross-memory attention typically attends to interdependent token relationships across model layers, EMMA leverages it for chunk-level relationship structures, supporting LDS systems. As chunks are processed, cross-memory attention updates the memory matrix without increasing GPU memory usage during training by stopping the gradient. A challenge with this memory structure is the potential loss of long-term details, as the memory matrix is overwritten at each step. To mitigate this, EMMA includes a long-term memory structure, storing the previous memory matrix into a long-term memory matrix at each step.

EMMA is trained using chunk-target pairs, aiming to minimize the negative log-likelihood of predicted tokens. A sentence-level segmentation algorithm constructs chunk-target pairs from long input documents and their target summaries. Documents are segmented into non-overlapping chunks with a defined maximum number of tokens, and each target summary sentence is paired with the chunk that maximizes the ROUGE-1 precision metric, forming small source-target pairs.

Standard ROUGE-1/2/L metrics are used for automatic LDS evaluation. Additionally, the authors computed a metric [37] $R = \frac{\text{avg}(r1,r2,rL)}{1 + \sigma_r^2}$, where σ_r^2 is the ROUGE F1 score variance, penalizing heterogeneous results. Qualitative analysis was performed based on informativeness, fluency, factuality, and succinctness. The main findings were, based on these metrics the system performed well with significantly less memory requirements. Code and evaluations can be found at [38]

2.2. Query- and Workflow-based Systems

Query-Focused Summarization (QFS) aims to produce summaries that answer specific questions of interest, enabling greater user control and personalization [37]. It's a task that brings elements of both Q&A from context, and summarization of lengthy content. As a practical matter for a research assistant, tracking progress in this sort of goal-directed task variation is of interest.

2.2.1. QuerySum: A Multi-Document Query-Focused Summarization Dataset Augmented with Similar Query Clusters [39]

The paper introduces QuerySum, a large-scale dataset designed to support research in Query-Focused Summarization (QFS), particularly emphasizing non-factoid queries such as "What," "Why," and "How" questions, which constitute 74% of the dataset. The authors also propose a model for query-aware summarization, which incorporates self-attention mechanisms tailored for query-awareness.

The query-aware summary generation involves two primary components: semantic-based document pooling and relevance-based inter-document attention. Semantic-based pooling discriminates terms of importance to obtain the query-aware document representation. Relevance-based attention integrates query-document relevance into the inter-document attention layer by using a kernel-based relevance model to produce relevance scores, which modify the dot product attention layer to focus on the most relevant documents during summarization.

The proposed model is based on a transformer architecture with modified self-attention layers in the encoder to account for query-awareness. It employs semantic-based pooling for multi-document representation, where each document is processed through multiple transformer encoders to create a contextualized representation (CR). These CRs are concatenated and pooled to form a document-level representation that is relevant to the query.

The pooled representation is then fed into a relevance-based inter-document attention layer, which consists of a two-layer feed-forward network with a ReLU activation. This layer aggregates and normalizes the cosine similarity between the query and documents using a softmax function. The output of this softmax function serves as weights for summing up the term representations, thereby informing the model of the relevance of each document to the user's query. This process modifies the dot product attention layer and produces an attention score to focus on the most relevant document during summarization.

Finally, the responsive documents are fed into a pretrained query-aware summarization model, PEGASUS. Summaries are evaluated using ROUGE scores, and the overall approach outperforms other methods, including transformer variants with different query and document embedding strategies, as well as other pretrained models like BERTSUM and CTRLSUM.

2.2.2. ChatCite: LLM Agent with Human Workflow Guidance for Comparative Literature Summary [40]

ChatCite is an example of a single-agent, stateful, fixed-workflow summarization process; the authors' goal is to improve on Chain-of-Thought based summarization [41], [42], [43], [44], [45], which they assert leaves out extraction of key elements, lacks comparative analysis, and fails to produce appropriate structure. ChatCite extracts key elements from relevant literature and generates summaries with an iterative multi-step process. Given a set of papers, ChatCite first uses 7 guiding questions and the source content as a prompt to the LLM. This is implemented using Q&A from context on each reference to elicit research questions, methodology, results, conclusions, contributions, innovations, and limitations. This information is stored in memory for iterative generation of a comparative summary. Taking the proposed work description, and the key elements for the first reference paper, ChatCite's "reflective incremental generator" generates *n* candidate literature summaries. For each additional paper these candidates are evaluated and filtered to the best *m* summaries. The new paper's key elements are added to generate a new literature summary from each candidate; each expansion produces *nm* further candidates. After the final reference is added, the optimal candidate is used as the final result. The authors compare ROUGE metrics, human evaluation, and G-Score, an LLM-generated metric inspired by G-Eval [46] to score consistency, coherence, the degree to which they are comparative, their integrity, fluency, and the accuracy of citations.

ChatCite was tested by generating related-work sections for target papers where the relevant cited papers were known, but the workflow could logically be extended with an information retrieval (IR) step. ChatCite's fixed workflow differs from RL or cognitive-workflow strategies in that composition subtasks are not being selected; deciding what to do next is implicit given the key elements and candidate summary in each iteration. Adding IR, explicit compositional subtasks, and self-evaluation would be logical future work.

2.2.3. MALADE: Orchestration of LLM-powered Agents with Retrieval Augmented Generation for Pharmacovigilance [47]

MALADE is an example of a multi-agent, stateful, actor/critic, fixed-workflow pharmacological support tool in which the goal is to generate trustworthy, evidence-based summaries of medical knowledge. Its primary function is to identify adverse drug events (ADEs), through a process of information retrieval and reporting. Each agent in the system is given a specific subtask and is assigned an additional critic agent to verify the task agent's behavior and responses. The three agents are DrugFinder, responsible for drug discovery based on a drug category; DrugAgent, responsible for information retrieval, extraction of side effects, and unstructured summarization of ADEs; and CategoryAgent, responsible for structured report generation. Each agent can take into account critical feedback in its performance of the task and works iteratively until the critic is satisfied. Individual agents were generally encapsulated as a set of string transformation functions, either directly prompting an LLM, eliciting a response from a user, or interacting with a tool using a parser and formatter to work through strings.

One of the findings of this paper was to avoid gratuitous use of LLMs for deterministic tasks; for example, a Toolformer [48] style agent can be used to generate SQL queries for a drug database to get prescription frequencies, but upon inspection in their case the queries were a simple function of the initial list of drugs and could be coded up conventionally, reducing the cost of LLM services while potentially increasing reliability. In contrast, there is no simple algorithm to supplement missing FDA labels associating drugs to outcomes. This type of relationship inference is frequently cast as a graph edge prediction task and a challenging but appropriate pursuit.

2.3. LLM Agent Systems

As we characterize the literature we see content-focused and query-focused summarization approaches as largely one-step processes. As other efforts begin define the summarization task as a multi-step composition process with a variety of subtasks [49], they also begin using LLM-backed autonomous agents and optionally including collaboration with and guidance from a user. While there are few publications on LLM agents applied to the process of writing and composition, this is part of a larger body of work [50] focused on general task orchestration (e.g. [51]) and social simulations (e.g. [52]). Most recently, LLM-backed agent-based systems have been seeking to capitalize on the benefits of reinforcement learning strategies and flexible workflows.

2.3.1. DEBATE: Devil's Advocate-Based Assessment and Text Evaluation [53]

One of the common problems cited in the field of document summarization is the difficulty in evaluating summary quality [54]. DEBATE is an agent-based content evaluation strategy that demonstrates better correlation with human evaluations of summarization and dialogue generation than previous scoring methods. The metric for DEBATE itself is Spearman or Kendall-Tau correlation between the final scores DEBATE produces and human scores – for summarization, correlation was measured with human scores of coherence, consistency, fluency, and relevance.

The authors adopt the experimental design from MacDougall and Baum [55] and include four agents – Commander (leader), Scorer, Critic, and Tie-breaker – implemented with AutoGen [56]. The Scorer agent is instructed to evaluate a summary or other product, and the Critic assesses the other agent's scoring arguments, potentially resolving the bias in LLM used to produce the Scorer's answers. The Commander agent exists to facilitate a debate between the two and bring information in from previous debates stored in memory, and the Tie-breaker will optionally resolve debates that left the Critic unsatisfied in the allotted number of rounds. During experimentation the authors varied the number of iterations in the debate, the positivity of the "persona" used in the Critic's prompts, whether or not to use a persona that was explicitly described as a "Devil's Advocate", and whether or not to allow Tie-breakers. The best performance (i.e. correlation with human evaluations) across all but one summary evaluation criteria was shown by using Devil's Advocate prompts and the Tie-breaker agent, with all agents using GPT-4 [57]. The exception was using GPT-4 to implement MultiAgent (the same architecture with a plain persona instead of Devil's Advocate), measured with Spearman correlation of human evaluation of Engagingness on Topical-Chat results.

2.3.2. Reflexion: Language Agents with Verbal Reinforcement Learning [58]

Reflexion agents take feedback signals from task performance and build a memory designed to help optimize subsequent decisions. The "action policy" maps from the system state and agent memory to generate text and decisions, and memories are generated as textual feedback explicitly judging why past actions scored the way they did. As a result the model learns to produce text that leads to good decisions through an implicit policy [59] without model fine tuning. While this system uses a single memory pool it would be compatible with to M-RAG's strategy of partitioning and revising memories [60].

The system uses three agents: Actor, Evaluator, and Self-Reflection.

- The actor is an LLM used as f(state, memory) → text, actions. This was implemented with Chain-of-Thought [41] and ReAct [61] as options during testing.
- The evaluator takes the short-term-memory "trajectory" of recently generated state/text/action sequences from the Actor and generates a reward score in the style of reinforcement learning. This scoring function is domain and task-specific, and the authors used different scoring methods for code generation and decision-making tests.
- The self-reflection model generates nuanced and specific feedback justifying the score (comparable to DEBATE [53], which used "devil's advocate" criticism of scores to improve the scoring rather than subsequent actions). This goes into the long term memory that shapes the actor's actions.

In principle, reflection can pick up on what caused poor scores, it will generate memories that steer the actor away from bad actions in the future.

2.3.3. Llama3+Crew+Groq = Text Summarization Agent [62]

As an honorable mention, this article is meant as a simple technical demonstration, not a peer-reviewed publication, and does not provide comparative performance metrics, but it is a practical example of off-the-shelf agent-based frameworks applied to summarization. Architectures such as this may pave the way for novel methods of deconstructing writing and composition as well as other research support tasks.

The author presents a multi-agent, stateful, fixed-workflow summarization process; the workflow steps are implemented with single-function agents, organized around a shared state, and managed using the CrewAI framework. The agents execute summarization tasks using cloud LLM services through Groq (vs. OpenAI, Cohere, Azure, or other LLM compute providers). More importantly, this kind of architecture could provide the underpinnings of advanced agents as described in the Section 4.4 below.

3. COMPARATIVE ANALYSIS

Like other machine learning tasks, document summarization has proceeded over the last several decades from hard-coding perceived patterns to describing the desired product, and from hard-coding the process to describing the desired intermediate products.

- Early published work used hard-coded rules of thumb for scoring content that might be important and implemented the process of building summaries from this content.
- Scoring was gradually taken over by neural networks, as was then generation and validation of summaries, training systems on representative data points.
- Large language models were then used to generate summaries with no explicit process and no *a priori* labeled data.
- Today, some systems are providing more detailed guidance in the form of natural language, "programming" the large general purpose language models with domain-specific questions and step-by-step instructions. These also depend less on perfect internal embeddings in favor of working with observable external memory.

We have seen only a few examples of this last step, so it is too early to assert that breaking down the problem will provide performance improvements, but it does provide a more auditable production process that is relevant to AI security & reliability.

Leveraging "natural language programming" style implementation of document summarization and other tasks, there are several areas of innovation we see in the field: domain-specific summarization methods (such as adapting the task to account for the niche needs of medicine, law, etc.) and new evaluation methods that are better aligned with human assessments.

Breaking problems such as summarization down also paves the way for reinforcement learning and interactive, collaborative solutions. Agent based systems have begun to emerge that demonstrate both functional agents (such as MALADE and ChatCite) as well as autonomous peer-to-peer agent architectures that capitalize on LLM chat systems (such as DEBATE).

4. FUTURE WORK

4.1. Explainable AI for Summarization

Gaps in the literature point to opportunities for more "explainable" document summarization. It is not a given that when large language models are prompted interactively through a dialog or given "natural" writing and composition subtasks to perform, that they will faithfully follow the process in a way a person expects and produce results that change systematically in response to sensible changes in the input. To unpack "explainable AI" more explicitly in this context, it is necessary both to improve alignment of the overall process with human expectations and do in ways that are not misleading. This motivates research in both human task analysis to inform workflow breakdowns and natural language prompts that are accurate representations of how people comprehend and summarize content, as well as research into the causal mechanisms that govern LLM performance and how those mechanisms apply to judgement-heavy tasks such as summarization.

4.2. Collaborative Summarization

Gaps in the literature point to opportunities for more interactive human-agent collaboration for summarization. This could be approached as an interactive extension for query-focused summarization building on [63], or continued experimentation with autonomous agents. There are likely domains that would benefit from users iteratively engaging in the writing process at smaller scales leveraging enormous amounts of content (such as national security reporting in the service of decisions made by a small number of executives), without needing a completely automated process that generates an enormous number of summaries of small portions of content (such as summarizing all books, movies, etc. for larger groups that have no specific focus of attention).

4.3. Variable-Scope Summarization

The current limits of document summarization suggest opportunities that would advance STEM process support in general. Summarization of content at a high level is often used in STEM to filter what is given further attention, starting with a brief discussion establishing context across multiple topics, and then increasingly detailed summaries of more specialized and focused information, ultimately supporting development of step-by-step instructions that only "summarize" across repeated performance of tasks that have inherent variability. We have not seen summarization literature that attempts to afford an end user control of summaries of increasing detail in a narrowing domain, but it is a logical extension of the concept incorporating most of the same tasks.

4.4. Integrated Autonomous Agent

There are a number of LLM-based solutions that could readily be combined to take on not just summarization but any number of other tasks as well. Just as an integration effort – which would presumably lead to innovation in turn – one could envision combining the following components into a single architecture:

- Workflow and action space descriptions to provide human-aligned schemas for agent operation, guided by work in psychology such as [49].
- Self-reflection to maintain memories that steer the system away from past mistakes and towards the merits of past solutions. [58]
- Devil's advocate evaluation [53] and reinforcement learning from human feedback [64] to ensure internal scoring of actions is well correlated with human scores.

- Repurposing “jailbreaking” algorithms to optimize memories s.t. in hindsight better actions would have been taken. [65], [66], [67]
- Partitioning and ongoing refinement of memory pools, that allow state-aware selection of relevant guiding memories. [60]
- Thinking through actions before final commit to each action. [47]

REFERENCES

[1] A. L. Brown, J. D. Day, and R. S. Jones, “The Development of Plans for Summarizing Texts,” *Child Dev.*, vol. 54, no. 4, pp. 968–979, 1983, doi: 10.2307/1129901.

[2] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Text,” presented at the Conference on Empirical Methods in Natural Language Processing, Jul. 2004.

[3] W. Xiao, I. Beltagy, G. Carenini, and A. Cohan, “PRIMERA: Pyramid-based Masked Sentence Pre-training for Multi-document Summarization,” Mar. 16, 2022, *arXiv*: arXiv:2110.08499.

[4] R. Yuan, Z. Wang, Z. Cao, and W. Li, “Preserve Context Information for Extract-Generate Long-Input Summarization Framework,” *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, Art. no. 11, Jun. 2023, doi: 10.1609/aaai.v37i11.26631.

[5] P. Cui and L. Hu, “Sliding selector network with dynamic memory for extractive summarization of long documents,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5881–5891. Accessed: Aug. 22, 2024. [Online]. Available: https://aclanthology.org/2021.naacl-main.470/

[6] W. Xiao and G. Carenini, “Extractive Summarization of Long Documents by Combining Global and Local Context,” Sep. 17, 2019, *arXiv*: arXiv:1909.08089. Accessed: Aug. 22, 2024. [Online]. Available: http://arxiv.org/abs/1909.08089

[7] A. Gidiotis and G. Tsoumakas, “A divide-and-conquer approach to the summarization of long documents,” *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 3029–3040, 2020.

[8] Z. Mao *et al.*, “DYLE: Dynamic Latent Extraction for Abstractive Long-Input Summarization,” Apr. 24, 2022, *arXiv*: arXiv:2110.08168. Accessed: Aug. 22, 2024. [Online]. Available: http://arxiv.org/abs/2110.08168

[9] Y. Zhang *et al.*, “An Exploratory Study on Long Dialogue Summarization: What Works and What’s Next,” Sep. 09, 2021, *arXiv*: arXiv:2109.04609. Accessed: Aug. 22, 2024. [Online]. Available: http://arxiv.org/abs/2109.04609

[10] S. Abnar and W. Zuidema, “Quantifying attention flow in transformers,” *ArXiv Prepr. ArXiv200500928*, 2020, Accessed: Aug. 20, 2024. [Online]. Available: https://arxiv.org/abs/2005.00928

[11] R. Jie, X. Meng, X. Jiang, and Q. Liu, “Unsupervised Extractive Summarization with Learnable Length Control Strategies,” *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 16, Art. no. 16, Mar. 2024, doi: 10.1609/aaai.v38i16.29797.

[12] S. Martello and P. Toth, “Upper Bounds and Algorithms for Hard 0-1 Knapsack Problems,” *Oper. Res.*, vol. 45, no. 5, pp. 768–778, Oct. 1997, doi: 10.1287/opre.45.5.768.

[13] S. Martello and P. Toth, “Lower bounds and reduction procedures for the bin packing problem,” *Discrete Appl. Math.*, vol. 28, no. 1, pp. 59–70, 1990.

[14] S. Martello and P. Toth, *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc., 1990. Accessed: Aug. 19, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/98124

[15] W. Kool, H. van Hoof, and M. Welling, “Estimating Gradients for Discrete Random Variables by Sampling without Replacement,” Feb. 14, 2020, *arXiv*: arXiv:2002.06043. Accessed: Aug. 19, 2024. [Online]. Available: http://arxiv.org/abs/2002.06043

[16] Y. Bengio, N. Léonard, and A. Courville, “Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation,” Aug. 15, 2013, *arXiv*: arXiv:1308.3432. Accessed: Aug. 19, 2024. [Online]. Available: http://arxiv.org/abs/1308.3432

[17] M. Li, J. Qi, and J. H. Lau, “Compressed Heterogeneous Graph for Abstractive Multi-Document Summarization,” *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 11, Art. no. 11, Jun. 2023, doi: 10.1609/aaai.v37i11.26537.

[18] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The Long-Document Transformer,” *ArXiv*, Apr. 2020, Accessed: Sep. 21, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Longformer%3A-The-Long-Document-Transformer-Beltagy-Peters/71b6394ad5654f5cd0fba763768ba4e523f7bbca

[19] X. Wang *et al.*, “Heterogeneous Graph Attention Network,” in *The World Wide Web Conference*, San Francisco CA USA: ACM, May 2019, pp. 2022–2032. doi: 10.1145/3308558.3313562.

[20] H. Linmei, T. Yang, C. Shi, H. Ji, and X. Li, “Heterogeneous graph attention networks for semi-supervised short text classification,” in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 4821–4830. Accessed: Aug. 21, 2024. [Online]. Available: https://aclanthology.org/D19-1488/

[21] X. Yang, C. Deng, T. Liu, and D. Tao, “Heterogeneous graph attention network for unsupervised multiple-target domain adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1992–2003, 2020.

[22] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, “Disenhan: Disentangled heterogeneous graph attention network for recommendation,” in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1605–1614. Accessed: Aug. 21, 2024. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3340531.3411996

[23] Q. Li, Y. Shang, X. Qiao, and W. Dai, “Heterogeneous dynamic graph attention network,” in *2020 IEEE international conference on knowledge graph (ICKG)*, IEEE, 2020, pp. 404–411. Accessed: Aug. 21, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9194495/

[24] X. Mo, Z. Huang, Y. Xing, and C. Lv, “Multi-agent trajectory prediction with heterogeneous edge-enhanced graph attention network,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9554–9567, 2022.

[25] T. Yang, L. Hu, C. Shi, H. Ji, X. Li, and L. Nie, “HGAT: Heterogeneous graph attention networks for semi-supervised short text classification,” *ACM Trans. Inf. Syst. TOIS*, vol. 39, no. 3, pp. 1–29, 2021.

[26] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, “A comprehensive survey on graph neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2020.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2008.

[28] J. Zhou *et al.*, “Graph neural networks: A review of methods and applications,” *AI Open*, vol. 1, pp. 57–81, 2020.

[29] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "PEGASUS: pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, in ICML'20, vol. 119. JMLR.org, Jul. 2020, pp. 11328–11339. Accessed: Sep. 24, 2023. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/3524938.3525989

[30] H. Jin, T. Wang, and X. Wan, "Multi-granularity interaction network for extractive and abstractive multi-document summarization," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 6244–6254. Accessed: Aug. 20, 2024. [Online]. Available: https://aclanthology.org/2020.acl-main.556/

[31] W. Li, X. Xiao, J. Liu, H. Wu, H. Wang, and J. Du, "Leveraging Graph to Improve Abstractive Multi-Document Summarization," May 20, 2020, *arXiv*: arXiv:2005.10043. Accessed: Aug. 20, 2024. [Online]. Available: http://arxiv.org/abs/2005.10043

[32] A. R. Fabbri, I. Li, T. She, S. Li, and D. R. Radev, "Multi-News: a Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model," Jun. 19, 2019, *arXiv*: arXiv:1906.01749. Accessed: Aug. 20, 2024. [Online]. Available: http://arxiv.org/abs/1906.01749

[33] D. G. Ghalandari, C. Hokamp, N. T. Pham, J. Glover, and G. Ifrim, "A Large-Scale Multi-Document Summarization Dataset from the Wikipedia Current Events Portal," May 20, 2020, *arXiv*: arXiv:2005.10070. Accessed: Aug. 20, 2024. [Online]. Available: http://arxiv.org/abs/2005.10070

[34] C. B. Clement, M. Bierbaum, K. P. O'Keeffe, and A. A. Alemi, "On the Use of ArXiv as a Dataset," Apr. 30, 2019, *arXiv*: arXiv:1905.00075. Accessed: Aug. 20, 2024. [Online]. Available: http://arxiv.org/abs/1905.00075

[35] M. Li, *oaimli/HGSum*. (Aug. 12, 2024). Python. Accessed: Aug. 20, 2024. [Online]. Available: https://github.com/oaimli/HGSum

[36] G. Moro, L. Ragazzi, L. Valgimigli, G. Frisoni, C. Sartori, and G. Marfia, "Efficient memory-enhanced transformer for long-document summarization in low-resource regimes," *Sensors*, vol. 23, no. 7, p. 3542, 2023.

[37] G. Moro, L. Ragazzi, and L. Valgimigli, "Carburacy: Summarization Models Tuning and Comparison in Eco-Sustainable Regimes with a Novel Carbon-Aware Accuracy," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 12, Art. no. 12, Jun. 2023, doi: 10.1609/aaai.v37i12.26686.

[38] D. U. NLP, *disi-unibo-nlp/emma*. (Jul. 27, 2023). Accessed: Aug. 22, 2024. [Online]. Available: https://github.com/disi-unibo-nlp/emma

[39] Y. Liu, Z. Wang, and R. Yuan, "QuerySum: A Multi-Document Query-Focused Summarization Dataset Augmented with Similar Query Clusters," *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 17, Art. no. 17, Mar. 2024, doi: 10.1609/aaai.v38i17.29836.

[40] Y. Li, L. Chen, A. Liu, K. Yu, and L. Wen, "ChatCite: LLM Agent with Human Workflow Guidance for Comparative Literature Summary," Mar. 04, 2024, *arXiv*: arXiv:2403.02574. doi: 10.48550/arXiv.2403.02574.

[41] J. Wei *et al.*, "Chain of Thought Prompting Elicits Reasoning in Large Language Models," *ArXiv*, Jan. 2022, Accessed: Jan. 28, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Chain-of-Thought-Prompting-Elicits-Reasoning-in-Wei-Wang/1b6e810ce0afd0dd093f789d2b2742d047e316d5

[42] X. Wang *et al.*, “Self-Consistency Improves Chain of Thought Reasoning in Language Models,” 2022, doi: 10.48550/ARXIV.2203.11171.

[43] G. Adams, A. Fabbri, F. Ladhak, E. Lehman, and N. Elhadad, “From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting,” Sep. 08, 2023, *arXiv*: arXiv:2309.04269. doi: 10.48550/arXiv.2309.04269.

[44] M. Turpin, J. Michael, E. Perez, and S. R. Bowman, “Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting,” Dec. 09, 2023, *arXiv*: arXiv:2305.04388. doi: 10.48550/arXiv.2305.04388.

[45] G. Bao, H. Zhang, L. Yang, C. Wang, and Y. Zhang, “LLMs with Chain-of-Thought Are Non-Causal Reasoners,” Feb. 25, 2024, *arXiv*: arXiv:2402.16048. doi: 10.48550/arXiv.2402.16048.

[46] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, “G-Eval: NLG Evaluation using Gpt-4 with Better Human Alignment,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds., Singapore: Association for Computational Linguistics, Dec. 2023, pp. 2511–2522. doi: 10.18653/v1/2023.emnlp-main.153.

[47] J. Choi *et al.*, “MALADE: Orchestration of LLM-powered Agents with Retrieval Augmented Generation for Pharmacovigilance,” Aug. 2024. Accessed: Aug. 19, 2024. [Online]. Available: https://www.semanticscholar.org/paper/MALADE%3A-Orchestration-of-LLM-powered-Agents-with-Choi-Palumbo/ebe9756678c804f9d2d0e053354cee5e2bffa073

[48] T. Schick *et al.*, “Toolformer: Language models can teach themselves to use tools,” *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, Accessed: Aug. 22, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html

[49] L. Flower and J. R. Hayes, “A Cognitive Process Theory of Writing,” *Coll. Compos. Commun.*, vol. 32, no. 4, p. 365, Dec. 1981, doi: 10.2307/356600.

[50] L. Wang *et al.*, “A Survey on Large Language Model based Autonomous Agents,” *Front. Comput. Sci.*, vol. 18, no. 6, p. 186345, Dec. 2024, doi: 10.1007/s11704-024-40231-1.

[51] Y. Qin *et al.*, “ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs,” Oct. 03, 2023, *arXiv*: arXiv:2307.16789. doi: 10.48550/arXiv.2307.16789.

[52] J. S. Park, J. C. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, “Generative Agents: Interactive Simulacra of Human Behavior,” Aug. 05, 2023, *arXiv*: arXiv:2304.03442. doi: 10.48550/arXiv.2304.03442.

[53] A. Kim, K. Kim, and S. Yoon, “DEBATE: Devil’s Advocate-Based Assessment and Text Evaluation,” 2024, doi: 10.48550/ARXIV.2405.09935.

[54] M. F. Mridha, A. Lima, K. Nur, S. Das, M. Hasan, and M. Kabir, “A Survey of Automatic Text Summarization: Progress, Process and Challenges,” *IEEE Access*, vol. 9, Nov. 2021, doi: 10.1109/ACCESS.2021.3129786.

[55] C. MacDougall and F. Baum, “The Devil’s Advocate: A Strategy to Avoid Groupthink and Stimulate Discussion in Focus Groups,” *Qual. Health Res. - QUAL Health RES*, vol. 7, pp. 532–541, Nov. 1997, doi: 10.1177/104973239700700407.

[56] Q. Wu *et al.*, “AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation,” Oct. 03, 2023, *arXiv*: arXiv:2308.08155. doi: 10.48550/arXiv.2308.08155.

[57] OpenAI *et al.*, “GPT-4 Technical Report,” Dec. 18, 2023, *arXiv*: arXiv:2303.08774. doi: 10.48550/arXiv.2303.08774.

[58] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, “Reflexion: Language Agents with Verbal Reinforcement Learning,” Oct. 10, 2023, *arXiv*: arXiv:2303.11366. doi: 10.48550/arXiv.2303.11366.

[59] E. Brooks, L. Walls, R. Lewis, and S. Singh, *In-Context Policy Iteration*. 2022. doi: 10.48550/arXiv.2210.03821.

[60] Z. Wang, S. X. Teo, J. Ouyang, Y. Xu, and W. Shi, “M-RAG: Reinforcing Large Language Model Performance through Retrieval-Augmented Generation with Multiple Partitions,” 2024, doi: 10.48550/ARXIV.2405.16420.

[61] S. Yao *et al.*, “ReAct: Synergizing Reasoning and Acting in Language Models,” Mar. 09, 2023, *arXiv*: arXiv:2210.03629. doi: 10.48550/arXiv.2210.03629.

[62] T. Dave, “Llama3+Crew+Groq = Text Summarization Agent ✍️ ,” Medium. Accessed: Aug. 19, 2024. [Online]. Available: https://blog.gopenai.com/llama3-crew-groq-text-summarization-agent-%EF%B8%8F-1d1ed9ac9937

[63] Z. Wang, B. Gan, and W. Shi, “Multimodal Query Suggestion with Multi-Agent Reinforcement Learning from Human Feedback,” in *Proceedings of the ACM on Web Conference 2024*, in WWW ’24. New York, NY, USA: Association for Computing Machinery, May 2024, pp. 1374–1385. doi: 10.1145/3589334.3645365.

[64] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[65] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson, “Universal and Transferable Adversarial Attacks on Aligned Language Models,” Jul. 27, 2023, *arXiv*: arXiv:2307.15043. Accessed: Oct. 17, 2023. [Online]. Available: http://arxiv.org/abs/2307.15043

[66] G. Deng *et al.*, “MasterKey: Automated Jailbreak Across Multiple Large Language Model Chatbots,” in *Proceedings 2024 Network and Distributed System Security Symposium*, 2024. doi: 10.14722/ndss.2024.24188.

[67] J. Yu, X. Lin, Z. Yu, and X. Xing, “GPTFUZZER: Red Teaming Large Language Models with Auto-Generated Jailbreak Prompts,” Oct. 04, 2023, *arXiv*: arXiv:2309.10253. Accessed: Oct. 17, 2023. [Online]. Available: http://arxiv.org/abs/2309.10253

[68] J. Pilault, R. Li, S. Subramanian, and C. Pal, “On Extractive and Abstractive Neural Document Summarization with Transformer Language Models,” presented at the EMNLP 2020, Online: Association for Computational Linguistics, Nov. 2020, pp. 9308–9319. doi: 10.18653/v1/2020.emnlp-main.748.

[69] Y. Dong, A. Mircea, and J. C. K. Cheung, “Discourse-Aware Unsupervised Summarization of Long Scientific Documents,” Jan. 13, 2021, *arXiv*: arXiv:2005.00513. Accessed: Sep. 21, 2023. [Online]. Available: http://arxiv.org/abs/2005.00513

[70] H. P. Edmundson, “New Methods in Automatic Extracting,” *J. ACM*, vol. 16, no. 2, pp. 264–285, Apr. 1969, doi: 10.1145/321510.321519.

[71] L. Huang, L. Wu, and L. Wang, "Knowledge Graph-Augmented Abstractive Summarization with Semantic-Driven Cloze Reward," May 03, 2020, arXiv:2005.01159. doi: 10.48550/arXiv.2005.01159.

[72] T. Chen, X. Wang, T. Yue, X. Bai, C. X. Le, and W. Wang, "Enhancing Abstractive Summarization with Extracted Knowledge Graphs and Multi-Source Transformers," *Appl. Sci.*, vol. 13, no. 13, p. 7753, Jan. 2023, doi: 10.3390/app13137753.

[73] A. Slobodkin, P. Roit, E. Hirsch, O. Ernst, and I. Dagan, "Controlled Text Reduction," Oct. 24, 2022, *arXiv*: arXiv:2210.13449. doi: 10.48550/arXiv.2210.13449.

[74] D. Miller, "Leveraging BERT for Extractive Text Summarization on Lectures," Jun. 07, 2019, *arXiv*: arXiv:1906.04165. doi: 10.48550/arXiv.1906.04165.

[75] D. R. Radev, H. Jing, and M. Budzikowska, "Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies," *NAACL-ANLP 2000 Workshop Autom. Summ.* -, vol. 4, pp. 21–30, 2000, doi: 10.3115/1117575.1117578.

[76] G. Erkan and D. R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, Dec. 2004, doi: 10.1613/jair.1523.

[77] O. Ernst *et al.*, "Summary-Source Proposition-level Alignment: Task, Datasets and Supervised Baseline," in *Proceedings of the 25th Conference on Computational Natural Language Learning*, Online: Association for Computational Linguistics, Nov. 2021, pp. 310–322. doi: 10.18653/v1/2021.conll-1.25.

[78] P. J. Liu *et al.*, "Generating Wikipedia by Summarizing Long Sequences," presented at the International Conference on Learning Representations, Feb. 2018. Accessed: Sep. 25, 2023. [Online]. Available: https://openreview.net/forum?id=Hyg0vbWC-

[79] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Bidirectional encoder representations from transformers," 2016.

[80] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," Feb. 04, 2018, *arXiv*: arXiv:1710.10903. doi: 10.48550/arXiv.1710.10903.

[81] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating Embeddings for Modeling Multi-relational Data," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2013. Accessed: Sep. 30, 2023. [Online]. Available: https://papers.nips.cc/paper_files/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html

[82] H. P. Luhn, "The Automatic Creation of Literature Abstracts," *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, Apr. 1958, doi: 10.1147/rd.22.0159.

[83] J. Carbonell and J. Goldstein, "The use of MMR, diversity-based reranking for reordering documents and producing summaries," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, Melbourne Australia: ACM, Aug. 1998, pp. 335–336. doi: 10.1145/290941.291025.

[84] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Comput. Netw.*, vol. 30, pp. 107–117, 1998.

[85] J. M. Kleinberg, "Hubs, authorities, and communities," *ACM Comput. Surv.*, vol. 31, no. 4es, pp. 5-es, Dec. 1999, doi: 10.1145/345966.345982.

[86] P. J.-J. Herings, G. van der Laan, and D. Talman, "Measuring the Power of Nodes in Digraphs," Oct. 05, 2001, *Rochester, NY*: 288088. doi: 10.2139/ssrn.288088.

[87] R. Arora and B. Ravindran, "Latent Dirichlet Allocation and Singular Value Decomposition Based Multi-document Summarization," presented at the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy: IEEE, Dec. 2008. doi: 10.1109/ICDM.2008.55.

[88] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J Mach Learn Res*, vol. 3, pp. 993–1022, 2003, doi: 10.1016/B978-0-12-411519-4.00006-9.

[89] M. G. Ozsoy, F. N. Alpaslan, and I. Cicekli, "Text summarization using Latent Semantic Analysis," *J. Inf. Sci.*, vol. 37, no. 4, pp. 405–417, Aug. 2011, doi: 10.1177/0165551511408848.

[90] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," *Proc. 43rd Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 39–48, Jul. 2020, doi: 10.1145/3397271.3401075.

[91] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. doi: 10.18653/v1/K16-1028.

[92] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *CoRR*, Sep. 2014, Accessed: Aug. 28, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Neural-Machine-Translation-by-Jointly-Learning-to-Bahdanau-Cho/fa72afa9b2cbc8f0d7b05d52548906610ffbb9c5

[93] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 11, 2014, *arXiv*: arXiv:1412.3555. Accessed: Sep. 30, 2023. [Online]. Available: http://arxiv.org/abs/1412.3555

[94] S. Jean, K. Cho, R. Memisevic, and Y. Bengio, "On Using Very Large Target Vocabulary for Neural Machine Translation," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 1–10. doi: 10.3115/v1/P15-1001.

[95] L. Liu, Y. Lu, M. Yang, Q. Qu, J. Zhu, and H. Li, "Generative Adversarial Network for Abstractive Text Summarization," Nov. 26, 2017, *arXiv*: arXiv:1711.09357. doi: 10.48550/arXiv.1711.09357.

[96] I. J. Goodfellow *et al.*, "Generative adversarial networks," *Commun. ACM*, vol. 63, pp. 139–144, 2014.

[97] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," Apr. 25, 2017, *arXiv*: arXiv:1704.04368. Accessed: Sep. 21, 2023. [Online]. Available: http://arxiv.org/abs/1704.04368

[98] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Sep. 02, 2014, *arXiv*: arXiv:1408.5882. doi: 10.48550/arXiv.1408.5882.

APPENDIX A. PRE-LLM APPROACHES

This section reviews selected pre-LLM research addressing longer documents, capitalizing on content details, or taking on subtasks of summarization. Authors tried to capitalize on some of the following:

- Document Structure – [68], [69] build on the legacy of work such as [70], exploiting informative patterns in document structure such as headings or "topic sentences". These results suggest that in feature-enhanced models, extractive summary and structural cues may remain valuable.
- Content Structure – [3], [71], [72] guide summarization with more explicit information extraction techniques, particularly entity and relationship extraction.
- Variation in Task Breakdown – of the papers surveyed, many scored [2], [3], [29], [69], [73] individual passages in the source material (either as part of training or input preprocessing), or embedded, clustered, and filtered passages [74], [75], [76] before composing the final summary either trivially (for extractive summarization) or using more flexible text generation methods (for abstractive summarization).
- Human Valuation – In [73] the authors note the relationship of highlighting to extractive summary scoring [68], [74], propositional alignment [77], and the above content structure based techniques. This work is well aligned with reinforcement learning with human feedback (RLHF, [64]), which we see as a critical niche subtask for continued progress in summarization.

End-to-end solutions make summarization straightforward, but limit transparency if a model does not produce faithful or complete summary text. Several papers [3], [29], [69], [73], have emphasized ROUGE's limitations and relied on human evaluations to perform more nuanced scoring of content and fluency.

A.1. Handling More Content

Liu et al. 2018 [78] This paper approaches Wikipedia article generation as a multi-document summarization problem using a two-stage extractive-abstractive framework. Initially, information is coarsely extracted from source web pages. The extracted text is concatenated, putting the most relevant sentences first. This extractive phase is treated as an Information Retrieval task to identify and rank salient sentences. The second stage involves a sequence transduction problem, where a decoder-only Transformer processes long documents and generates summaries of user-specified lengths. The authors introduce a novel architecture, T-DMCA, which uses memory-compressed attention to handle longer sequences efficiently through strided convolutions.

Zhang et al. 2020 [29] PEGASUS introduces a pre-trained abstractive summarization model. The model uses a novel self-supervised objective, Gap Sentence Generation (GSG), where key sentences are masked and reconstructed. This method outperforms traditional masked-language modeling (MLM). The pre-trained model is fine-tuned on specific summarization datasets, with the most salient gap sentences identified using a compound ROUGE score. The authors test two BERT architectures (Base and Large) and six GSG variants, noting that pretrained models transfer better when domains are aligned. They also highlight the input size limitations (at the time) of transformer architectures.

Dong et al. 2021 [69] This paper introduces HipoRank, a scoring method for extractive summarization that leverages sentence position and sentence-section similarity. HipoRank prioritizes sentences near the beginning and end of sections, as well as those similar to section content. They focus on summarizing scientific articles, emphasizing the importance of preserving original text in legal and medical documents. Sentences are embedded using pretrained transformers such as [79].

Section representations are built with mean pooling, and sentence importance is calculated based on their position and similarity to section representations. The highest scoring sentences are selected until a predefined word limit is reached.

A.2. Capitalizing on Structure

Huang et al. 2020 [71] This paper combines a sequential document encoder and a knowledge graph encoder, for abstractive summarization. The graph encoder capitalizes on entity and relation extraction across multiple sentences and events, for document-level context and for paragraph-level context. The sequential encoder integrates a BiLSTM and attention layer with RoBERTa, while the decoder uses a single-layer LSTM. Training involves a new objective function based on a multiple-choice cloze test and reinforcement learning with a "multiple choice cloze reward." The model performs slightly worse than fine-tuned BART but better than other summarizers.

Xiao et al. 2022 [3] Presents PRIMERA, a foundational model optimized for summarization, which can be fine-tuned or integrated into other systems. This builds on PEGASUS but uses more explicit extracted information. The model is initialized with a pre-trained Longformer-Encoder-Decoder and retrained for summarization using clusters of related documents, and frequently mentioned entities are used as a sign of relevance. Sentences with the highest ROUGE score across document clusters are selected for training, making the model unsupervised. In testing it achieved high ROUGE scores and favorable human evaluations for precision, recall, F1 of factual content, grammar, clarity, and coherence.

Chen et al. 2023 [72] This paper addresses the issue of fictitious or incorrect information in abstractive summarizations by leveraging knowledge graphs to augment text embeddings before text generation. Knowledge graph embeddings – generated using a Graph Attention Network (GAT, [80]) with (subject, predicate, object) triplets – capture entity connections and global context. These embeddings are linked to text embeddings using the TranE method [81] and combined for input into a fine-tuned BART model. While the results are not superior to BART, testing suggests the Wikipedia dataset presents additional challenges vs. news article summarization, highlighting the impact of content density and organization.

A.3. Summarization Subtasks

Miller 2019 [74] This paper presents an extractive summarization method for lecture transcripts that requires no training or fine-tuning. Sentences are embedded using pre-trained transformers, and K-means clustering identifies 'centroid' sentences for summarization. The author finds that averaging the word embeddings produces better sentence representations than using the [CLS] token's embedding. Experiments with BERT, GPT-2, and an ensemble show the ensemble performs best, but BERT's second-to-last layer is recommended for efficiency. The method faces challenges with indirect references and was outperformed by other approaches.

Pilault et al. 2020 [68] This paper proposes summarizing long documents in two stages, using structure-informed extractive summarization to prompt an abstractive summarizer. Using technical papers, it generates extractive summaries of the abstract, introduction, and body, then inputs these into a transformer in a specific order. Two extractive models are tested: a hierarchical seq2seq sentence pointer, and a sentence classifier. The hierarchical model performs better on arXiv data, while the classifier excels on PubMed data. The approach outperforms several legacy models, but BART was not tested. The key contribution is using reordered extractive summaries as supplemental input for abstractive summarization.

Slobodkin et al. 2022 [73] This paper explores separating document summarization into two tasks: identifying discrete pieces of information (highlights) and generating a summary from these highlights. This approach allows for more modular text processing and better optimization of subtasks. The authors use the Longformer Encoder-Decoder (LED) model with global attention tied to supplemental "tag" tokens marking highlighted text spans. Experiments show that LED can generate coherent summaries that include only the highlighted facts, without extraneous details. MTurker was used to produce gold standard highlights and the system was scored for quality using fact precision and recall.

APPENDIX B. LEGACY APPROACHES

While the most highly performing approaches today are dominated by transformers, understanding early feature-extraction-driven text summarization provides useful context.

Luhn 1958 [82] Pulls statistically interesting sentences into an "auto-abstract." Favors "significant" words vs. very frequent or rare words, and scores significant word clusters as (n^2)/c where n is the number of significant words within a given window size of each other and c is the cluster size (cluster can exceed a window with chaining).

Edmundson 1969 [70] Selection of sentences for a document summary using 3 new components that are superior to frequent words: pragmatic "cue" words, title and heading words, and structural information from sentence location.

Carbonell 1998 [83] Maximum margin relevance (MMR) is used to produce summaries of single documents that avoid redundancy. This is for retrieval and summarization; hi marginal relevance means a document is relevant to a query and contains minimal similarity to previously selected documents.

Radev 2000 [75] MEAD; generation of a single summary of multiple documents (an event cluster pulled from a topic detection and tracking system), using cluster centroids produced by a topic detection and tracking system and two new techniques: cluster-based sentence utility (CBSU, the relevance of a sentence to a cluster) and cross-sentence informational subsumption (CSIS, to eliminate redundancy).

Erkan 2004 [76] Extractive text summarization using LexRank, which computes sentence importance as eigenvector centrality in a graph of sentences. The graph is an adjacency matrix weighted with intra-sentence cosine similarity. Sentences are represented as bag-of-word TFIDF vectors.

Mihalcea 2004 [2] TextRank is a graph-based ranking model for text processing, using PageRank [84] or alternatively HITS [85] or Positional Function [86]. Sentences are the nodes in the graph; sentences "vote" for one another based on similarity, in this case word overlap normalized by sentence length. The highest ranked sentences are used in extractive summaries.

Arora 2008 [87] Latent Dirichlet allocation [88] applied to multi-document summarization. LDA is used to build a Bayesian topic model jointly modeling document topic frequency and topic word frequency, and then singular value decomposition on sentence vectors allows finding sentences that are orthogonal to one another (reducing redundancy) and best represent the topics.

Ozsoy 2011 [89] Latent semantic analysis strategies use singular value decomposition of a word/sentence co-occurrence matrix for finding semantically similar words and sentences. The authors further extend this to either reduce the value of multi-topic sentences or to focus summarization on primary topics. These methods are evocative of the similarity scores used for information retrieval in ColBERT [90].

Nallapati 2016 [91] Abstractive summarization using attentional Recurrent Neural Network (RNN) encoder-decoder models [92]. Initial model is a bidirectional gated recurrent unit GRU-RNN [93] using the "large vocabulary trick" [94]. The authors also tried augmented embeddings in a switching decoder/pointer architecture; or using a bidirectional RNN with word- and sentence-level attention.

Liu 2017 [95] Uses a Generative Adversarial Network [96], and an abstractive summary generator – comprising a bidirectional LSTM encoder and an attention-based LSTM decoder per [97] – with a CNN text classifier [98] discriminator model trained to classify summaries as machine or human generated.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Technical Library	1911	sanddocs@sandia.gov

This page left blank

Sandia
National
Laboratories

Sandia National Laboratories
is a multimission laboratory
managed and operated by
National Technology &
Engineering Solutions of
Sandia LLC, a wholly owned
subsidiary of Honeywell
International Inc. for the U.S.
Department of Energy’s
National Nuclear Security
Administration under contract
DE-NA0003525.