Jay Mhaiskar

Eunice Chin, Derek Howell

SCI 400

5 December 2025

Meaning and Word Overlap in AI Summaries: Measuring Similarity and Style

Literature Review

Text summarization has evolved from early rule-based methods to modern systems driven by large language models (LLMs). Early approaches relied on *lexical chains*, which are groups of words that are related in meaning and help identify the main topics in a text (Barzilay & Elhadad, 1997). Later improvements increased computational efficiency so these methods could handle larger documents (Silber & McCoy, 2002). Another line of work introduced *graph-based summarization*, where sentences are treated as nodes in a graph and connected by measures of similarity such as cosine similarity. Sentences that are highly connected in the graph receive higher importance scores through *eigenvector centrality*, a mathematical method that identifies influential nodes (Erkan & Radev, 2004). These techniques demonstrated that both wording and meaning can be analyzed in structured, quantitative ways.

With the rise of LLMs, summarization shifted from simply extracting sentences to *abstractive* methods, where models generate new phrasing. Much of the field continues to rely on ROUGE, a metric that measures textual overlap using n-grams (short word sequences), word pairs, and sentence fragments (Mridha et al., 2021; Nazari et al., 2018). Although widely used, ROUGE is limited because it measures exact matches rather than meaning. For instance, the sentences “the screen is really clear” and “the phone display is extremely clear” communicate the same idea but share few identical words, resulting in a low ROUGE score (Ganesan 2018 pg. 3).

Because of this, researchers increasingly use *embedding-based metrics*. Text embeddings are numerical representations of meaning: each sentence is converted into a vector (a list of numbers) that captures semantic relationships in a high-dimensional space. *Cosine similarity* measures how close these vectors point to each other, providing an estimate of how similar the texts are in meaning (Maples). However, cosine similarity also has limitations: it ignores sentence structure, can be affected by the large number of dimensions in the embedding space, and sometimes overestimates similarity when two texts share vocabulary but differ in meaning (Erkan & Radev, 2004). Despite these limitations, it remains appropriate for this study because it evaluates semantic similarity more effectively than surface-level word overlap.

Repetition is another important factor in AI-generated text. Historical systems such as *Strachey's Love Letter* program produced repetitive expressions, and similar patterns appear in modern interactive systems like AI Dungeon (Zhu, 2021). Repetition can therefore act as a *fingerprint*, meaning a stable and recognizable pattern in a model's output. This idea is supported by theoretical work such as Average Repetition Probability (ARP), which mathematically models how likely a system is to repeat itself across outputs (Fu et al., 2021). Although repetition is a known phenomenon, it has not been deeply integrated into summarization evaluation, leaving a gap in understanding how such patterns may distinguish one LLM from another. This connection between repetition and fingerprinting raises a larger issue: whether different LLMs produce meaningfully distinct summaries or tend to converge on similar outputs.

Currently, ChatGPT, Gemini, and Grok are the most visible LLMs as they are all backed by large corporations. Given the widespread use of these three LLMs for summarization tasks, it is important to determine whether their outputs meaningfully differ or whether they converge due to similar training data and design. Existing research highlights concerns about homogenization,

where different models produce similar outputs, potentially reducing diversity and amplifying shared biases (Liu et al., 2023). Work on detecting AI-generated text has explored watermarking, provenance tracking, and classifiers (Srinivasan, 2024), though these techniques face challenges such as false positives and vulnerability to adversarial manipulation (Salter et al., 2024). Other studies show that targeted edits to model parameters can embed persistent fingerprints (Wang et al., 2025), but such interventions require internal access and cannot be applied externally. This study instead focuses on *implicit* fingerprints, natural patterns produced by each model's inherent training and architecture. To interpret these potential similarities and differences, it is necessary to understand the stages of training that shape an LLM's behavior.

Large language models are trained in two major stages. In the first stage, pre-training, the model learns basic language skills from massive text datasets collected from the internet. This step does not teach stylistic preferences. Instead, it builds grammar, vocabulary and broad world knowledge (Brown et al., 2020; Raffel et al., 2020; Wei et al., 2022). In the second stage, post training or fine tuning, the model is trained again on curated datasets that include human feedback, instruction-following examples and company-specific preference data. This step shapes tone, safety behavior and style (Ouyang et al., 2022; Bai et al., 2022).

These considerations lead to the central research question: How much do LLM generated summaries overlap in meaning and wording, and does each model show a consistent style when summarizing the same article many times? To address this, the study uses Jaccard similarity to measure lexical overlap and cosine similarity to measure semantic similarity across embedding vectors. Test 1 examines cross-model similarity, while Test 2 tests within-model consistency by generating repeated summaries.

Understanding similarity and style supports both scientific goals, such as analyzing emergent structure in generative models, and practical goals, such as improving the evaluation, transparency and diversity of LLM outputs.

Methodology

This study examined how three large language models, ChatGPT 5, Gemini 2.5 Flash and Grok 4, summarize news articles and whether their outputs converge or show model specific patterns. To make comparisons meaningful, the models were evaluated under controlled conditions. Each model was accessed through a premium API to avoid differences in capability across free and paid tiers. All were given the same summarization task using an identical prompt: "Summarize this article in 150 words." Using a fixed word limit kept the outputs comparable and prevented similarity scores from being affected by differences in length.

The dataset consisted of 100 CNN news articles. Summaries generated by the models were lightly cleaned to remove filler expressions that do not contribute meaningfully to the analysis. Two similarity measures were used. Jaccard similarity captured lexical overlap by comparing the sets of words used by each model. Cosine similarity measured semantic closeness by comparing vector embeddings of the summaries. Using both metrics offered a combined view of how similar the summaries were in both word choice and meaning. The pseudocode for both similarity calculations is shown in Table 1, which outlines the exact computational steps used in the analysis.

The study proceeded in two stages. In Test 1, each of the 100 articles was summarized once by each model. The resulting summaries were compared across models to assess the degree of convergence in wording and meaning. In Test 2, a single article was summarized ten times by

each model to evaluate internal consistency. These repeated summaries were compared back to the article to identify stable stylistic patterns that may function as model specific fingerprints. All repeated outputs were stored in a dedicated dataset so they could be analyzed consistently. This design allowed the study to measure both cross model similarity and within model consistency while keeping external variables such as prompt wording, output length, personalization history and API version under strict control.

Table 1: Pseudo Code

Lexical Similarity:	Semantic Similarity:

def jaccard_similarity(text1, text2):
set1, set2 = text_to_set(text1), text_to_set(text2)
if not set1 or not set2:
return 0.0
return len(set1 & set2) / len(set1 | set2)
	def cosine_similarity(text1, text2):
	counter1, counter2 = text_to_counter(text1), text_to_counter(text2)
	common = set(counter1) & set(counter2)
	dot_product = sum(counter1[w] * counter2[w] for w in common)
	mag1 = math.sqrt(sum(v**2 for v in counter1.values()))
	mag2 = math.sqrt(sum(v**2 for v in counter2.values()))
	if mag1 == 0 or mag2 == 0:
	return 0.0
	return dot_product / (mag1 * mag2)

Hypotheses and Predictions

This study examined two core hypotheses related to cross model similarity and within model consistency.

Hypothesis 1: Summaries of the same article across different models would show significant overlap in content and wording.

Prediction: If this hypothesis were supported, Jaccard and cosine similarity scores would fall within a similar range for ChatGPT, Gemini and Grok when summarizing the same 100 articles. The models would show converging patterns in how they represented meaning and selected key information.

Hypothesis 2: Repeated summaries of the same article by the same model would show consistent model specific stylistic fingerprints.

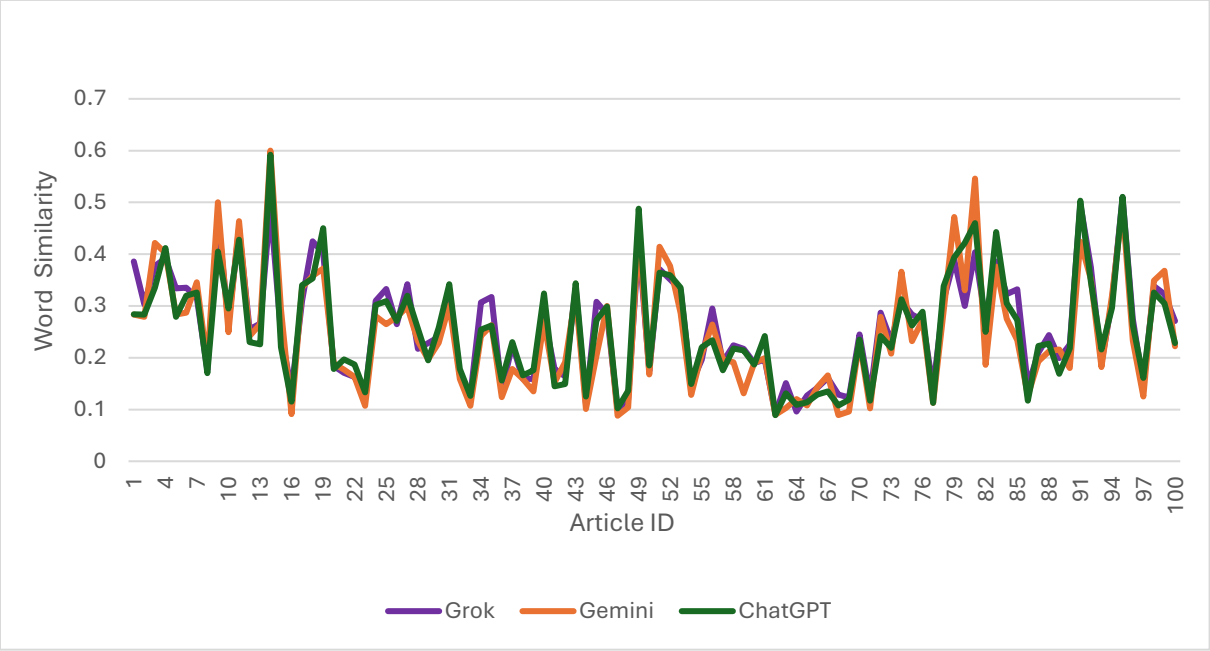
Prediction: If this hypothesis were supported, each model's ten repeated summaries would be more similar to one another than to summaries produced by other systems. The repeated outputs would form tight internal clusters, indicating stable stylistic patterns shaped during fine tuning.

Results

Test 1: Cross-Model Similarity

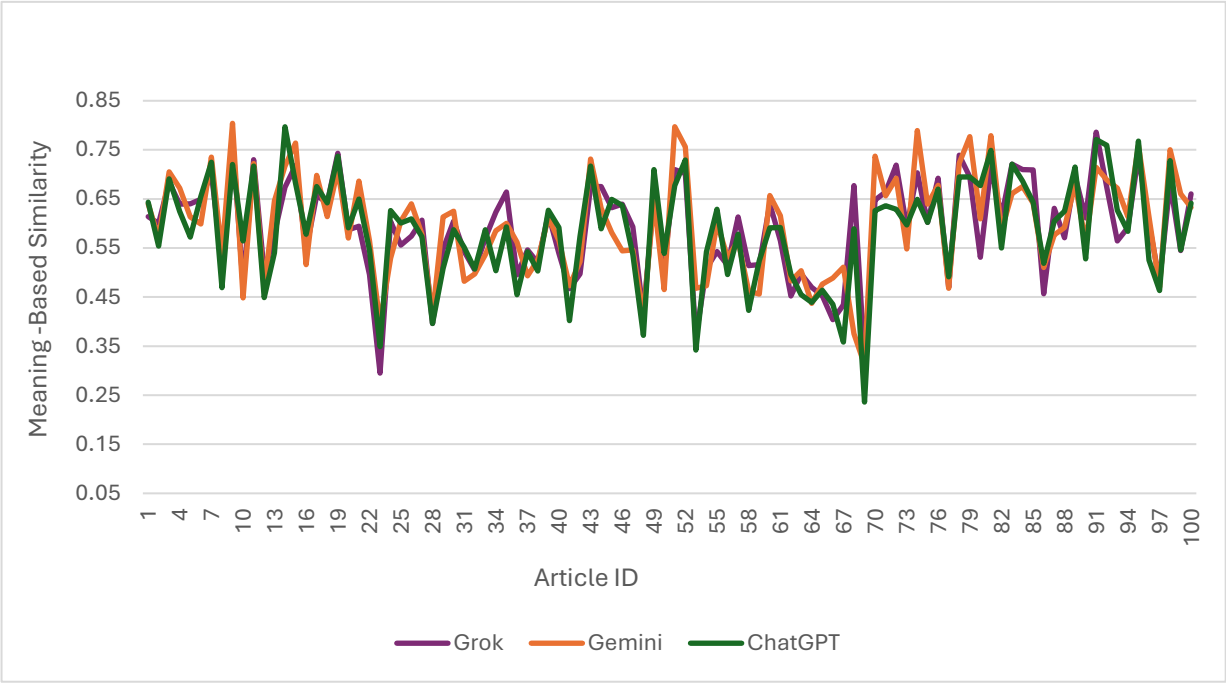
Similarity analyses were conducted on summaries generated by Grok-4, Gemini 2.5 Flash and ChatGPT-5 for 100 CNN articles. Jaccard similarity quantified lexical overlap, and cosine similarity quantified semantic similarity using vector embeddings.

Graph 1



Graph 1 shows that Grok, Gemini and ChatGPT follow almost identical patterns in lexical overlap across all 100 articles. Their mean Jaccard values were 0.259 for Grok, 0.247 for Gemini and 0.255 for ChatGPT, indicating that while the models do not use the exact same words, they draw from very similar portions of the source text. The close tracking of the Jaccard curves across articles shows that the three systems consistently choose comparable vocabulary to represent key information, even when the specific terms differ. This pattern suggests a shared approach to identifying what is lexically important in a news article.

Graph 2



Graph 2 reveals even stronger convergence across the models. Grok scored 0.590, Gemini scored 0.595 and ChatGPT scored 0.586 in mean cosine similarity, showing that the systems encode meaning in nearly identical ways. The curves rise and fall at the same article IDs, which indicates that all three models detect and prioritize the same underlying ideas, events and thematic structure in the source text. Even when they choose different words, their internal semantic representations align closely. This tight synchronization reinforces the conclusion that these LLMs interpret meaning in a highly similar manner.

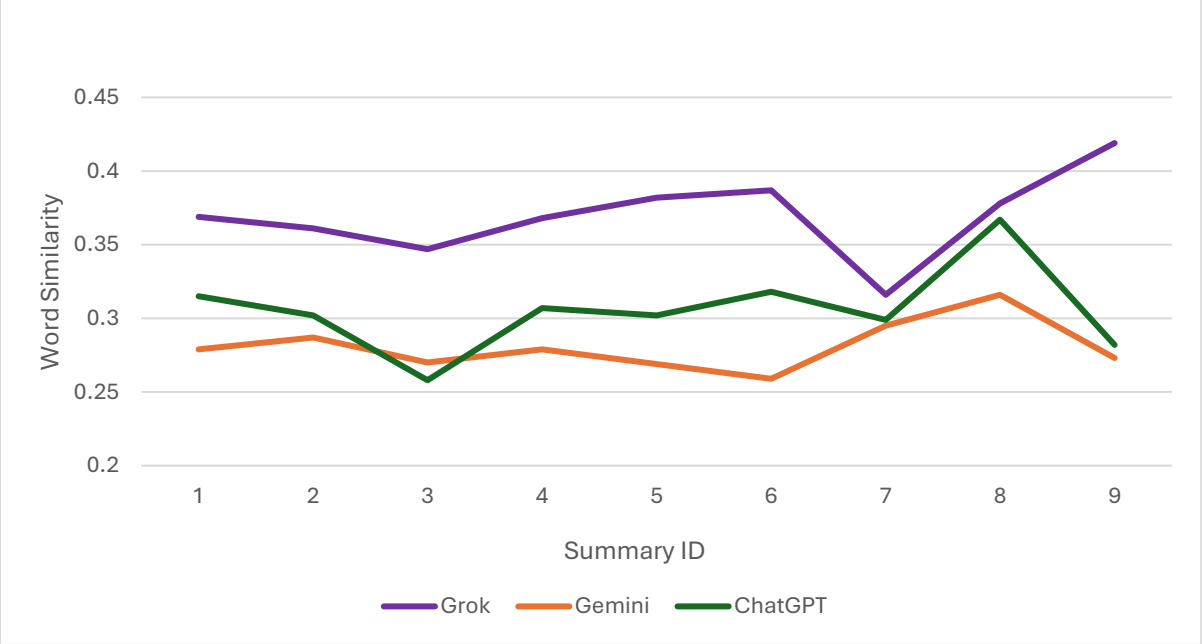
Articles with high semantic similarity scores were those with a single clear main point and clearly presented facts. These articles made the key information very clear, which caused the models to agree strongly. In contrast, the articles with low semantic similarity contained multiple potential main points, such as geopolitical stories, complex political interactions or emotional narratives with quotes and personal testimonies. In these cases, the models could not easily

identify one dominant theme. Each system selected a different angle or thread from the article, which resulted in lower semantic similarity scores.

Test 2: Within-Model Consistency

Test 2 assessed intra-model consistency by generating 10 summaries per model for a single CNN article and comparing each summary back to the article. All models exhibited markedly higher within-model similarity than was observed across models.

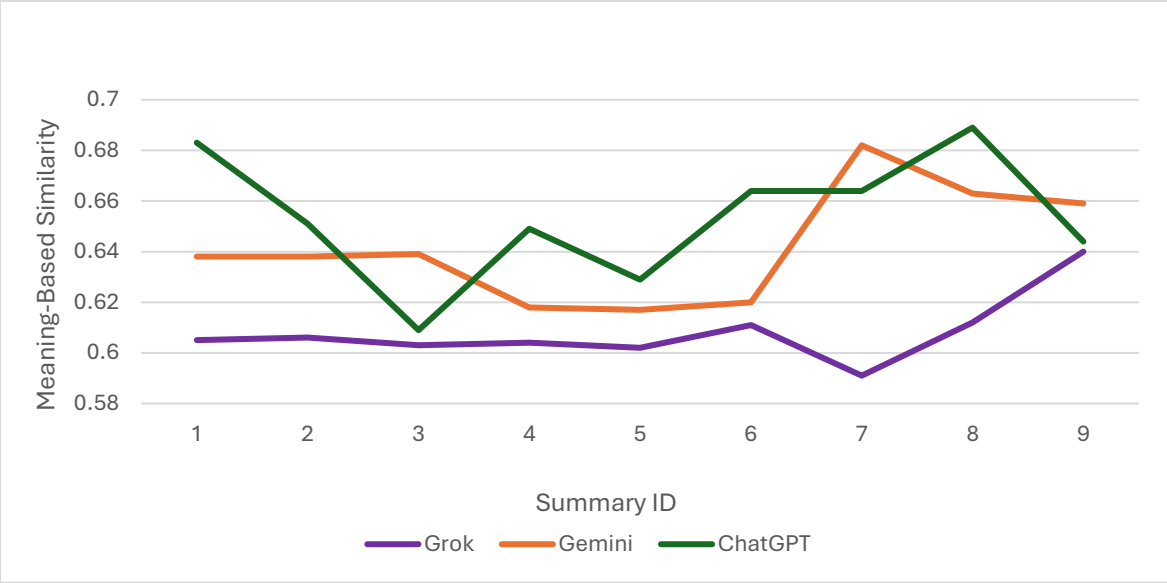
Graph 3



Graph 3 shows that each system becomes more lexically consistent when summarizing the same article multiple times. Grok produced a mean Jaccard value of 0.370, indicating the strongest and most stable word choice patterns among the three models. Gemini showed a lower but still consistent mean Jaccard value of 0.281, while ChatGPT produced a mean of 0.306. All three values are higher than their cross model Jaccard scores, which shows that each LLM tends

to reuse similar vocabulary across repeated summaries. These results suggest that each system has a characteristic lexical style that remains stable across iterations.

Graph 4



Graph 4 highlights strong internal semantic consistency across all three models. Grok produced a mean cosine value of 0.608, Gemini showed a higher mean of 0.642 and ChatGPT demonstrated the strongest semantic stability with a mean of 0.654. These values are all higher than the models' cross model cosine scores, indicating that each system represents meaning more consistently within itself than across systems. Once a model identifies what it considers the essential meaning of an article, it reproduces that interpretation with only minor variation. This supports the presence of model specific semantic fingerprints. The patterns observed in Test 2 show that each model is internally consistent but differs from the others, offering quantitative evidence that companies use different post training that led to distinct stylistic fingerprints.

Discussion

The convergence observed in both lexical and semantic similarity in Test 1 points to meaningful overlap in how the three systems were pre-trained. All models produced nearly identical mean cosine scores (Grok = 0.590, Gemini = 0.595, ChatGPT = 0.586), and their semantic curves rose and fell in the same places across the 100 articles. Likewise, their mean Jaccard values were closely aligned (Grok = 0.259, Gemini = 0.247, ChatGPT = 0.255), with all three models showing the same pattern of lexical overlap across articles. These results show that the models not only extracted meaning in almost the same way but also selected vocabulary from similar portions of the source text, even when their phrasing differed. Since pre-training is the stage where an LLM learns general language structure, grammar and broad world knowledge, this level of alignment suggests that Grok, Gemini and ChatGPT were trained on large corpora that are similar in scale, composition and distribution. If their pre-training differed substantially, greater divergence would be expected in both meaning and word-level patterns. Instead, the consistency across both similarity metrics indicates shared foundational language representations shaped during the pre-training stage.

The results from Test 2 show clear evidence that the models' post-training, or fine-tuning, is where their distinct behaviors emerge. When each system summarized the same article ten times, all three displayed stronger internal consistency than in the cross-model analysis of Test 1. Grok produced a mean Jaccard score of 0.370 and a cosine score of 0.608, indicating stable lexical and semantic patterns across iterations. Gemini showed a mean Jaccard value of 0.281 and a cosine value of 0.642, while ChatGPT demonstrated the strongest semantic stability, with a Jaccard mean of 0.306 and a cosine mean of 0.654. These within-model scores were all higher than their Test 1 values, showing that each model aligns more closely with its own prior output

than with summaries generated by other systems. This pattern reflects the influence of fine-tuning, which is the stage where a model learns stylistic behaviors, preference patterns and company-specific norms. The fact that the three models diverged from one another in their repeated lexical and semantic patterns suggests that their fine-tuning datasets and objectives differ in meaningful ways. In contrast to the shared pre-training signals seen in Test 1, the increased consistency within each system in Test 2 quantifies the stylistic fingerprints that arise from fine-tuning and distinguishes one model's behavior from another.

Limitations

This study has several limitations. First, the dataset included only CNN articles. This ensured consistency in style and structure, but it also limited the variability of inputs the models encountered. A broader dataset spanning multiple news organizations was not used because of time and resource constraints, but future work should incorporate articles from outlets such as BBC, AP, Fox and Al Jazeera to test whether the patterns observed here generalize across different genres and political framings. Second, the summarization prompt was standardized to maintain experimental control. Although this reduced variability was introduced by instruction design, it also meant the study could not examine how different prompts might alter lexical or semantic overlap. Future research should test a range of prompts, including those that encourage close paraphrasing or stylistic imitation. Finally, in Test 2, each model generated only ten repeated summaries. This choice balanced computational cost with the need for repeated measures, but a larger sample such as one hundred iterations would provide more robust estimates of internal consistency and might reveal additional fine-tuning patterns. Expanding these aspects in future studies would strengthen the generalizability and depth of the findings.

Conclusion

This study showed that large language models converge strongly when summarizing news content. Across 100 CNN articles, the three systems produced nearly identical levels of semantic similarity, with mean cosine scores of 0.590 for Grok, 0.595 for Gemini and 0.586 for ChatGPT. Their lexical overlap, while lower in magnitude, was also tightly aligned, with mean Jaccard values of 0.259, 0.247 and 0.255. These values quantify the degree of cross model convergence and indicate that all three models relied on similar sections of the source text and extracted meaning in almost the same way. In contrast, Test 2 showed that each model demonstrated stronger similarity to its own repeated summaries than to those of other systems, with within model cosine scores rising to 0.608 for Grok, 0.642 for Gemini and 0.654 for ChatGPT. This increase in similarity quantifies the presence of model-specific fingerprints that appear to originate from differences in fine-tuning. Together, these findings show that while pre-training produces a shared foundation that drives cross model convergence, fine-tuning creates distinctive internal styles. Quantifying these effects provides a clearer picture of where LLMs behave similarly and where they diverge, offering a basis for future work on model evaluation and AI-generated text detection

Works Cited

Auriemma Citarella, Alessia, et al. “Assessing the Effectiveness of ROUGE as Unbiased Metric in Extractive vs. Abstractive Summarization Techniques.” *Journal of Computational Science*, vol. 87, May 2025, p. 102571. *DOI.org (Crossref)*, https://doi.org/10.1016/j.jocs.2025.102571.

Bai, Yuntao, et al. “Constitutional AI: Harmlessness from AI Feedback.” arXiv:2212.08073, arXiv, 15 Dec. 2022. *arXiv.org*, https://doi.org/10.48550/arXiv.2212.08073.

Barzilay, Regina, and Michael Elhadad. “Using Lexical Chains for Text Summarization.” *Intelligent Scalable Text Summarization*, 1997. *ACLWeb*, https://aclanthology.org/W97-0703/.

Brown, Tom B., et al. “Language Models Are Few-Shot Learners.” arXiv:2005.14165, arXiv, 22 July 2020. *arXiv.org*, https://doi.org/10.48550/arXiv.2005.14165.

Cheng, Ruijia, et al. “Mapping the Design Space of Human-AI Interaction in Text Summarization.” arXiv:2206.14863, arXiv, 29 June 2022. *arXiv.org*, https://doi.org/10.48550/arXiv.2206.14863.

Erkan, Gunes, and Dragomir R. Radev. “LexRank: Graph-Based Lexical Centrality as Salience in Text Summarization.” *Journal of Artificial Intelligence Research*, vol. 22, Dec. 2004, pp. 457–79. *arXiv.org*, https://doi.org/10.1613/jair.1523.

Fu, Zihao, et al. “A Theoretical Analysis of the Repetition Problem in Text Generation.” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, May 2021, pp. 12848–56. *DOI.org (Crossref)*, https://doi.org/10.1609/aaai.v35i14.17520.

Ganesan, Kavita. “ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks.” arXiv:1803.01937, arXiv, 5 Mar. 2018. *arXiv.org*, https://doi.org/10.48550/arXiv.1803.01937.

Hristozova, Nina. “To ROUGE or Not to ROUGE?” *Towards Data Science*, 16 Apr. 2021, https://towardsdatascience.com/to-rouge-or-not-to-rouge-6a5f3552ea45/.

Liu, Yu Lu, et al. “Responsible AI Considerations in Text Summarization Research: A Review of Current Practices.” arXiv:2311.11103, arXiv, 18 Nov. 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2311.11103.

Maples, Sydney. *The ROUGE-AR: A Proposed Extension to the ROUGE Evaluation Metric for Abstractive Text Summarization*.

Mridha, M. F., et al. “A Survey of Automatic Text Summarization: Progress, Process and Challenges.” *IEEE Access*, vol. 9, 2021, pp. 156043–70. *DOI.org (Crossref)*, https://doi.org/10.1109/ACCESS.2021.3129786.

Nazari, nasrin, and M. Amin Mahdavi. “A Survey on Automatic Text Summarization.” *Journal of AI and Data Mining*, no. Online First, May 2018. *DOI.org (CSL JSON)*, https://doi.org/10.22044/jadm.2018.6139.1726.

Ouyang, Long, et al. “Training Language Models to Follow Instructions with Human Feedback.” arXiv:2203.02155, arXiv, 4 Mar. 2022. *arXiv.org*, https://doi.org/10.48550/arXiv.2203.02155.

Raffel, Colin, et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. *Zotero*.

Salter, Steven, et al. “Human-Created and AI-Generated Text: What’s Left to Uncover?” *Intelligent Computing*, edited by Kohei Arai, vol. 1017, Springer Nature Switzerland,

2024, pp. 74–80. Lecture Notes in Networks and Systems. *DOI.org (Crossref)*, https://doi.org/10.1007/978-3-031-62277-9_5.

Silber, H. Gregory, and Kathleen F. McCoy. “Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization.” *Computational Linguistics* [Cambridge, MA], vol. 28, no. 4, 2002, pp. 487–96. *ACLWeb*, https://doi.org/10.1162/089120102762671954.

Srinivasan, Siddarth. “Detecting AI Fingerprints: A Guide to Watermarking and Beyond.” *Brookings*, 4 Jan. 2024, https://www.brookings.edu/articles/detecting-ai-fingerprints-a-guide-to-watermarking-and-beyond/.

Tianhua, Zhu. “From Textual Experiments to Experimental Texts: Expressive Repetition in ‘Artificial Intelligence Literature.’” *Theoretical Studies in Literature and Art*, 2021.

Wang, Shida, et al. “FPEdit: Robust LLM Fingerprinting through Localized Knowledge Editing.” arXiv:2508.02092, arXiv, 4 Aug. 2025. *arXiv.org*, https://doi.org/10.48550/arXiv.2508.02092.

Wei, Jason, et al. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.” arXiv:2201.11903, arXiv, 10 Jan. 2023. *arXiv.org*, https://doi.org/10.48550/arXiv.2201.11903.

Zhang, Yang, et al. “A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods.” Version 2, arXiv:2403.02901, arXiv, 20 Mar. 2025. *arXiv.org*, https://doi.org/10.48550/arXiv.2403.02901.