Evaluating Output Novelty in Iterative Prompting in Educational Content Generation

Vickey Ghimire, Jaljala Shrestha Lama, Bijay Dhungana, Nazmus Sadat, Nicholas Caporusso
School of Computing and Analytics
Northern Kentucky University
Highland Heights, KY, United States
{sadatm1, caporusson1}@nku.edu

***Abstract*—Large Language Models (LLMs) have been explored as a valuable tool for creating course outlines. To this end, iterative prompting can be utilized to correct and expand the output. However, iterative prompting often results in LLMs producing redundant content that requires extensive manual review. In this context, distinguishing between genuinely novel content and rephrased existing information creates a trade-off between LLM capabilities and validation efforts. To address this, this paper proposes a novelty metric that combines lexical similarity measures with semantic analysis to automatically classify generated content as novel or repeated, thus simplifying human review. We evaluated this approach using five widely used LLMs (ChatGPT 4o, Claude 3.7 Sonnet, Gemini 2.5 Flash, DeepSeek v3, and DeepSeek r1) to create a Java programming course outline over multiple iterations. Our methodology automated the classification of 90.52% of the generated content, significantly reducing the number of items requiring manual review. Moreover, our findings show that the proposed metric can also be utilized for ranking the performance of models in the context of iterative prompting.**

***Index Terms*—Large Language Models, Iterative Prompting, Content Novelty Detection, AI in Education**

I. Introduction

Recent advances in Large Language Models (LLMs) have transformed text-based generative Artificial Intelligence (AI) into an everyday tool that can be seamlessly integrated into workflows, including educational authoring.

A growing scholarly literature focuses on how LLMs can support learning and teaching, particularly course outline and content creation [1]–[3]. Many studies, including [4], showed that LLMs are particularly helpful in instructional design at all levels, particularly in structuring academic content such as course outlines, syllabi, assignments, and even personalized learning paths. Thanks to LLMs' ability to comprehend context and replicate human-like reasoning, instructors are increasingly using them to update syllabi and course outlines and even design them from scratch. For instance, LLMs can analyze existing curricular content and propose optimized course outlines and structures, helping educators focus more on supporting students and pursuing learning outcomes rather than managing course logistics. LLM-aided content generation is particularly valuable in rapidly evolving fields such as software engineering, data science, and cybersecurity, where materials can become outdated even between academic terms.

Especially in the context of new course design, instructors often need to refine the initial, high-level output of the LLM through iterative prompting to address the inherent complexity of course outlines and obtain a more in-depth structure or to explore additional content to be included. To this end, they can ask the LLM to revise the structure to incorporate emerging topics, refine the learning outcomes, expand or reduce the scope of specific areas, or make the course more practical. However, the open question is how LLMs respond through iterations, and particularly whether they enrich the outline with new content or merely rephrase items from the previous answer. Indeed, instructors can manually review each subsequent version produced by the LLM to identify and integrate the novel content. However, this affects the trade-off between the advantage offered by the LLM's editorial work and the effort required to analyze the output's novelty and relevance, particularly in the context of multiple iterations or even conversations. This is especially true regardless of whether the initial prompt already contains a draft outline. Indeed, decisions about topic relevance might be primarily influenced by specific course objectives and instructors' perspectives, and thus require additional context and review. On the contrary, content novelty in the context of iterative prompting could be automatically measured to facilitate manual revisions.

This paper presents a novel approach to automatically measure the novelty of LLMs' output in the context of iterative prompting, particularly for generating course outlines. Our method enables the automatic evaluation and comparison of different models' behavior and performance in response to iterative prompting across various applications and fields, thereby significantly simplifying manual review. Furthermore, the proposed content novelty measure could be utilized to evaluate different prompting strategies, model families, or temperature settings, which, in turn, could ultimately feed back into the LLM architecture to improve reasoning techniques.

II. Related Work

LLM-aided educational content generation, including curriculum, lesson plans, and assessment items, has been explored by several studies. Sridhar et al. [1] demonstrated that GPT-4 can draft learning objectives according to Bloom's taxonomy, enabling instructors to create syllabi in 50% less time. However, the study also noted variability in objective specificity between disciplines. Sajja et al. [5] developed an AI-augmented intelligent educational framework that automatically generates course-specific intelligent assistants for different disciplines and academic levels. Fan et al. [6] developed an interactive LLM-based system to assist new teachers in preparing lesson plans, including a structured outline, materials, and examples. Yan et al. [4] surveyed 118 studies and showed that GPT models effectively produce course outlines and quiz questions. However, the results often require manual editing to ensure domain precision and pedagogical alignment.

Dornburg and Davin [7] explored ChatGPT's capability in creating foreign language lesson plans, examining how the detail level of user prompts affects the quality and consistency of the content. The study found that adding more context may not always guarantee improved results, and identical prompts can yield significantly different results. Bao et al. [8] introduced a framework for iteratively generating and evaluating explanations for multiple-choice question quizzes. They demonstrated that subsequent prompting yielded more effective explanations for questions. Zheng et al. [3] proposed a framework for generating customized lesson plans, where the LLM conducts a self-critique based on educator-defined criteria and refines the plan accordingly. Experienced educators indicated that the proposed method can produce better lesson plans than other LLM-based methods and human-designed plans. In [9], [10], the authors tested different models in educational applications, including official GMAT questions and automated essay scoring for English learners, and reported persistent reliability issues across the models. Other studies [11], [12] proposed frameworks to determine the originality of LLM-generated content with respect to the training data. They show that while methods like temperature scaling or denial prompting raise originality, they usually hurt quality, requiring a trade-off between the two [11]. Also, at the level of broader sentence structure, sometimes LLM-generated text is as novel or even more novel than human text. The authors of [13] proposed an approach to improve LLM responses by iterative prompting. They showed that iterative prompting alone, without fine-tuning or reinforcement learning, can improve LLM performance in tasks such as math reasoning and code optimization.

Although these works significantly contribute to the field, no prior work has systematically explored the uniqueness of content generation by LLMs in the context of creating course outlines. This paper addresses this gap by proposing an algorithm to assess the novelty of the LLMs' output and the consistency across the responses in iterative prompting, particularly in the context of educational content.

III. Assessing Novelty in Iterative Prompting

Although iterative prompting is widely used, current benchmarks usually focus on aspects such as factual correctness or writing style, and there is a lack of ways to measure whether LLMs genuinely introduce new concepts or merely rearrange existing ideas, an important distinction particularly for knowledge-driven contexts, including education.

Therefore, we propose a novelty metric that evaluates whether LLMs truly add new content when prompted repeatedly, or if they mostly repeat or rephrase information from earlier outputs, answering the question "*How can we measure how much new information an LLM produces when prompted to refine and expand its response*"? When an instructor asks an LLM to define the outline of a course, after the first output (i.e., a list of topics), the response to subsequent prompts asking for more detailed information will consist of a new list of topics that can be divided into the following groups:

- *Exactly the same* content. This was already present in a previous version and, therefore, can be disregarded.
- *Completely different* content. Novel contributions from the latest response that expand and help explore a topic horizontally or vertically, or introduce new topics.
- *Similar* content, including repetitions, paraphrased versions of previously introduced items, but incorporating some new information not presented in the previous iteration(s). This type of content could be considered mostly similar to or mostly different from previously seen output. However, part of this material, typically at the center of the similarity spectrum, will involve uncertain or ambiguous content, vague, unclear, or potentially overlapping entries requiring further analysis, clarification, or disambiguation that even human supervision might not be able to classify into the previous categories.

With respect to the latter content type, our work aims to reduce the need for human effort in evaluating which similar topics from the new response are mostly the same, mostly new, or too ambiguous to be categorized. This involves applying a fuzzier definition of content similarity and thresholds that enable labeling ambiguous content as either mostly similar content or mostly different (as shown in Figure 1) when compiling a new version of the outline. To this end, we use automated methods to maximize the content correctly classified in the first two categories. Surface-level comparisons and traditional lexical similarity measures such as cosine similarity or Jaccard indices work well at the endpoints of the spectrum, where content is completely different or exactly the same. Unfortunately, toward the center of the spectrum, they progressively fail to capture the nuanced semantic relationships inherent in natural language. On the other hand, more sophisticated tools, including LLMs, are unsuitable for topic-level categorization tasks due to their complexity and cost. Furthermore, they might still perform randomly at the center of the spectrum.

Therefore, our "novelty" score (NOV) scaffolds several pairwise similarity calculation methods into a unique algorithm to achieve a trade-off between accurate classification and feasible automation. Also, the algorithms utilized for comparison can vary based on the specific case. For instance, in this paper, we use a hybrid filter based on techniques that require minimal computational resources (i.e., Jaccard) to check for complete similarity between the new content (i.e., list of topics in the latest outline) and the existing content (i.e., list of topics in the initial or previous response), thus tackling the ends of the spectrum. Then, we apply more advanced techniques involving semantic similarity (e.g., transformer-based) to categorize the remaining as mostly similar or mostly different while

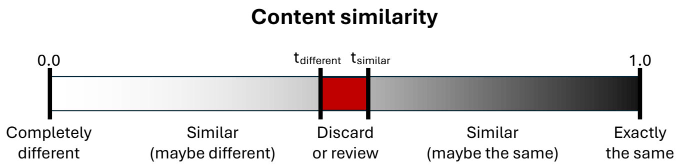

Fig. 1. Content similarity spectrum highlighting the area usually requiring the most human supervision to disambiguate similarity.

maintaining low computational costs. Finally, we define two thresholds (i.e., t_{diff} and t_{same}) where new content that does not fit the two categories can be discarded or needs review. As a result, NOV involves solving an optimization problem that maximizes F1 while reducing the distance between t_{diff} and t_{same} to avoid flagging too much content as to be discarded or reviewed. By doing this, NOV automatically flags the new content as novel, existing, or to be discarded or reviewed. This is particularly effective for creating course outlines, syllabi, and, generally, organizing content into a structure. In fact, NOV can be used in the context of iterative prompting to:

1) Automatically assign a score to the output of every iteration and use it to compile an aggregated outline, thus minimizing the effort required for human review.
2) Integrate the outputs of multiple conversations across different LLMs to draw from different sources.
3) Evaluate the performance of LLMs and benchmark their strategies and effectiveness in iterative prompting, aiding users in choosing a model.
4) Progressively and autonomously expand an outline vertically (i.e., specialize the topics) and horizontally (i.e., find additional topics) through automatic prompt iteration.
5) Enhance LLMs with an additional Chain of Thought (CoT) layer designed to maximize the performance of iterative prompting with minimal computational overhead.

Finally, in line with the ultimate goal of our work, NOV can be utilized to aid and evaluate LLMs in producing outlines from structured and unstructured content, particularly in the context of Retrieval Augmented Generation (RAG).

IV. Study

The goal of this study was to apply the proposed methodology to a real-world scenario to demonstrate its feasibility, assess its performance, illustrate a use case for educational content creation, and discuss its benefits and implications based on empirical evidence.

A. Materials and Methods

In this study, we considered five LLMs: ChatGPT 4o, ClaudeAI 3.7 Sonnet, Gemini 2.5 Flash, and two versions of DeepSeek (r1 and v3). We used these models because they are among the most popular and widely used LLMs. Each model offers a unique perspective and reasoning style, including models with CoT reasoning. In addition to comparing their performances in response to iterative prompting, this enabled us to evaluate the applicability of our approach to different scenarios. Although we focused on these five models only, our methodology can be used with any LLM.

For our study, we focused on technical knowledge, specifically Java and Object-Oriented Programming, because it is one of the most widely taught and used programming languages, and we asked the LLMs to produce an outline for a 16-week course. Java is an established language, commonly included in university curricula, and still relevant in industry-level applications. As a result, it is particularly suitable to be utilized as a standardized benchmark for evaluating how different LLMs generate structured educational content.

B. Protocol

Our data collection involved five people, each generating output from each LLM considered. The LLMs were initially given a prompt asking them to create the course outline. The prompt contained detailed instructions on the context of the course, target audience, and output format. Concerning the latter component of the prompt, the LLM was instructed to organize the content into two levels of headings and a list of at least three topics for each subheading. Then, a different prompt asked the LLM to improve upon the previous response by revising the outline to expand it and add relevant topics while maintaining the same output format. In each conversation, the LLM was asked to further enhance its response a total of four times. The same prompt was utilized in each iteration.

At the end of the data collection process, we obtained 25 conversations, each consisting of five iterations. Although this number might seem small, each conversation involved a total of at least 200 topics (42 topics per iteration, on average), which resulted in a cumulative total of more than 5,200 individual course topics to be analyzed. The LLM output from every iteration was stored in markdown format in separate files, which were then parsed and pre-processed to remove non-relevant components of the response (e.g., introductory paragraph and conclusion) and keep only the course outline.

Each conversation was analyzed separately, and the output of each iteration was compared with the previous one. Pairwise similarity was calculated at the topic level with each of the string comparison techniques discussed earlier, resulting in a similarity score for each topic, which then enabled calculating novelty metrics representing each iteration and the entire conversation. Finally, conversation-level metrics were aggregated by model to compare the strategies of each LLM.

V. Results and Discussion

After processing the markdown documents containing the individual iterations for each conversation, we calculated simple statistics describing content that is exactly the same, completely different, and similar. As shown in Table I, out of 9054 topics produced across five iterations, 3072 (33.93%) involve different content, 3694 (40.80%) are exactly the same, and 2288 (25.27%) would require further examination by a human. The data show that, in general, as the user iterates over the same response, LLMs keep expanding the outline, producing between one-third and two-thirds of new content.

TABLE I
ITERATION-WISE SIMILARITY DISTRIBUTION

	Output 2	**Output 3**	**Output 4**	**Output 5**	**Total**
Different	667	708	822	875	3072
Same	796	911	949	1038	3694
Similar	474	513	631	670	2288
Total	1937	2132	2402	2583	9054

Subsequently, we aggregated the data by model. Figure 2 represents the similarity distribution of the content generated by each model across the five iterations. As shown in the figure, DeepSeek r1 obtained the highest percentage of different topics, minimizing repeat content. This model performed best, as it creates course outlines that explore new topics with minimal human intervention. Gemini Flash 2.0 was fairly balanced and slightly more stable than ChatGPT 4o. On the contrary, Claude 3.7 and DeepSeek v3 demonstrate high redundancy and poor ability to produce new content through iterative prompting. Our findings suggest that, based on similarity metrics alone, DeepSeek r1 outperforms the other models and should be preferred for creating course outlines. Nevertheless, the performances of Gemini Flash 2.0 and ChatGPT 4o render them valid alternatives. However, even considering the three best-performing models, similarity metrics show that across the five iterations, DeepSeek r1, Gemini Flash 2.0, and ChatGPT 4o respectively produce 47.06%, 35.71%, and 32.25% of content labeled as similar and requiring further human revision. For instance, in one iteration ChatGPT 4o produced *"Defining methods, parameters, return values."*, and in the following iteration it generated *"Defining methods, parameters, passing return types, varargs."*.

To automatically disambiguate content similarity, we computed the NOV score described in the previous section. Specifically, we set thresholds t_{diff} and t_{same} at 0.46 and 0.54, respectively, because they were validated to produce the most accurate results (as discussed below). Specifically, the NOV score was calculated using a second-stage filter aggregating semantic measures (i.e., cosine similarity and transformer-based models). Out of 2288 topics that were considered similar (i.e., 25.28% of the total topics produced by the LLMs), and, thus, required human intervention, the NOV score enabled the automatic classification of 2071 items (i.e., 90.52%) as either mostly different (65.2%) or mostly similar (25.31%). In addition to reducing human revision (only 9.48% require manual review) and improving the overall iterative prompting workflow, NOV scores provide a clearer benchmark of the models. As shown in Figure 3, while the ranking of the five LLMs remains unchanged, their NOV scores show that most of the content produced by DeepSeek r1 (i.e., 94.36%) across multiple iterations involves additional information, which distances this model from its competitors. In comparison, Gemini Flash 2.0, ChatGPT 4o, Claude 3.7, and DeepSeek v3 produced only 56.51%, 55.64%, 42.6%, and 14.78% of new content, respectively.

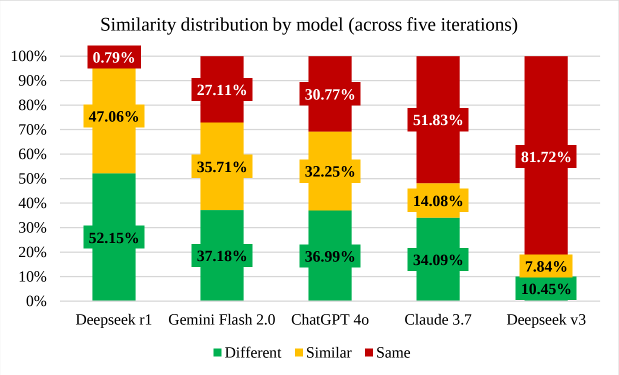

Fig. 2. Content similarity distribution by model.

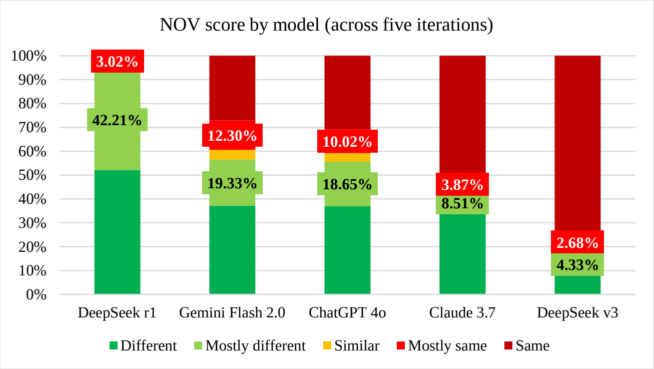

Fig. 3. NOV score by model (data labels highlight conversion of similar content into mostly different and mostly same content).

A. *Performance evaluation*

Furthermore, we evaluate the performance of our method in terms of computational cost and accuracy, to assess which algorithm results in the best trade-off and whether our proposed approach can be reliably applied to course outline generation.

The computational cost of similarity calculation using Jaccard, cosine similarity, and BERT was on average 0.014 ± 0.022 ms, 0.044 ± 0.041 ms, 0.59 ± 0.29 ms, respectively, showing that transformer-based models required one order of magnitude more time. On the contrary, the LLM took an average of 1058.23 ± 2863.77 ms to complete one calculation. These results are also reported in Table II. Although all performances largely depend on the runtime environment, the difference between the time required by the three algorithms and the LLM's performance makes the latter the least suitable for our methodology. Each topic required an average of 36.86 comparisons, and each iteration involved an average of 128.65 topic pairs comparisons, for a total of approximately $5,000$ calculations per iteration, on average, leading to an average additional computational cost of 70 ms, 220 ms, and $2,950$ ms using Jaccard, Cosine similarity, and BERT, respectively. While the delay introduced by the first two algorithms is negligible, the longer processing times required by transformer-based models can be noticed by the end-user and could affect their experience. However, the additional delay (i.e., ∼4 seconds on average) is consistent with the time required by LLMs that refine their output through Chain of Thought (CoT), including thinking models.

Finally, we evaluated the accuracy of our approach in calculating the NOV score. To this end, we randomly selected a sample conversation for each model, resulting in a total of 25 iterations across five models, involving 1090 topics (21% of the total). Then, we manually labeled each topic pair in each iteration as *mostly new* (i.e., 1.0) or *mostly repeated* (i.e., 0.0) content. Then, to evaluate the effectiveness of our approach, we calculated the accuracy metrics for the components of NOV (i.e., lexical, semantic, and LLM-based similarity analysis) and compared all the possible alternative options. Our findings, reported in Table II, show that no single-model configuration dominates quality metrics, leading to the need for trade-offs, which confirms the need for a scaffolded approach. Specifically, when utilized individually, Jaccard achieves high lexical precision and the best accuracy, but very low recall (i.e, 30%), leaving many items unlabeled or wrongly negative. As a result, it works best for comparing content at the very end of the similarity spectrum. Compared to cosine similarity, transformer-based models capture deeper semantics without increasing false positives, making them ideal as a second-stage filter. Our analysis also shows that using lexical and semantic pairing as a first-stage similarity filter helps recall without sacrificing much accuracy, minimizing the percentage of topics left uncategorized (i.e., 0.56%). Conversely, the LLM classifies almost everything as positive, resulting in the lowest precision and confirming that, even from an accuracy standpoint, it is unsuitable as a first pass. Although adding the LLM as the last-stage filter results in the strongest overall metrics with very moderate skip (i.e., 3.72%), this requires an additional trade-off regarding computational costs.

TABLE II
PERFORMANCE ANALYSIS OF INDIVIDUAL SIMILARITY METHODS
(JCB INDICATES THE INTEGRATION OF JACCARD, COSINE, AND BERT)

Metric	**Jaccard**	**Cosine**	**BERT**	**LLM**	**JCB**
Compute time (ms)	0.014	0.044	0.59	1,058.23	0.65
Accuracy	64.78%	61.83%	61.98%	48.44%	63.89%
Precision	59.49%	53.16%	53.04%	44.92%	56.84%
Recall	30.13%	60.87%	64.53%	99.33%	52.74%
F1	40.00%	56.76%	58.22%	61.86%	60.43%
Skipped topics	8.18%	6.51%	8.09%	1.77%	3.72%

VI. CONCLUSION AND FUTURE WORK

We developed a systematic approach to measure the novelty of LLM-generated content through iterative prompting, focusing on creating educational course outlines. By evaluating outputs from five LLMs, including models with the chain-of-thought feature, our framework effectively distinguished genuinely novel content from repeated or minimally changed information, and found that thinking models demonstrate superior performance. Furthermore, the proposed NOV score reduced human effort by enabling automatic classification of most generated content. Additional material is available at https://github.com/NKU-HCI-lab/project-LLM-NOV.git.

Our findings highlight the utility of this approach in educational settings and suggest broader applications in structured content generation. Future research will focus on comparing different algorithms for NOV calculation, automating iterative prompting, and extending our methodology to a broader range of subjects, including technical documentation, legal drafting, and training manuals. We also plan to broaden the scope of our automated classification by incorporating relevance checks for LLM-generated output, to ensure the content is not only novel but also relevant to the intended application and audience.

REFERENCES

[1] P. Sridhar, A. Doyle, A. Agarwal, C. Bogart, J. Savelka, and M. Sakr, “Harnessing LLMs in curricular design: Using GPT-4 to support authoring of learning objectives,” *arXiv preprint arXiv:2306.17459*, 2023.

[2] A. K. Law, J. So, C. T. Lui, Y. F. Choi, K. H. Cheung, K. Kei-ching Hung, and C. A. Graham, “AI versus human-generated multiple-choice questions for medical education: a cohort study in a high-stakes examination,” *BMC Medical Education*, vol. 25, no. 1, p. 208, 2025.

[3] Y. Zheng, X. Li, Y. Huang, Q. Liang, T. Guo, M. Hou, B. Gao, M. Tian, Z. Liu, and W. Luo, “Automatic Lesson Plan Generation via Large Language Models with Self-critique Prompting,” in *International Conference on Artificial Intelligence in Education*. Springer, 2024, pp. 163–178.

[4] L. Yan, L. Sha, L. Zhao, Y. Li, R. Martinez-Maldonado, G. Chen, X. Li, Y. Jin, and D. Gašević, “Practical and ethical challenges of large language models in education: A systematic scoping review,” *British Journal of Educational Technology*, vol. 55, no. 1, pp. 90–112, 2024.

[5] R. Sajja, Y. Sermet, D. Cwiertny, and I. Demir, “Platform-independent and curriculum-oriented intelligent assistant for higher education,” *International Journal of Educational Technology in Higher Education*, vol. 20, p. 42, 2023.

[6] H. Fan, G. Chen, X. Wang, and Z. Peng, “LessonPlanner: Assisting novice teachers to prepare pedagogy-driven lesson plans with large language models,” in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–20.

[7] A. Dornburg and K. J. Davin, “ChatGPT in foreign language lesson plan creation: Trends, variability, and historical biases,” *ReCALL*, pp. 1–16, 2024.

[8] Q. Bao, J. Leinonen, A. Y. Peng, W. Zhong, G. Gendron, T. Pistotti, A. Huang, P. Denny, M. Witbrock, and J. Liu, “Exploring iterative enhancement for improving learnersourced multiple-choice question explanations with large language models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 28, 2025, pp. 28 955–28 963.

[9] V. Ashrafimoghari, N. Gürkan, and J. W. Suchow, “Evaluating large language models on the GMAT: Implications for the future of business education,” *arXiv preprint arXiv:2401.02985*, 2024.

[10] A. Pack, A. Barrett, and J. Escalante, “Large language models and automated essay scoring of English language learner writing: Insights into validity and reliability,” *Computers and Education: Artificial Intelligence*, vol. 6, p. 100234, 2024.

[11] V. Padmakumar, C. Yueh-Han, J. Pan, V. Chen, and H. He, “Beyond memorization: Mapping the originality-quality frontier of language models,” *arXiv preprint arXiv:2504.09389*, 2025.

[12] W. Merrill, N. A. Smith, and Y. Elazar, “Evaluating n-gram novelty of language models using rusty-dawg,” *arXiv preprint arXiv:2406.13069*, 2024.

[13] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang *et al.*, “Self-refine: Iterative refinement with self-feedback,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 46 534–46 594, 2023.