GENERATIVE AI FOR MULTIPLE CHOICE STEM ASSESSMENTS

C. Spirou Perdikoulias[1], C. Vance[1], S. Watt[2]

[1]Digital Education Company Ltd. (CANADA)
[2]University of Waterloo (CANADA)

Abstract

Artificial intelligence (AI) technology enables a range of enhancements in computer-aided instruction, from accelerating the creation of teaching materials to customizing learning paths based on learner outcomes. However, ensuring the mathematical accuracy and semantic integrity of generative AI output remains a significant challenge, particularly in Science, Technology, Engineering and Mathematics (STEM) disciplines. In this study, we explore the use of generative AI in which "hallucinations", typically viewed as undesirable inaccuracies, can instead serve a pedagogical purpose. Specifically, we investigate the generation of plausible but incorrect alternatives for multiple choice assessments, where credible distractors are essential for effective assessment design. We describe the Moebius platform for online instruction, with particular focus on its architecture for handling mathematical elements through specialized semantic packages that support dynamic, parameterized STEM content. We examine methods for crafting prompts that interact effectively with these mathematical semantics to guide the AI in generating high-quality multiple choice distractors. Finally, we demonstrate how this approach reduces the time and effort associated with creating robust teaching materials while maintaining academic rigour and assessment validity.

Keywords: Computer Aided Instruction, Generative AI, STEM Education, Online Assessment.

1 INTRODUCTION

Online learning environments have transformed the delivery of Science, Technology, Engineering and Mathematics (STEM) education by enabling dynamic, interactive, and adaptive instructional experiences. These environments support formative assessment practices that provide immediate feedback, guiding learners through iterative cycles of error and correction. Research in cognitive science suggests that learning is strengthened when students engage with "desirable difficulties," where effortful processing and corrective feedback support more durable understanding and retention [1].

Assessment plays a central role in this process. In STEM disciplines, high-quality assessments not only evaluate knowledge but also reinforce conceptual understanding and reveal misconceptions. However, creating such assessments, particularly those that are varied, rigorous, and aligned with learning objectives, remains time-intensive. Algorithmic question generation systems have addressed this challenge by producing parameterized problems and enabling automated grading through computation engines [2]. These systems allow instructors to generate multiple variations of a problem while maintaining mathematical consistency.

Recent advances in generative artificial intelligence (AI), particularly large language models (LLMs), introduce new opportunities to accelerate the creation of instructional content. These models can generate questions, explanations, and feedback at scale. However, their application in mathematics is limited by well-documented weaknesses in symbolic reasoning, multi-step problem solving, and logical consistency [3] [4] [5]. LLMs may produce outputs that appear plausible but are mathematically incorrect, reflecting pattern recognition rather than true understanding.

In this work, we explore a constrained application of generative AI in which these limitations can be leveraged productively. Specifically, we investigate the generation of plausible but incorrect alternatives, distractors, for multiple choice assessments. In this context, inaccuracies are not inherently problematic; rather, they can be valuable if they reflect realistic student misconceptions. This reframing allows AI "hallucinations" to serve a pedagogical function when combined with appropriate validation mechanisms.

We present an approach that integrates generative AI with a computation-enabled learning platform to support the creation of multiple choice STEM assessments. Our contributions include:

- A structured prompt design strategy for generating plausible distractors.
- A workflow that preserves mathematical semantics through structured representations.
- A validation pipeline that ensures correctness, uniqueness, and evaluability of generated content.

This work demonstrates how generative AI can be safely and effectively integrated into STEM assessment workflows by combining generative capabilities with deterministic validation.

1.1 Challenges of Generative AI in Mathematical Contexts

While generative AI has shown strong performance in natural language tasks, its application in mathematics remains limited. Mathematical reasoning requires precise symbolic manipulation, adherence to formal rules, and multi-step logical consistency. Studies have shown that LLMs struggle with these tasks, particularly as problem complexity increases [3] [4].

For instance, [3] conducted an evaluation of various advanced LLMs, including GPT-4o and Mixtral, using word problems typically encountered in Secondary-level mathematics. Their findings revealed that all exhibited errors of varying levels of accuracy, in spatial reasoning, strategic planning, and arithmetic. Notably, some models produced correct answers through incorrect logic, indicating a lack of genuine understanding in their problem-solving processes. “Common failure modes include unwarranted assumptions, over-reliance on numerical patterns, and difficulty translating physical intuition into mathematical steps” [3].

Furthermore, a recent paper [5] introduced the concept of the “Generative AI Paradox,” highlighting that while generative models can produce outputs resembling expert-level work, their capabilities “are not contingent upon—and can therefore exceed—their ability to understand those same types of outputs” [5], concluding that they often produce outputs without a true understanding of the underlying concepts. This divergence between generation and comprehension underscores the models’ reliance on pattern recognition rather than genuine reasoning, leading to outputs that may be syntactically plausible but semantically incorrect.

These limitations pose significant challenges for the direct generation of instructional content in STEM education, where correctness and rigour are essential.

1.2 Addressing AI Limitations Through Constrained Assessment Formats

Despite the well-documented limitations of generative AI in mathematical reasoning, not all assessment types are equally affected by these challenges. In particular, multiple choice questions offer a unique opportunity to effectively leverage generative AI within a highly structured format that naturally constrains the model’s output and simplifies the validation process.

Multiple choice assessments define a finite and predetermined set of response options, enabling greater control over both the content and the evaluation process. While often perceived as focusing on concept recognition, in many STEM disciplines these questions still require students to engage in full problem-solving processes to arrive at the correct answer. The key difference lies in the constrained output space, which simplifies validation and reduces ambiguity. This makes multiple choice assessments a promising context for integrating generative AI.

1.3 Designing and Generating Effective Distractors for STEM Assessment

Distractors in multiple choice questions serve a critical pedagogical function by targeting common errors and reinforcing conceptual understanding through contrastive reasoning. In STEM education, distractors are not merely incorrect answers; rather, they are plausible alternatives that require students to engage in the full problem-solving process to identify the correct response.

In mathematics, distractors often reflect systematic errors such as algebraic missteps, sign mistakes, or misunderstandings of function properties. These tailored “hallucinations” encourage students to distinguish between conceptual understanding and surface-level familiarity, thereby supporting deeper learning. Prior work [6] highlights the instructional value of well-designed distractors, noting that they “can elicit diagnostic information about students misconceptions, which is valuable for instructional decisions” [6].

Despite these benefits, distractors introduce pedagogical risks if not carefully constructed. Poorly designed distractors may be too obviously incorrect, reducing their effectiveness, or unintentionally

correct, compromising assessment validity. As noted in [7], distractors must strike a balance between plausibility and pedagogical intent, reinforcing the need for careful design and validation, particularly in mathematically rigorous domains.

To address this challenge, we developed a structured prompt strategy to guide generative AI models in producing credible distractors. This approach relies on curated prompt instructions that explicitly define the mathematical context, expected reasoning patterns, and common error types to be reflected in the generated options. By constraining both the structure and intent of the output, the model is guided toward producing alternatives that are not only syntactically valid, but also pedagogically meaningful.

In practice, this involves specifying error categories, such as incorrect identity application, sign inversion, or flawed intermediate steps, within the prompt itself, enabling the model to generate distractors that mirror authentic student misconceptions. Additionally, enforcing structured output formats ensures that generated responses can be reliably parsed and integrated into the assessment system. Empirical findings [4] support this approach, demonstrating that carefully designed prompts, particularly those that provide guidance analogous to tutoring, can significantly improve the quality and reasoning consistency of AI-generated outputs.

Together, these strategies enable the generation of distractors that are both plausible and instructionally valuable, while maintaining alignment with the underlying mathematical concepts and assessment objectives.

2 METHODOLOGY

2.1 System Context: Computation-Enabled Assessment Platform

To support the discussion of AI-driven question generation, we employ a computation-enabled learning platform designed for STEM instruction and assessment, Möbius. The selected platform integrates a computer algebra system (CAS) with an authoring environment that supports parameterized question generation and automated evaluation.

2.1.1 Platform Overview

Möbius is a cloud-hosted, browser-accessible platform designed for the delivery of STEM instruction and assessment. The platform supports a range of question types, including mathematical formula entry, algorithmically generated problems, and multiple choice questions. A key feature is the ability to define variables and expressions programmatically, allowing different learners to receive distinct instances of the same problem.

Mathematical expressions are represented using structured formats such as LaTeX or MathML, preserving both visual presentation and semantic meaning. This enables symbolic manipulation, equivalence checking, and automated grading.

The Multiple Choice (MC) question type in Möbius allows for the presentation of a question prompt alongside a set of predefined response options, supporting both static and dynamic content. One or more answers can be designated as correct, and each option can include mathematical expressions and may be generated dynamically based on parameterized variables. Other characteristics include:

- Support for dynamic values and variables in prompts and answer options.
- Customizable feedback at the level of individual choices.
- Support for single- and multiple-answer configurations.
- Mathematical notation support for accurate display and preservation of semantics of expressions and equations.

This structure provides a controlled environment for integrating generative AI, as outputs can be constrained and evaluated systematically.

2.1.2 Möbius System Architecture

The Möbius platform is built on a layered architecture, organized to separate the presentation of content from the execution of mathematical computations. This structure supports system scalability, modular development, and integration of third-party services. Within the context of using Generative AI

to author questions in Möbius, the architecture is as follows: The Presentation Layer is responsible for handling user interaction and orchestrating computational requests. It includes:

- **Möbius UI:** Manages the collection of user input and the delivery of question content using Möbius Algorithm Syntax.
- **Möbius Question Generator:** Parses input from the Möbius UI, substitutes parameters into commands, and directs computation tasks to the appropriate service. It also processes the output from the Services Layer and prepares it for display via the Möbius UI.

The Services Layer executes the computational logic needed for question generation and evaluation. It comprises:

- **Vertex AI:** A third-party multi-model AI Machine Learning Engine that executes prompts and instructions from the Möbius Question Generator.
- **Möbius Math Engine:** A core service, orchestrating the execution of algorithmic expressions derived from questions authored in the proprietary Möbius syntax. These expressions can encompass nested code written in Maple, Python, or Java, allowing for complex and multifaceted mathematical computations within a single question.

When a user submits a request, the Möbius Question Generator routes it to the AI engine (for prompts) and then to the appropriate computation service, with results returned and displayed with feedback in the UI, as summarized in Figure 1.

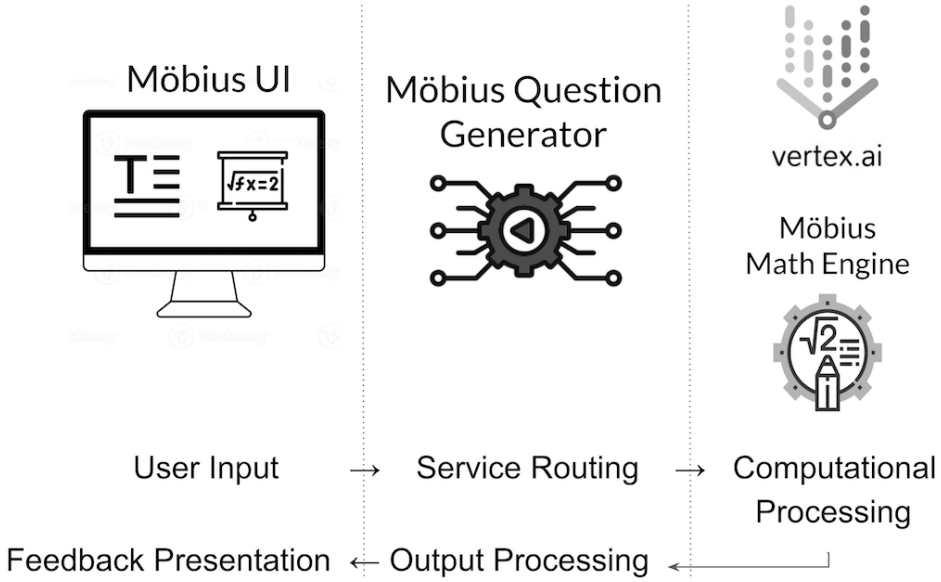

Figure 1: AI-Generated Multiple Choice Questions in Möbius: High-level system diagram.

This structure allows for a combination of real-time symbolic computation, parameterized content rendering, and automated evaluation in a manner that is extensible and adaptable to new question generation techniques.

2.2 Verifying the Uniqueness of the Correct Option

Ensuring that each multiple choice item contains a single, clearly identifiable correct answer is essential for both fairness and validity. To support this, we designed a verification process that incorporates the Möbius Math Engine within the Question Generation workflow. At the time of writing, this component of the overall question generation process is architectural, and not fully implemented.

In this design, after distractors are generated, all answer options are submitted to the engine for symbolic evaluation. If more than one option simplifies to the correct answer or represents a valid form of the same solution, the design allows for the prompt to be adjusted, and the distractors regenerated. The designed workflow is shown in Figure 2.

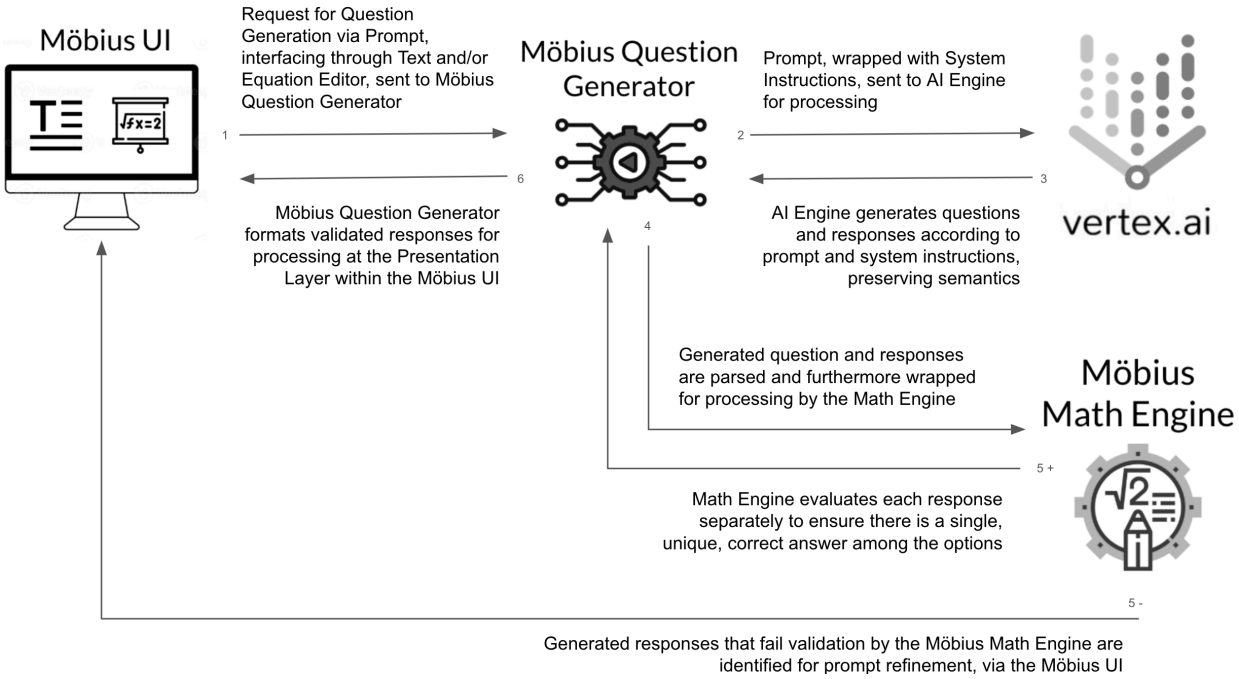

Figure 2: A flow diagram showing the validation process of multiple choice options within the Möbius computation pipeline

Another verification method involves encoding logical constraints directly into the prompt:

> *Ensure only one answer evaluates to the correct result; all other responses should reflect common incorrect reasoning paths without coinciding with the correct value.*

This strategy, encompassing prompt design plus post-processing, offers a robust means to maintain answer uniqueness.

2.3 Preserving Mathematical Semantics

In mathematics education, preserving the semantic structure of expressions is essential for ensuring both instructional accuracy and computational fidelity. This becomes especially important when integrating generative AI into STEM assessments, where ambiguity in representation can lead to misinterpretation and incorrect evaluation.

Möbius addresses this through a built-in equation editor that enables authors and students to enter expressions using traditional two-dimensional notation. These expressions are encoded in a mathematical markup language that preserves both visual structure and hierarchical meaning. For example, a derivative such as $\frac{d}{dx}\sin x^2$, retains operator precedence, function nesting, and variable relationships, ensuring unambiguous interpretation.

This semantic preservation is particularly valuable in workflows involving large language models (LLMs). By generating outputs in structured formats such as MathML or LaTeX, AI-generated content can be reliably integrated into systems that support symbolic manipulation, automated grading, and feedback generation. In Möbius, such expressions are processed directly by the question generator and computation engine, enabling validation and transformation without reliance on natural language interpretation, a known source of error in LLMs [8].

Maintaining mathematical intent throughout the generation and assessment pipeline ensures alignment between instructional goals and computational accuracy.

3 RESULTS: SYSTEM VALIDATION FOR RELIABILITY AND REPEATABILITY

The integration of generative AI models into structured STEM assessment platforms necessitates robust testing to ensure that the resulting workflows are reliable, repeatable, and pedagogically sound. In the context of Möbius, we focus on validating the full authoring pipeline, starting from the initial AI

prompt to the delivery and validation of questions within the platform. This section outlines the multi-layered testing approach used to ensure the technical and instructional integrity of AI-assisted question generation.

3.1 Validating Mathematical Encoding in Prompts and Responses

Accurate handling of mathematical expressions is foundational to STEM question generation. In this workflow, expressions authored in the Möbius Equation Editor are translated into a mathematical markup language for transmission to Vertex AI. Since these encodings preserve both visual and semantic meaning, any distortion compromises question integrity.

To confirm the fidelity of mathematical inputs and outputs, we perform testing using representative prompts that include complex expressions, such as nested fractions, function calls (e.g., $\sin\left(\frac{\pi}{4}\right)$), and symbolic algebra. Each test ensures:

- Mathematical content is transmitted without alteration.
- Returned markup is structurally complete and interpretable.
- Semantic meaning is preserved without encoding errors.

This ensures that expressions retain their intended meaning throughout the pipeline.

3.2 Robust Parsing and Error Handling in AI-Generated Workflows

Vertex AI returns structured outputs in JavaScript Object Notation (JSON), which must be reliably parsed into Möbius data structures representing key question components, including stems, answer options, feedback, and correct answer designations. Validation at this stage ensures schema consistency, accurate extraction of all elements, and correct rendering of embedded mathematical expressions. Test cases also include malformed responses, such as missing fields or ambiguous answer identifiers, to confirm that errors are detected and handled explicitly rather than silently.

To further ensure robustness, failure scenarios are simulated within the Möbius computation pipeline. These include invalid prompts, API rate limits, and unsupported configurations. When such conditions arise, Möbius detects the issue, presents clear user-facing error messages, and logs relevant metadata to support debugging and traceability. This approach aligns with the overall validation workflow illustrated in Figure 2, ensuring that both structural integrity and system resilience are maintained throughout the AI-assisted question generation process.

3.3 Validating AI-Generated Content

The final validation layer evaluates correctness, uniqueness, and evaluability using the Möbius Math Engine. Unlike manual evaluation methods described in [9], this approach focuses on computational validation.

The key goals at this stage include:

- Verifying that the correct answer satisfies evaluation criteria.
- Ensuring distractors produce distinct incorrect results.
- Confirming that feedback logic executes correctly.

Operating on semantically encoded input, the engine evaluates answers as mathematical objects, enabling transformation, simplification, and equivalence checking. As illustrated in Figures 2 and 3, each option is parsed, evaluated, and verified to ensure exactly one correct answer. Invalid results trigger prompt refinement, ensuring that final questions are both mathematically accurate and pedagogically sound.

```
FUNCTION validateSingleAnswer(question)
    // 1. Create an empty list to store answers that are verified as correct
    SET correctAnswersList TO new empty List
    // 2. Loop through every answer provided in the question
    FOR EACH answer IN question.getAnswers()
        // 3. Use an external engine to check if this answer is a valid solution for the equation.
        IF mobiusMathEngine.validAnswer(question, answer) IS TRUE THEN
            // 4. If it is valid, add it to our list
                ADD answer TO correctAnswersList
        END IF
    END FOR
    // 5. After checking all answers, inspect the list of valid solutions
    // 6. If the list is empty, no correct answer was found. This is an error.
    IF correctAnswersList is empty THEN
            THROW Exception("No correct answer found for question: ", question, correctAnswersList)
    // 7. If the list has more than one, the question is ambiguous. This is an error.
    ELSE IF size of correctAnswersList IS GREATER THAN 1 THEN
            THROW Exception("Multiple correct answers found for question: ", question, correctAnswersList)
    END IF
    // 8. If no errors were thrown, the list must have exactly one answer.
    // Return TRUE.
    RETURN (size of correctAnswersList IS EQUAL TO 1)

END FUNCTION
```

Figure 3: Logic structure for Math Engine evaluation

4 CONCLUSIONS

We have explored the use of generative AI in the development of multiple choice assessments for mathematical subjects, where structured formats and well-defined concepts help mitigate many of the challenges associated with AI-generated content. This work was conducted within the context of the Möbius platform for online learning.

Our results demonstrate that generative models can produce plausible and pedagogically valuable distractors, despite their known limitations in mathematical reasoning and symbolic manipulation.

When applied within a constrained and validated framework, generative AI offers several practical advantages. It:

- reduces the time and effort required to create high-quality assessments,
- supports content variation through parameterization, and
- preserves academic rigour through validation at key integration points.

This approach not only addresses practical challenges in content development, but also reframes AI “hallucinations” as a useful mechanism for generating convincing distractors. As noted by Alhazmi et al. [9], while recent advances in large language models have improved contextual coherence, they continue to struggle with producing distractors that are both pedagogically valid and semantically distinct. This reinforces the importance of validation workflows that maintain mathematical rigour while leveraging AI’s generative capabilities.

Mathematical domains are particularly well-suited to a generate–validate workflow, as candidate solutions can often be verified using numerical or symbolic computation. Future work will focus on extending this approach for commercial applications by ensuring that validation processes are robust, transparent, and scalable. While this study employed a sequenced validation loop within Möbius to verify answer uniqueness, mathematical coherence, and semantic integrity, further work is required to formalize and automate this pipeline. This includes the development of automated test suites for symbolic equivalence, correctness, and feedback-driven prompt refinement.

Overall, this work highlights the potential for generative AI to significantly accelerate the creation of STEM assessments when combined with discipline-specific platforms and rigorous validation. Formalizing the validation layer represents a key step toward enabling reliable, large-scale deployment.

REFERENCES

[1] John Dunlosky, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham. Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest,* 14(1):4–58, 2013.

[2] Alice Barana, Marina Marchisio, and Matteo Sacchet. Interactive feedback for learning mathematics in a digital learning environment. *Education Sciences*, 11(6):279–307, 2021.

[3] Johan Boye and Bierger Moëll. Large language models and mathematical reasoning failures. (arXiv:2502.11574), 2025.

[4] Jing Qian, Hong Wang, Zekun Li, Shiyang Li, and Xifeng Yan. Limitations of language models in arithmetic and symbolic induction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 9285–9298, 01 2023.

[5] Peter West, Ximing Lu, Nouha Dziri, Faeze Brahman, Linjie Li, Jena D. Hwang, Liwei Jiang, Jillian Fisher, Abhilasha Ravichander, Khyathi Chandu, Benjamin Newman, Pang Wei Koh, Allyson Ettinger, and Yejin Choi. The generative AI paradox: "What it can create, it may not understand". In *The Twelfth International Conference on Learning Representations*, 2024.

[6] T. M. Haladyna, S. M. Downing, and M. C. Rodriguez. A review of multiple-choice item-writing guidelines for classroom assessment. *Applied Measurement in Education*, 15(3):309–334, 2002.

[7] Marie Tarrant, James Ware, and Ahmed M. Mohammed. An assessment of functioning and non-functioning distractors in multiple-choice questions: a descriptive analysis. *BMC Medical Education*, 9(40), 2009.

[8] R. Yamauchi, S. Sonoda, A. Sannai, and W. Kumagai. LPML: LLM-prompting markup language for mathematical reasoning. (arXiv:2309.13078), 2023.

[9] Elaf Alhazmi, Quan Sheng, Wei Emma Zhang, Munazza Zaib, and Fahad Alhazmi. Distractor generation in multiple-choice tasks: A survey of methods, datasets, and evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.