Evaluating the Maze-Solving Capabilities of Pretrained Language Models

Gabriel Del Castillo*, Nate Webster†
Department of Computer Science
Colorado School of Mines
Golden, CO, USA
*gdelcastillo@mines.edu, †nwebster@mines.edu

***Abstract*—**

I. Introduction

Large Language Models (LLMs) such as GPT-4 and LlaMa 3 are able to produce remarkably accurate results when faced with various natural language processing tasks []. As the scale and reach of these models have expanded, so has the focus on developing benchmarks to measure the limits and capabilities of LLMs []. However, there has been little research examining how pretrained language models handle maze solving tasks, along with identifying possible common failure patterns. The ability to solve maze-like structures has long served as a way to evaluate the cognitive and algorithmic performance of various subjects, with research ranging from rodent navigation [1], to spatial understanding in infants [2], to path planning for robotic systems [3].

Motivated by this, as well as the prospect of integrating LLMs into physical robotic agents capable of speech, we analyze the responses of four different language models after being prompted with one of several solvable maze configurations, evaluate the validity and optimality of these answers, and measure if parameter count, model distribution type, or maze representation have an impact on task performance.

II. OLD PROPOSAL STARTS HERE

For our project, we plan on building on the work by Ivanitskiy et al. [7] to generate solvable maze configurations, each of which will be provided to a number of pretrained language models through API requests. To determine whether the solution provided by the language model is correct or not, these will be verified by either the solution found in [7] (in cases with a unique solution), or a breadth-first search (BFS) solver (in cases with multiple solutions). All incorrect solutions (or replies detailing the model's inability to solve the maze) will be manually analyzed to discover failure patterns.

The list of models to be evaluated has not been explicitly defined yet; however, it will be divided into three categories of models — open source (e.g., Groq's Qwen 2.5 32B, DeepSeek's r1), open weights (e.g., Mistral's Nemo) and proprietary models (e.g. OpenAI's gpt-4, Anthropic's Claude 3). Each prompt will consist of the following question, along with an ASCII representation of the maze.

Task	Completion Date
Initial Setup	04/04/25
Evaluation Code Implementation	04/13/25
Paper Draft	04/20/25
Final Touch-Ups & Report	04/29/25

TABLE I: Rough Timeline

"For the following maze, given a start point (denoted by an S), a goal (denoted by an X), and inaccessible areas (denoted by # characters), please provide a path going from start to end"

The specific wording of the prompt, along with the representation fo the maze, are potential avenues for further research in this project. The maze-dataset package published by [7] includes ASCII, pixel image, and token-based representations of any given maze. Each of these would be provided to a language model, and their corresponding responses would be compared accordingly.

Table I provides a rough estimate of the tasks to be completed, as well as completion dates for each one. Granted, these are just estimates and are subject to change as issues are found and tasks are completed. All dates were listed with an assumed final project submission date of April 30th, 2025.

III. Minimum Viable Example

1. Select a language model to evaluate from [4].
2. Obtain a valid API key for the selected model, along with an example maze from [7] with a single, unique solution.
3. Set up a coding project with the required libraries, dependencies, and environment variables.
4. Write the necessary code for providing the model with the maze and aforementioned prompt.
5. Manually compare the model's response to the actual solution, as detailed in [7].

I estimate the total amount of time this process will take is 3 hours, even when accounting for the possibility of roadblocks.

IV. Hypotheses

H1: *Model performance scales at minimum linearly with parameter count*

H2: *Distribution type — open source, open weights, proprietary — has minimal effect on model performance*

V. Proposed Method

To test my hypotheses, a variety of models will be selected from those listed in [4], including various parameter counts and an approximately equal number of models across each distribution type previously established. As detailed previously, the prompt & mazes the models are given will remain constant and serve as the control throughout the experiment. In turn, the responses of these models will be recorded and observed on a basis of correctness. Every correct solution to the maze provided will contribute positively to the "success rate" of a model. On the other hand, an incorrect solution, or inability to provide a solution all-together, negatively impacts the aforementioned success rate and will be manually analyzed for potential trends.

I expect to find that a mix of correct and incorrect solutions, including a few instances where a model is unable to provide any solution at all. To account for the unlikely instance where the models selected provide correct answers exclusively, constraints may be added to the original message, by way of the prompt (e.g., limiting the number of moves, asking the model to go through inaccessible areas) or the maze (e.g., providing unsolvable maze configurations).

VI. LLM Usage

The primary way that I used language models throughout the writing of this proposal was to find references to relevant literature for using maze-solving as a way to measure intelligence. A transcript of this usage can be found here: https://claude.ai/share/9628e541-21d3-403f-a631-3b430a388034

References

[1] E. C. Tolman, "Cognitive Maps in Rats and Men," Psychol. Rev., vol. 55, no. 4, pp. 189-208, Jul. 1948.

[2] J. Piaget and B. Inhelder, "The Child's Conception of Space," Int. J. Psychol., vol. 2, no. 3, pp. 241-242, 1967.

[3] Path-finding Algorithms: A. Stentz, "Optimal and Efficient Path Planning for Partially Known Environments," in IEEE International Conference on Robotics and Automation, 1994, pp. 3310-3317.

[4] "LLM Index — Promptmetheus," Promptmetheus, Mar. 13, 2025. https://promptmetheus.com/resources/llm-index (accessed Mar. 26, 2025).

[5] A. Gawrylewski, "The AI Future Is Here," Scientific American, Mar. 04, 2025. https://www.scientificamerican.com/article/the-ai-future-is-here/

[6] Mucci, Tim. "The Future of AI: Trends Shaping the next 10 Years." Ibm.com, 11 Oct. 2024, www.ibm.com/think/insights/artificial-intelligence-future.

[7] M. I. Ivanitskiy et al., "A Configurable Library for Generating and Manipulating Maze Datasets," arXiv:2309.10498 [cs.LG], Sep. 2023.