



**Department of Electrical and Computer Engineering
North South University**

CSE498R Research Report

PuzzleMaster: Exploring LLM Performance with Prompting

GROUP MEMBERS:

NAME	ID
TAUFIQUL ALAM	1921606042
MD TAMJID HOSAIN	2012952642
SUMMIYA SUNJIDA KASHPIA	2012712042

Faculty Advisor:

Dr. Mohammad Ashrafuzzaman Khan [AzK]

Assistant Professor

ECE Department

SUMMER, 2024

LETTER OF TRANSMITTAL

June, 2023

To

Dr. Mohammad Abdul Matin

Chairman,

Department of Electrical and Computer Engineering

North South University, Dhaka

Subject: Submission of CSE498R Research Report on “PuzzleMaster: Exploring LLM Performance with Prompting”

Dear Sir,

With due respect, we would like to submit our CSE498R Research Report on “PuzzleMaster: Exploring LLM Performance with Prompting” as a part of our BSc program. The report deals with the development and evaluation of PuzzleMaster an AI bot which is created to solve a wide range of puzzles and riddles. We found this project to be very beneficial since it gave us practical experience evaluating the problem-solving skills of various LLMs and their potential for practical uses. We tried to the maximum competence to meet all the dimensions required from this report.

We will be highly obliged if you kindly receive this report and provide your valuable judgment. It would be our immense pleasure if you find this report useful and informative to have an apparent perspective on the issue.

Sincerely Yours,

Taufiqul Alam

.....

Student Name1

ECE Department

North South University, Bangladesh

Md Tamjid Hosain

.....

Student Name2

ECE Department

North South University, Bangladesh

Summiya Sunjida Kashpia

.....

Student Name3

ECE Department

North South University, Bangladesh

APPROVAL

Taufiqul Alam (ID: 1921606042), Md Tamjid Hosain (ID: 2012952642) and Summiya Sunjida Kashpia (ID: 2012712042) from Electrical and Computer Engineering Department of North South University, have worked on the CSE498R Research Project titled “PuzzleMaster: Exploring LLM Performance with Prompting” under the supervision of Dr. Mohammad Ashrafuzzaman Khan [AzK] partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

Supervisor’s Signature

.....

Dr. Mohammad Ashrafuzzaman Khan [AzK]

Assistant Professor

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

Chairman’s Signature

.....

Dr. Mohammad Abdul Matin

Professor & Chair

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

DECLARATION

This is to declare that this project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All project related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been properly acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students' names & Signatures

1. Student Name1

Taufiqul Alam

2. Student Name2

Md Tamjid Hosain

3. Student Name3

Summiya Sunjida Kashpia

ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Dr. Mohammad Ashrafuzzaman Khan [AzK], Assistant Professor, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

ABSTRACT

PuzzleMaster: Exploring LLM Performance with Prompting

PuzzleMaster is an Ai bot that can solve any kind of riddles and puzzles (Logical puzzles, Riddles, Mathematical puzzles and Word-based puzzles) we throw at it which helps us evaluate the problem-solving abilities of five different LLMs we worked with (Gemini, Cohere, Llama2.7, Llama3 and Llama 3.2 instruct) using prompting. After successfully getting results we compare them by their accuracy, completion rate, response time, response quality, creativity and errors. The aim of our project is to evaluate the problem-solving capabilities of different LLM models using brain teasers and complex riddles which needs humanlike humor and thinking to solve. This helps us to choose the right model for our tasks to be done more effectively in real world scenarios.

TABLE OF CONTENTS

LETTER OF TRANSMITTAL	2
APPROVAL	4
DECLARATION	5
ACKNOWLEDGEMENTS	6
ABSTRACT	7
LIST OF FIGURES	10
LIST OF TABLES	11
Chapter 1 Introduction	12
1.1 Background and Motivation	12
1.2 Purpose and Goal of the Project	12
1.3 Organization of the Report	13
Chapter 2 Research Literature Review	14
2.1 Existing Research and Limitations	14
Chapter 3 Methodology	15
3.1 System Design	15
3.2 Hardware and/or Software Components	15
3.3 Hardware and/or Software Implementation	17
Chapter 4 Investigation/Experiment, Result, Analysis and Discussion	18
4.1 Experiments Performed	18
4.2 Results & Discussion	19
Chapter 5 Impacts of the Project	24
5.1 Impact of this project on societal, health, safety, legal and cultural issues	24
5.2 Impact of this project on environment and sustainability	24
Chapter 6 Project Planning and Budget	25

Chapter 7 Complex Engineering Problems and Activities	27
7.1 Complex Engineering Problems (CEP).....	27
7.2 Complex Engineering Activities (CEA)	28
Chapter 8 Conclusions	29
8.1 Summary	29
8.2 Limitations	29
8.3 Future Improvement.....	30
References	31

LIST OF FIGURES

Figure 1. Flowchart of System Design.	15
Figure 2. Accuracy of different models	20
Figure 3. LLM Models Response.	20
Figure 4. Gantt Chart.	25
Figure 5. Future Consideration.	30

LIST OF TABLES

TABLE I. A SAMPLE SOFTWARE/HARDWARE TOOLS TABLE	16
TABLE II. ACCURACY of EACH MODEL USING PROMPTING TABLE	19
TABLE III. COMPLETION TIME TABLE	21
TABLE IV. RESPONSE TIME TABLE	22
TABLE V. ERROR RATE TABLE	22
TABLE VI. COMPLEX ENGINEERING PROBLEM ATTRIBUTES TABLE	27
TABLE VII. COMPLEX ENGINEERING PROBLEM ACTIVITIES TABLE	28

Chapter 1 Introduction

1.1 Background and Motivation

Advancements in Artificial Intelligence (AI) have made puzzle-solving a critical benchmark for evaluating logical reasoning, creativity, and problem-solving capabilities in Large Language Models (LLMs). Exploring LLMs' abilities in this domain provides valuable insights into their potential and limitations, particularly in complex reasoning tasks [1].

This project, PuzzleMaster, assesses five state-of-the-art LLMs—Gemini, Cohere, Llama 2.7, Llama 3, and Llama 3.2 Instruct—on solving logical, mathematical, and word-based puzzles. By categorizing puzzles as rule-based or rule-less, we use prompts to evaluate these models and identify disparities between their performance and human-like reasoning.

This project is motivated by real-world applications where efficiency can be increased by choosing the appropriate AI model for tasks requiring complex problem-solving. In order to close the gap between AI and human cognitive capacities and enhance AIs capacity for reasoning and original problem-solving our findings emphasize the necessity for more comprehensive datasets and better methodologies.

1.2 Purpose and Goal of the Project

The purpose of PuzzleMaster is to evaluate and compare the problem-solving abilities of five advanced LLMs—Gemini, Cohere, Llama 2.7, Llama 3, and Llama 3.2 Instruct—by testing them on various puzzles. The objective is to evaluate their performance using prompts, comparing their capacity to solve rule-based and rule-less puzzles while paying particular attention to accuracy, creativity, response time, and error rates. A thorough assessment of these models' advantages and disadvantages in handling challenging reasoning tasks is the projects contribution. The use of a distinct taxonomy for classifying puzzles and a systematic assessment framework that improves comprehension of LLMs capacity for human-like reasoning make this project unique.

1.3 Organization of the Report

This report is organized into several chapters, each focusing on different aspects of the project. Chapter 2 delves into the research literature, offering insights into the current landscape of LLMs and their role in puzzle-solving. Chapter 3 lays out the methodology, outlining how the models were selected, the puzzle types used, and the evaluation approach. In Chapter 4, we present the heart of the project: the investigation, results, and in-depth analysis, followed by a discussion on the implications of the findings. Chapter 5 explores the broader impacts of the project, while Chapter 6 provides a detailed look at project planning, budgeting, and resources. Chapter 7 highlights the complex engineering problems and activities encountered throughout the project. Finally, Chapter 8 concludes the report, reflecting on the key discoveries and suggesting potential avenues for future exploration.

Chapter 2 Research Literature Review

2.1 Existing Research and Limitations

Recent research has examined the problem-solving capabilities of Large Language Models (LLMs) in solving logical, mathematical, and word-based puzzles. Giadikiaroglou et al. [1] conducted a survey on puzzle solving with LLMs, noting their strong performance in structured problems but highlighting difficulties in puzzles requiring humor, abstract reasoning, or creativity. Similarly, Li et al. [2] assessed LLMs' ability to solve logical puzzles using Minesweeper as a case study. They found LLMs excelled in simple problems but struggled with multi-step reasoning and ambiguity, common in real-world puzzles requiring human-like thinking.

Limitations:

1. **Inconsistent Performance on Complex Puzzles:** LLMs perform well on simple puzzles but struggle with complex, abstract puzzles requiring multi-step reasoning and human-like humor [1][2].
2. **Narrow Puzzle Scope:** Most studies focus on specific puzzle types (e.g., Minesweeper) without addressing a broad range of puzzles, including mathematical and word-based riddles [1][2].
3. **Lack of Comprehensive Model Comparisons:** Existing research typically evaluates individual models without comparing multiple LLMs across various puzzle domains [1][2].
4. **Limited Evaluation Criteria:** Studies often prioritize accuracy or speed, neglecting important factors like response quality, creativity, and error handling in human-like reasoning tasks [1][2]

This limitation has motivated our research to evaluate the performance of LLMs across a broader spectrum of puzzles, including riddles and logical challenges that require human-like reasoning, humor, and creativity. We aim to understand how different LLMs handle these complexities, ensuring a more comprehensive evaluation of their problem-solving capabilities.

Chapter 3 Methodology

3.1 System Design

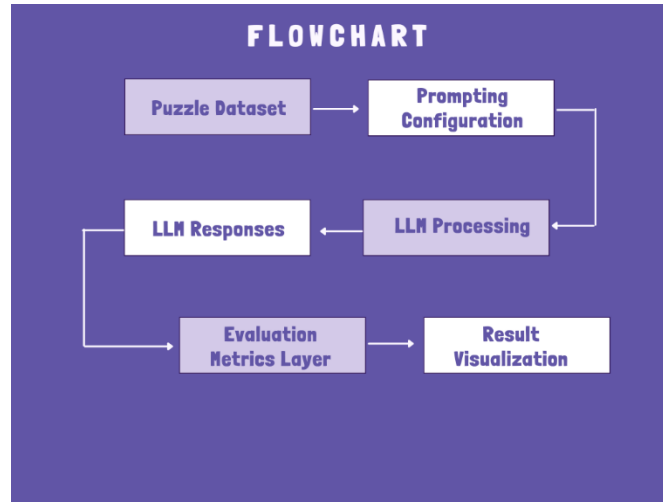


Figure1: Flowchart of System Design

The project design begins with the **Puzzle Dataset**, which includes a variety of riddles and puzzles. These puzzles are then fed into the system, where **Prompting Configuration** is applied to structure the input for the different Large Language Models (LLMs). After this, the LLMs process the puzzles and generate **LLM Responses**. These responses are then passed through an **Evaluation Metrics Layer**, where they are analyzed based on various criteria such as accuracy, creativity, and response time. Finally, the results are visualized to compare the performance of different LLMs and identify the model that excels in solving the puzzles. This structured approach allows for a comprehensive evaluation of the LLMs' problem-solving abilities.

3.2 Hardware and/or Software Components

For the **PuzzleMaster** project, which evaluates the problem-solving capabilities of Large Language Models (LLMs), various techniques were utilized. The project relies on a diverse puzzle dataset consisting of riddles, logical puzzles, and mathematical challenges, which are preprocessed using tokenization, normalization, and vectorization techniques. The core models employed are **Gemini**, **Cohere**, **Llama2.7**, **Llama3**, and **Llama3.2 instruct**, which were selected for their strong NLP capabilities. Evaluation of the models is based on accuracy, creativity, response time, and

completion rate, with custom metrics calculated using Python libraries. The models are integrated through **Hugging Face Transformers** and **LangChain**, and the results are presented via a **Streamlit** interface. Additionally, **FastAPI** is used to build an API for seamless interaction, while **Pandas** and **SQLite** are employed for data management and storage.

TABLE I. A SAMPLE SOFTWARE/HARDWARE TOOLS TABLE

Tool	Functions	Other similar Tools (if any)	Why selected this tool
Gemini	Language model for reasoning and puzzle-solving	Cohere, Llama	Chosen for its strong problem-solving abilities in complex tasks.
Cohere	Language model for natural language generation	Gemini, Llama	Selected for its versatility and high performance in puzzle solving
Llama2.7	Open-source model for general language tasks	Gemini, Cohere	Used for its robustness and open-source nature.
Llama3	Advanced model for complex logic and reasoning	Gemini, Cohere	Selected to test higher-level cognitive abilities.
Llama3.2 instruct	Model with better instruction-following capabilities	Gemini, Cohere, Llama3	Chosen for its precision in following specific puzzle instructions.
Python	Programming language for AI model integration and data handling	R, Julia, Matlab	Selected for its extensive libraries and data processing capabilities.
Streamlit	Tool for creating interactive interfaces	Flask, Dash	Used for its simplicity and quick deployment of user interfaces.

3.3 Hardware and/or Software Implementation

The project focuses on integrating five LLMs (**Gemini, Cohere, Llama2.7, Llama3, and Llama 3.2 instruct**) using **Python**. The models handle various puzzles and riddles, with evaluations based on accuracy, response time, and creativity. The software was developed using libraries like **transformers** and **streamlit** to provide a user interface and manage model interactions. No hardware was used in this AI-based project.

Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

4.1 Experiments Performed

The primary goal of this project was to evaluate the problem-solving abilities of five LLMs—Gemini, Cohere, Llama2.7, Llama3, and Llama 3.2 instruct—across a diverse range of puzzles, including logical, mathematical, and word-based riddles. The experiments were designed to test the models with prompts and assess their performance under different conditions.

Experimental Setup

- **Models Tested:** Gemini, Cohere, Llama2.7, Llama3, and Llama 3.2 instruct.
- **Puzzle Types:** Logical puzzles, mathematical riddles, word-based puzzles, and complex brain teasers.
- **Evaluation Criteria:**
 - **Accuracy:** Percentage of correct solutions provided by each model.
 - **Completion Rate:** Number of puzzles the models were able to complete.
 - **Response Time:** The time taken by each model to generate an answer.
 - **Response Quality:** Assessing clarity, detail, and usefulness of answers.
 - **Creativity:** Ability to provide innovative solutions for puzzles requiring more than just straightforward logic.
 - **Error Rate:** Frequency of mistakes or incomplete answers.

4.2 Results & Discussion

Accuracy of Each Model Using Prompting

The accuracy of each model with structured prompts is summarized in the table below:

TABLE II. ACCURACY OF EACH MODEL USING PROMPTING TABLE

Model	Accuracy with prompting %
Cohere	95%
Gemini	90%
Llama2.7	80%
Llama3	85%
Llama3.2 instruct	90%

Analysis: The results indicate that all models performed well with structured prompting. Cohere achieved the highest accuracy (95%), followed by Gemini and Llama 3.2 Instruct, both with an accuracy of 90%. Llama3 and Llama2.7 had slightly lower performance at 85% and 80%, respectively. These findings emphasize the significant role of input quality, as structured prompting plays a crucial part in enhancing the models' ability to solve complex puzzles effectively.

Below are figures comparing the accuracy of different models:

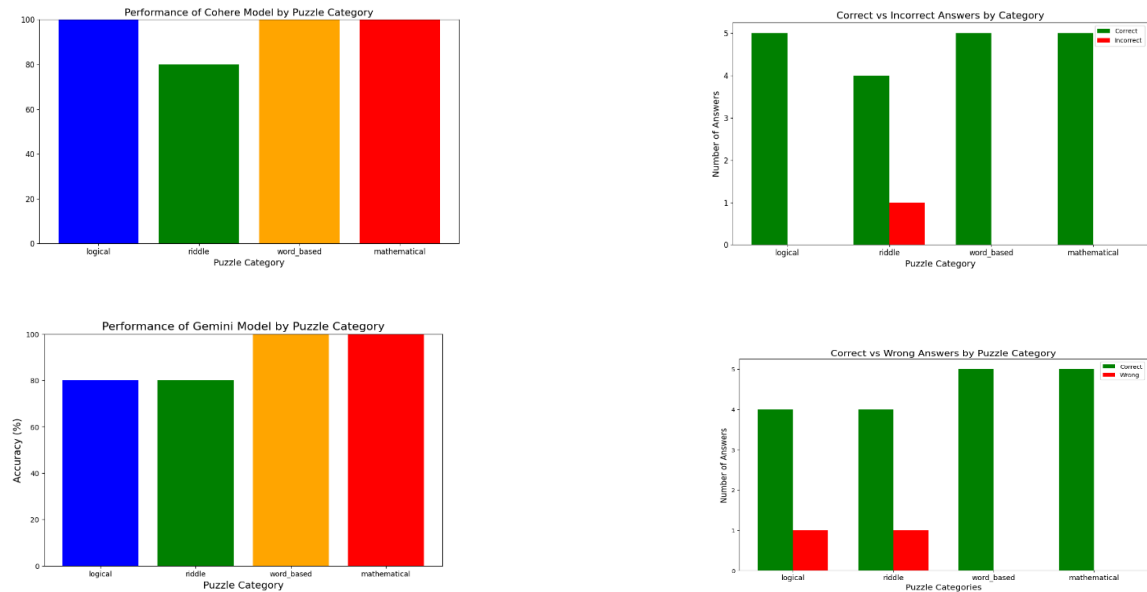


Figure 2: Accuracy of different models

Teaser: What begins with T, ends with T, and has T in it?
Model Answer: A teapot.
Expected Answer: A teapot.
Result: Correct
Time Taken: 0.89 seconds

Teaser: What has to be broken before you can use it?
Model Answer: An egg.
Expected Answer: An egg.
Result: Correct
Time Taken: 0.91 seconds

Teaser: What has keys but can't open locks?
Model Answer: A piano.
Expected Answer: A piano.
Result: Correct
Time Taken: 0.91 seconds

Teaser: What has a heart that doesn't beat?
Model Answer: The answer to the riddle is: "Artichoke".
Expected Answer: Artichoke.
Result: Correct
Time Taken: 0.90 seconds

User: You see a boat filled with people. It has not sunk, but when you look again, you don't see a single person. Why?
Gemini: Answer: All the people on the boat are married.

User: A man is found hanging from a ceiling with water on the floor. What happened?
Gemini: Answer: The man was hanging from an ice cube that melted.

User: I'm tall when I'm young, and I'm short when I'm old. What am I?
Gemini: Answer: A candle

User: What has a face, two hands, but no arms or legs?
Gemini: Answer: A clock

User: What gets wetter the more it dries?
Gemini: Answer: A towel

Teaser: A man is found hanging from a ceiling with water on the floor. What happened?
Model Answer: The man did not commit suicide by hanging, but rather he stood on a block of ice to tie the rope, and the ice melted.
Expected Answer: He stood on an ice block, which melted.
Result: Correct
Time Taken: 0.96 seconds

Teaser: A man pushes his car to a hotel and tells the owner he's bankrupt. Why?
Model Answer: The man was playing Monopoly.
Expected Answer: Because he's playing Monopoly.
Result: Correct
Time Taken: 0.88 seconds

Figure 3: LLM Models Response

Completion Time

We also measured the models’ completion rates and the quality of their responses. These metrics were essential in understanding how well the models could provide meaningful answers and handle multi-step reasoning tasks.

TABLE III. COMPLETION TIME TABLE

Model	Completion time with prompting
Cohere	23.06 seconds
Gemini	36.79 seconds
Llama2.7	39.56 seconds
Llama3	37.89 seconds
Llama3.2 instruct	34.56 seconds

The completion times for various models with prompting are as follows: Cohere takes the least time, with a completion time of 23.06 seconds. In contrast, Gemini has a slower response time of 36.79 seconds. The Llama models show slightly longer completion times, with Llama2.7 at 39.56 seconds and Llama3 at 37.89 seconds. The Llama3.2 Instruct model has a completion time of 34.56 seconds, which is a bit faster than the other Llama versions but still slower than Cohere.

Response Time

We measured the time taken by each model to solve the puzzles. The results are as follows:

TABLE IV. RESPONSE TIME TABLE

Model	Response time with prompting
Cohere	1.15 seconds
Gemini	1.84 seconds
Llama2.7	1.97 seconds
Llama3	1.89 seconds
Llama3.2 instruct	1.72 seconds

Table IV highlights the response times for different models with prompting vary, with Cohere being the fastest at 1.15 seconds. Gemini follows closely with a response time of 1.84 seconds. The Llama models exhibit slightly longer response times, with Llama2.7 at 1.97 seconds and Llama3 at 1.89 seconds. Meanwhile, Llama3.2 Instruct provides a relatively quicker response at 1.72 seconds, making it faster than the other Llama versions but still slower than Cohere.

Errors Rate

We also measured the number of errors, such as incorrect or incomplete answers.

TABLE V. ERROR RATE TABLE

Model	With Prompting Error%
Cohere	5%
Gemini	10%
Llama2.7	20%
Llama3 instruct	15%
Llama3.2 instruct	10%

The error percentages with prompting for different models are as follows: Cohere demonstrates the lowest error rate at 5%. Gemini has a higher error rate of 10%, which is the same as Llama3.2 Instruct. Llama3 Instruct has a slightly higher error rate at 15%. The model with the highest error percentage is Llama2.7, with an error rate of 20%.

Discussion:

The findings of this study underscore the transformative potential of structured prompting in enhancing the performance of large language models (LLMs) across multiple key metrics. Notably, Cohere emerged as the top-performing model, achieving an impressive accuracy of 95% and the fastest task completion time of 23.06 seconds. Its superior response time of 1.15 seconds and lowest error rate of 5% further highlight the model's robustness when guided by well-structured prompts.

Gemini and Llama3.2 Instruct demonstrated comparable strengths, with accuracies of 90% and error rates of 10%, marking significant improvements from baseline performance. While Gemini displayed competitive processing efficiency, Llama3.2 Instruct stood out for its remarkable efficiency gains, reducing task completion times to 34.56 seconds—a testament to the adaptability of structured prompting in streamlining computational processes.

Interestingly, Llama2.7 lagged behind its counterparts, showing the least improvement across all evaluated dimensions. This discrepancy suggests that the model may require further architectural enhancements or tailored prompting techniques to fully leverage its capabilities. The relatively modest gains of Llama2.7 highlight the need for model-specific strategies when employing structured prompting, as a one-size-fits-all approach may not yield optimal outcomes across diverse architectures. Overall, these results reaffirm the critical role of structured guidance in refining LLM performance.

Chapter 5 Impacts of the Project

5.1 Impact of this project on societal, health, safety, legal and cultural issues

This project enhances societal development by improving LLM capabilities in solving puzzles, which can support education, cognitive stimulation, and interactive learning tools. While it has no direct health impact, it can aid cognitive health by fostering mental engagement. Improved AI problem-solving contributes to safety-critical applications, reducing risks in areas like autonomous systems. Legally, it raises concerns about AI-generated content ownership, emphasizing the need for proper frameworks. Culturally, it promotes accessibility in education and entertainment while fostering inclusivity, though over-reliance on AI may affect traditional learning approaches.

5.2 Impact of this project on environment and sustainability

As an AI-based project, this work has minimal direct environmental impact due to the absence of physical hardware components. However, it contributes to sustainability by advancing the efficiency and capabilities of large language models (LLMs). Enhanced AI systems can be applied to various fields, such as optimizing energy management, reducing resource wastage, and automating processes in an environmentally friendly manner. Additionally, the project promotes computational efficiency, minimizing energy consumption during model usage and fostering the development of greener AI technologies.

Chapter 6 Project Planning and Budget

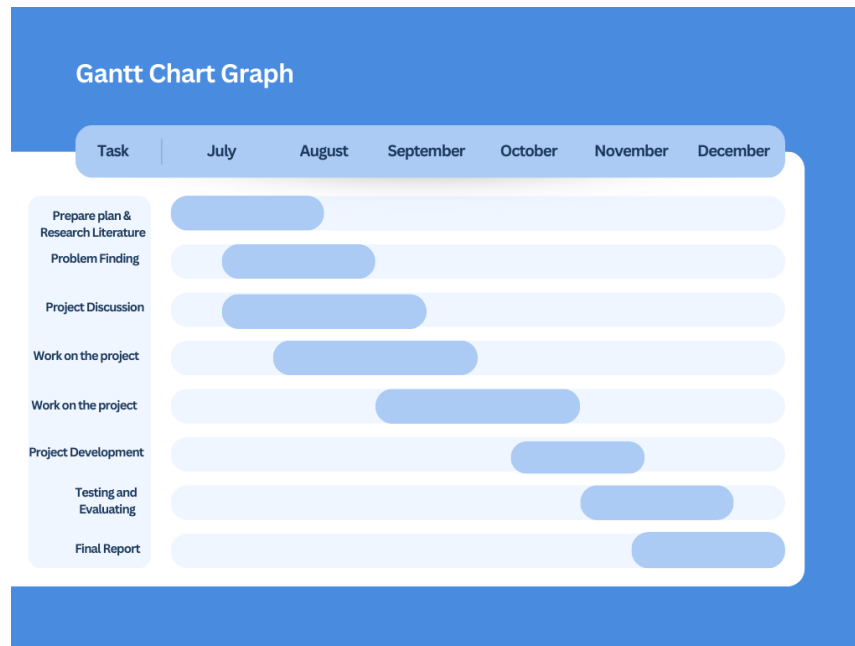


Figure 4: Gantt Chart.

The Gantt chart visualizes the project timeline from July to December. It breaks down the project into various tasks, including planning, literature research, problem finding, project discussion, development, testing, evaluation, and final report preparation. Each task is represented by a horizontal bar, with its length indicating the duration. The chart shows that the project starts in July with planning and literature research, followed by problem finding and discussion. Development begins in September and continues into October, overlapping with testing and evaluation. Finally, the project concludes with the final report preparation in December.

Budget:

As this project is AI-based and primarily software-driven, no hardware components were involved, significantly reducing costs. The budget included expenses for computational resources, tools, and software libraries.

Budget Breakdown:

1. **Computational Resources** (Cloud computing or local GPUs): \$50
2. **Software Libraries** (Open-source, minimal costs for premium tools): \$30
3. **Miscellaneous** (Documentation, internet, and other resources): \$20

Total Project Cost: \$100

This cost-efficient budget highlights the advantage of using AI technologies without hardware dependencies.

Chapter 7 Complex Engineering Problems and Activities

7.1 Complex Engineering Problems (CEP)

TABLE VI. COMPLEX ENGINEERING PROBLEM ATTRIBUTES TABLE

Attributes		Addressing the complex engineering problems (P) in the project
P1	Depth of knowledge required (K3-K8)	Requires expertise in AI, machine learning, NLP, data analysis, and LLM tuning (K3-K8).
P2	Range of conflicting requirements	Balancing model accuracy with processing efficiency and computational overhead.
P3	Depth of analysis required	Detailed analysis in prompt selection, model evaluation, and fine-tuning based on performance metrics.
P4	Familiarity of issues	Familiarity with LLM operation, prompt engineering, and model evaluation
P5	Extent of applicable codes	No predefined standards; relies on best practices for LLM evaluation.
P6	Extent of stakeholder involvement	Involves AI researchers, project team, and users for dataset integration and model evaluation.
P7	Interdependence	Requires integration of LLMs, prompt engineering, and dataset evaluation for effective results.

7.2 Complex Engineering Activities (CEA)

TABLE VII. COMPLEX ENGINEERING PROBLEM ACTIVITIES TABLE

Attributes		Addressing the complex engineering activities (A) in the project
A1	Range of resources	Requires human resources with AI and machine learning expertise, computational resources, and software tools.
A2	Level of interactions	Involves collaboration with team members, AI experts, and dataset users.
A4	Consequences to society / Environment	Contributes to AI advancements and improved problem-solving efficiency in various applications.
A5	Familiarity	Requires knowledge of AI tools, LLMs, and evaluation metrics, supporting SDGs.

Chapter 8 Conclusions

8.1 Summary

This project focuses on evaluating the problem-solving capabilities of large language models (LLMs) in solving brain teaser problems. By utilizing techniques like few-shot learning and model fine-tuning, the project aims to maximize the accuracy and efficiency of LLMs in solving a wide range of complex problems. The results will contribute to better understanding the potential and limitations of LLMs in real-world problem-solving scenarios, enhancing their applications in various fields such as education, AI-driven solutions, and cognitive research.

8.2 Limitations

1. **Data Constraints:** The project is limited by the availability of diverse brain teaser datasets, which can impact the model's performance in solving a wide range of problems.
2. **Model Dependence:** The language model used relies on pre-existing training, and while it can solve many problems, it may not perform optimally for all types of brain teasers.
3. **Computational Resources:** The performance of large language models can be computationally expensive, limiting the speed and scalability of the project, especially with larger problem set.

8.3 Future Improvement

To expand the scope of this project, we plan to integrate **computer vision** capabilities. This will enable our system to solve visual puzzles by processing and interpreting images, allowing the bot to handle challenges that require visual recognition and analysis. By incorporating advanced image processing techniques and deep learning models, we aim to enhance the bot's ability to solve not only text-based but also visual-based brain teasers, making it more versatile and intelligent in problem-solving.



Figure 5: Future Consideration

References

1. P. Giadikiaroglou, M. Lymperaiou, G. Filandrianos, and G. Stamou, "Puzzle Solving using Reasoning of Large Language Models: A Survey," *Artificial Intelligence and Learning Systems Laboratory*, School of Electrical and Computer Engineering, National Technical University of Athens. [Online]. Available: panosgiadi@gmail.com, {marialymp, geofila}@islab.ntua.gr, gstam@cs.ntua.gr
2. Y. Li, H. Wang, and C. Zhang, "Assessing Logical Puzzle Solving in Large Language Models: Insights from a Minesweeper Case Study," *Georgia Institute of Technology*, Atlanta, USA. [Online]. Available: {yinghaoli, hwang984, [chaozhang](mailto:chaozhang@gatech.edu)}@gatech.edu