SUTD SHARP Term 3 Honours sessions Large Language Models Homework

Goh Jet Wei 1007655

Tasks:

1. ***Load a small model on Google collab -- Qwen2.5-1B***

   Code provided in the Google Collab Jupyter Notebook file.

2. ***Load GSM8K dataset and leverage the loaded LLM to solve the mathematical test problems.***

   Code and output of solving the test problems provided in the Google Collab Jupyter Notebook file. I tested the Qwen2.5-1B LLM model with the first 5 questions and answers in the GSM8K dataset.

3. ***Find the common problems in the solutions.***

- Misinterpretation of Problem:
  - In several instances, the model misinterpreted key details or misunderstood the structure of the question, leading to incorrect computations or logic.
  - For example, in Janet's egg problem, the model assumed Janet had 48 eggs per day rather than what was said in the question of 16 eggs per day.
- Arithmetic and Calculation Errors:
  - The model occasionally made arithmetic mistakes.
  - For example, in the profit calculation for the house-flipping question, it used an incorrect formula of '130000 - 80000 - 50000 = $60,000', when the answer should be '$0' instead
- Incorrect Formula Usage or Steps:
  - Some solutions displayed flawed logic or incorrect formula usage.
  - For example, in the profit calculation for the house-flipping question, it says "The house is now worth 80000 + 50000 = $130,000. The value has gone up by 130000 * 0.15 = $19,000." However, the question says the

value of the house has gone up by 150%, and hence in reality the multiplied value should be 1.5, instead of 0.15.

4. *Propose techniques that can solve the problems. Provide examples.*

- Enhanced Parsing for Better Understanding of Context:
    - Use natural language processing (NLP) techniques such as Dependency Parsing to improve the model's ability to parse and isolate specific keywords or phrases that define the problem's context and requirements better.
    - Example: For Janet's egg problem, highlight terms like "per day" and ensure daily operations (e.g., eating 3 eggs, baking with 4) are applied correctly to the daily count of 16 eggs using better NLP techniques
- Basic Arithmetic Verification and Formula Application Checks:
    - Apply a post-processing layer to validate basic arithmetic and verify formula use according to expected steps. Checking against predefined problem patterns or using backtracking algorithms such as Constraint Satisfaction Backtracking could help confirm accuracy.
    - Example: For the house-flipping problem, ensure calculations follow the sequence of determining total costs, then applying profit percentage correctly before subtracting costs.
- Problem Template Matching:
    - Use template matching, such as pattern recognition and keyword detection algorithms such as RegEx and N-grams to categorize problems into types (e.g., profit calculation, rate problems, growth problems). Each identified category can have specific predefined solution patterns or steps and hence be used, to avoid unnecessary or incorrect operations.
    - Example: In questions about exponential growth, detect phrases like "doubles every hour" to apply simple exponential formulas and clarify steps, avoiding unnecessary logarithmic functions unless specified.
    - Example:
        - Profit Calculation Problems: Identified via phrases like "buy," "sell," "profit margin," "mark-up," and "total cost."
        - Rate Problems: Identified via phrases like "rate," "per hour," "per day," "miles per hour."
        - Growth Problems: Identified via phrases like "grows by," "increases," "doubles every," or "exponential."