Assignment 3: Tree-of-Thought

Due: 2024-11-07 15:29 PM

Tasks

1. Apply Tree-of-Thoughts for Sudoku Puzzle
2. Run Test Set and Get Result: **10 pts**

Total 10 pts

Python Notebook for Assignment 3

- Download the Python Notebook from the eTL.
  - “**PE2024F_Assignment_3.ipynb**” file
- Or you can download the Notebook file from the Google Drive
  - https://drive.google.com/drive/folders/1UzWGhqCEcD7hY5BbMD8eYKTMBjt2yCdJ?usp=sharing

1. Apply Tree-of-Thoughts for Sudoku

- For given skeleton codes, you should **design the propose prompt** and **design the Class for structured output**.
- But you can modify the other part of code freely to fit your prompt input or output.

TODO: Sudoku Solver with Tree-of-Thoughts

Define propose prompt for the next thought

```
[ ]  propose_prompt = PromptTemplate(
        ##############################
        #########    TODO   ##########
        ##############################
    )
```

Define class for the structured output

```
[ ]  class Guess(BaseModel):
        ##############################
        #########    TODO   ##########
        ##############################
```

2. Run Test Set and Get Result:

- **Run the test code** at the below part of the Notebook.
- Your score will be calculated as below:
  - score = {solve_ratio} * 20
- You will get **10 points** if the solve_ratio is larger than 0.5
- If your solve_ratio is the highest, then you will get a chance for presentation
  - If there are students with same solve_ratio, then compare the average loops

```
sudoku_puzzle_test = [
    "[[3, 4, '*', '*'], [1, '*', '*', '*'], ['*', '*', 2, 1], [2, '*', '*', '*']]",
    "[[3, '*', '*', 2], ['*', 1, 4, 3], ['*', 2, 3, 1], [1, '*', 2, 4]]",
    "[['*', '*', '*', 4], ['*', 1, '*', '*'], ['*', '*', '*', 3], ['*', 4, 2, 1]]",
    "[[1, '*', 2, '*'], [2, '*', '*', 3], [3, '*', 4, 2], [4, '*', 3, '*']]",
    "[['*', '*', '*', 4], ['*', 2, '*', '*'], [3, 4, 1, '*'], [2, 1, '*', '*']]",
    "[[2, '*', '*', 4], [1, 4, '*', 3], [3, 1, '*', '*'], ['*', 2, 3, 1]]",
    "[[2, '*', 3, '*'], [3, '*', 2, 1], [1, '*', '*', 2], [4, '*', '*', '*']]",
    "[['*', '*', 1, 3], [1, '*', '*', '*'], ['*', '*', '*', 1], [3, 1, '*', 4]]",
    "[['*', '*', '*', 4], [1, 4, 2, 3], [3, 1, 4, '*'], [4, 2, '*', '*']]",
    "[['*', 1, 2, 4], [2, '*', 1, '*'], ['*', '*', 3, '*'], ['*', 3, '*', '*']]"
]
```

```
# get solve ratio for the test puzzle set

solve_count = 0
total_count = len(sudoku_puzzle_test)
total_loops = 0

for puzzle in sudoku_puzzle_test:
  answer, solved, loops = solve_sudoku_tot(puzzle)
  if solved:
    if is_valid_sudoku(answer):
      solve_count += 1
      total_loops += loops

solve_ratio = solve_count / total_count if total_count > 0 else 0
average_loops = total_loops / solve_count if solve_count > 0 else 0
print(f"Solve ratio for the test puzzle set: {solve_ratio}")
print(f"Average loops for solving: {average_loops}")
```

Test code - it takes few minutes

Submission

- Due: 11/07 (Thu), 15:29 PM (right before class)
- After you have completed the assignment, you need to submit your **Python Notebook File or Python file as a .ipynb or .py** via eTL assignment submission
  - Ex) 2024-00000.ipynb or 2024-00000.py
  - If you cannot upload the file, you can also upload compressed .zip file.
- We will calculate deducted score according to the following equation:
  - (Original_score) × (1 − 0.1 × ceil(delayed_days))
  - After 5 days (120 hours), there is no score.
  - Ex1) 50 hours late: 70% of the original score
  - Ex2) 115 hours late: 50% of the original score
  - Ex3) 120 hours late: 0% of the original score
- Concept level discussion is encouraged, but discussion of code/prompt directly related to assignments is not allowed. The assignments must be students' own work.