Unmasking the giant: A comprehensive evaluation of ChatGPT's proficiency in coding algorithms and data structures

Sayed Erfan Arefin, Tasnia Ashrafi Heya, Hasan Al-Qudah, Ynes Ineza, Abdul Serwadda

Objectives

- Evaluating ChatGPT's proficiency in coding problem-solving
- Code quality Evaluatoion
- Incase of errors: Understanding the nature of errors
- Examines both GPT-3.5 and GPT-4 models
- Investigating potential data memorization during ChatGPT's training
- **Topics and Subtopics:**
  - Algorithms;
    - Dynamic programming
    - Greedy algorithms
    - Depth first search
    - Divide and conquer
    - Topological sort
  - Data structures
    - Priority queue
    - Array
    - Hash tables
    - Binary Search Tree
    - Stack
  - Strings problems

Methods

1. Tools used

- **LeetCode:** Online coding challenge platform
- **Pylint:** Checks adherence to coding standards

2. Data collection and processing

- LeetCode challenges entered into ChatGPT prompt
- ChatGPT generated code is submitted to LeetCode
- Submission Recording:
  - Success, Human success rate, Error messages (If any)
- Assessed code quality and reported problem type
  - Errors, Warnings, Refactors and Conventions

3. Selection of coding problems

- Complete Coding Challenges: 723 (1446 for both GPT models)
- Incomplete Coding Challenges: 673 (1346 for both GPT models)

4. Experiment configurations

- Investigating ChatGPT's recall ability or inferencing capacity with missing information.
  - **Complete Challenges** (Includes example, restrictions etc.)
  - **Incomplete Challenges** (Missing example, restrictions etc.)
- Investigating ChatGPT's memorization of problems and solutions.
  - Public data till **September 2021** used for ChatGPT training
  - **Train Set**: LeetCode problems before September 2021
  - **Test Set**: LeetCode problems after September 2021
- Incase of wrong answers: How wrong were ChatGPT's wrong solutions?

Results

1. How accurate is ChatGPT?

1.1. Complete Coding challenges

- For train set challenges,
  - GPT-4 is notably superior
  - GPT-3's performance comparable to that of humans
- For test set challenges,
  - Humans surpass both models
  - GPT-4's success rate is roughly double that of GPT-3

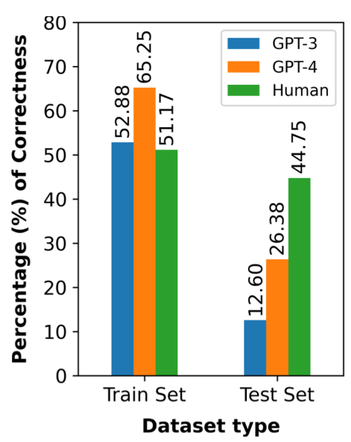
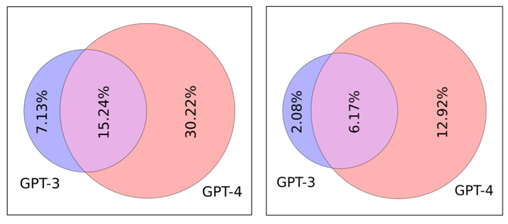


Fig 1: Correctness of GPT-3, GPT-4 & Humans for Train and Test sets.

1.1.1. GPT-4 vs GPT-3:



Train set Test set

Fig 3: Exclusive and inclusive correctness of GPT-3 and GPT-4 for all the problems in the train and test datasets

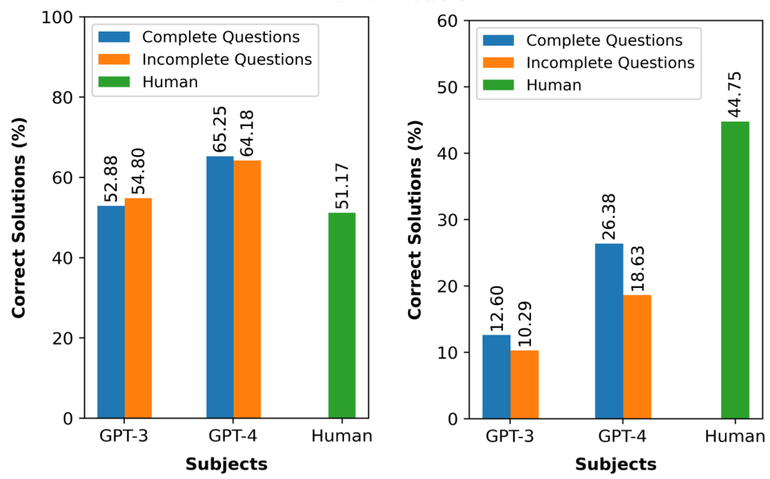
- Specific questions where GPT-3 succeeds but GPT-4 fails
  - GPT-3 exclusively solves 7.13% problems in the train set
  - GPT-3 exclusively solves 2.08% in the test set

Topic	No. of Questions (%)		Sub-topic	No. of Questions (%)	
	Complete coding challenges	Incomplete coding challenges		Complete coding challenges	Incomplete coding challenges
Algorithm	422 (58.40%)	407 (60.48%)	Dynamic	132 (31.30%)	124 (30.47%)
			Greedy	136 (32.23%)	129 (31.70%)
			Depth first search	99 (23.46%)	99 (24.32%)
			Divide and conquer	33 (7.82%)	33 (8.11%)
			Topological sort	22 (5.21%)	22 (5.41 %)
Data Structure	248 (34.30%)	228 (33.88%)	Priority queue	82 (33.06%)	82 (35.96 %)
			Array	49 (19.76%)	45 (19.74 %)
			Hash table	43 (16.94%)	42 (18.42 %)
			Stack	38 (15.73%)	33 (14.47 %)
			Binary Search Tree	36 (14.52%)	26 (11.40 %)
Strings	53 (7.30%)	38 (5.65%)			
Total Questions	723	673			

Table 1: Percentage of LeetCode questions of different topics compared to the total no. and percentage of question no. of sub-topics compared to the topics they belong to in the dataset.

1.2. Incomplete Coding Challenges

- Both models show similar performance levels complete or incomplete questions
- Humans, with access to complete information on all challenges, performed worse than both GPT models, which worked with incomplete information
- In test set, both GPT-3 and GPT-4 show a significant reduction in correct solutions compared to the training set
- Overall behavior observed in the pattern of correctness could be a combination of,
  - Some memorization
  - The robustness of the GPT models



(a)Train set (b)Test set

Fig 2: Performance of GPT models on the incomplete coding challenges

2. Cases that failed to produce a correct solution

- Incorrect solutions can be:
  - Run Time Error
  - Incorrect response
- **In terms of incorrect solutions (excluding runtime error),**
- **Training set:**
  - 60% to 77% of cases
- **Test set:**
  - Occurred 80% of the time or more

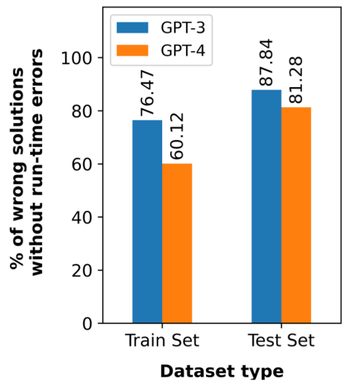


Fig 4: Percentage of all errors excluding runtime errors

3. How wrong were ChatGPT's wrong solutions?

- LeetCode provides the number of passed test cases
  - Fraction of test cases passed
  - As a measure of how wrong a solution is
- Incorrect solutions often passed a very low percentage of test cases

4. Notable problems based on PyLint Report

- **Error E0602:** A variable that was not defined is accessed
- **Warning W0621:** When one redefines a name from an outer scope
- **Refractor R0903:** A small number of public methods
- **Convention C0103:** Does not adhere to the naming conventions specific to its type (Variable, function name etc.)

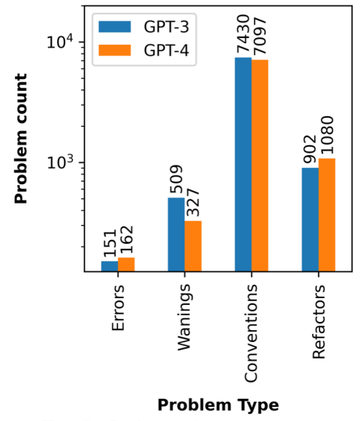


Fig 5: Code quality issues seen in ChatGPT solutions