

## Models Implemented

5 fine-tuning models with curie GPT-3

1. Describe a location (I've given you this code. You can adapt it for other models)
  - inputs: category, location name
  - output: location description
2. List the items that are at a location
  - inputs: category, location name, location description, number of items
  - output: list of item names
3. Describe an item
  - inputs: category, location name, location description, item name
  - output: item description
4. List connections from the current location
  - inputs: category, location name, location description, and optionally a partial list of existing connections (direction, location name) tuples
  - output: a list of (direction, location name) tuples
5. Get an item's properties
  - inputs: item name, item description, property (e.g. gettable)
  - output: True or False if the item has that property

This is to help us generate locations, items, descriptions, properties with given conditions.

## Game Built

I have built pure-AI-generated game categories, locations, items, connections. The logic is clear. We try to generate some items inside the building. Then, we keep looking for connections to the current location. We use DFS to implement that. After 10 rounds, there is a lot of information generated, so we can stop. You can also set a different "round value" or wait for the program to converge in the end. In my building logic, those locations should belong to the same category, so it would be successful working on small spots, like houses, apartments, campuses, etc.

## Evaluation

This is to evaluate how good our models are. I evaluated the property model (the last one) to check whether our model accuracy is good enough. The result turns out that it is not good, with only 55.18%. If using Davinci, the model is expected to be largely improved. However, I do not have much money left in openAI lol.

## AI Feasibility and Future Improvement

Indeed, it is feasible to generate all spots for the game with AI, but lacks some kind of interesting or logical components. Since the fine-tuning method implemented here mostly copes with one specific task in our game-building process, it cannot effectively have a good effect on some hidden logic related to task-targeted models. For example, locations could not be proved to be reasonably generated until there are many surrounding buildings known. However, when we construct the game location, we only considered the nearby one (only one). Hence, we should have some more training models inside the build\_game functions wrapped with the above 5 models.