CIS700 HW2

Zhihua Zhang - zhihuaz@seas.upenn.edu
Zach Li - zachli@seas.upenn.edu

January 31, 2022

1 What we have done

1.1 Model Fine-Tuning

Following the provided code to create fine-tune training data for location description given category and location name, we fine-tuned GPT3 (Curie or Babbage) for the following tasks:

- Generate list of item names (making distinction between in_objects and ex_objects) at a location given {category, location name, location description, number of items}
- Generate description for an item given {category, location name, location description, item name}
- Generate a list of connections (direction, destination) for a location given {category, location name, location description}
- Predict whether an item has a property given {item name, item description, property (e.g. is_drink)}

1.2 Game Generation

Since we decided to fine-tune models for each building block that constructs the game in subsection(1.1), we can now create a fully AI-generated text adventure game given an input category.

We performed the following steps to generate the game

(a) Initialize the game environment and specify the input category, which is the default ”Dark Forest” in our case;

(b) Call the fine-tuned model to generate a list of location (room) names with category as prompt input;

(c) For each room, sequentially produce and add the following to the prompts for the fine-tuned models: [1] location description and [2] in_item and ex_item names;

(d) For each item in the above step, sequentially obtain and include the following in the prompts for the fine-tuned models: [1] in_item and ex_item names descriptions [2] in_item and ex_item properties;

(e) When a room is complete, call the corresponding fine-tuned model to generate neighbors, which are a number of (direction, destination name) tuples.

(f) Because each neighbor (destination) is a new location (room), repeat step (b) - (e) until no new neighbors are produced or all of them are already included in the game.

1.3 Model Evaluation

Using the GPT3 Babbage model fine-tuned for item property prediction, we calculated property-wise recall and precision (by counting true positives, false positives, and false negatives against gold standard) to evaluate its performance. We obtained the following results:

Property	**Recall**	**Precision**
Gettable	0.531	0.863
Weapon	0.524	0.815
Surface	0.408	0.556
Container	0.406	0.500
Wearable	0.667	0.667
Drink	0.364	0.571
Food	0.571	0.727
Plural	0.830	0.865
Overall	**0.584**	**0.766**

2 Future work for fully generative text adventure games

We think it is possible to create high-quality text adventure games entirely by AI. However, we are not ready to go that far at the level of this assignment. For example, there are times that models produce nonsensical results, even though the game still playable.

(a) Category: The category “dark forest” is given in our case, which is not AI generated. But we can provide context information like “generate location settings for an adventure game” as prompt to achieve the goal.

(b) Connections: We experimented with two approaches to fine-tune our model (1) use all rooms in dataset including ones with empty connections and (2) only use rooms with valid connections. However, there are problems in both cases. When we kept all training samples in the first case, the model often fails to produce any connection output, likely due to class imbalance. When only using rooms with non-empty neighbors, the fine-tuned model can generate connections that might not make sense for a particular location. For example, we should not move forward from a cliff. These problems should be mitigated by re-sampling the training data so the positive and negative samples are close to a one-to-one ratio.

(c) When we played around with our fine-tuned models for different tasks, some generations does not make perfect sense. For example, an item automatically generated in location “The king’s library” has a name of “Large” with a description of “the large object on the ground is a doorway. it is difficult to see as the light shines through it”. Training a more powerful model such as Davinci could potentially address this issue, but they incur a high cost for fine-tuning and we have limited resources.