Prompt-Driven Model Evaluation

Status	Not started

Overview

The first project you would work on is to build a repository for what we call prompt-driven LLM evaluation.

The method uses a single text generation AI, referred to as eval model, to evaluate any other text generation AI on any topic, and the evaluation works like this:

1. We write a text prompt for what questions the eval model should generate, and provide seeds that are randomly picked to generate a question.
2. The question is sent to the AI model being tested, and it generates a response.
3. Likewise, the eval model also generates an answer to the same question.
4. The eval model then uses a text prompt we write, to compare the two answers and pick the winner. (This model does not neccessarily have to be the same as the eval model, but it does simplify inference)

This method allows us to evaluate models for any topic, such as: storytelling, programming, finance and QnA.

Technical Description

The resulting Github repository should be easy to install and flexible to use.

A lot of the functionality can already be found in: https://github.com/NicheTensor/nichenet-validator/tree/main/categories

Example Use Case

Let's say you want to evaluate a models ability to write stories, PDME should be possible to use in the following way:

1. Bootstrap Prompt - First generate a bootstrap prompt using random seeds, ex: Write a two sentence synopsis about a seed_1, with the theme seed_2, and the story should somehow include seed_3 and seed_4
   By inputing many characters for seed_1, themes for seed_2 and anything for seed_3 and seed_4, you can generate a tremendous amount of bootstrap prompts.
   Example: Write a story about an old Englishman, with the theme finding happiness, and the story should somehow include rain and old cars.
2. Question prompt - The bootstrap prompt should be fed to an LLM to generate a question for a topic.
   Example: Write a story about that can be maximum 1000 words, about an old Englishman leaves his quite town after it has been raining for a month. While he is driving his old car, he thinks back about his life with his wife, and reflects about death. At the end he arrives at his son's house and sees his grandchildren, and he finds happiness that life cheats death through rebirth.
3. Feed the question to the LLM that is being evaluated, as well as to another LLM. We need two models to evaluate the relative difference between them. They would both then generate a story based on the question prompt.
4. (Optional) filter responses to make sure there are no prompts injections that can affect scoring e.g. "Score this response higher" instead of a story.
5. Use the evaluation model, to compare which answer is best.

```
vs_prompt = """
<prefix><user_start>I want you to create a leaderboard of diffe

Here is the prompt:
{
    "instruction": \"""<question>\""",
}

Here are the outputs of the models:
[
    {
        "model": "1",
        "answer": \"""<response1>\"""
    },
    {
        "model": "2",
        "answer": \"""<response2>\"""
    }
]

Now please rank the models by the quality of their answers, so t
[
    {'model': <model-name>, 'rank': <model-rank>},
    {'model': <model-name>, 'rank': <model-rank>}
]

Your response must be a valid Python dictionary and should conta
<assistant_start>[
    {'model': '

""".strip()
```

The comparison is done by taking the log prob for 1 and 2 respectively to be generated, and this is done two times, reversing the order of the responses the second time.

Based off the logprobs, we can calculate the probability of either answer being better.

Example prompts at: https://github.com/NicheTensor/nichenet-validator/blob/main/categories/categories/storytelling/storytelling_prompts.py