

Junior AI Engineer Interview Task

Task description:

Create an agentic system that can answer users' questions about certain situations in a board game.

The board game rulebook can be found here:

<https://cdn.1j1ju.com/medias/8d/c5/21-gloomhaven-rulebook.pdf>

I.

Implement a RAG-based solution (it can be basic or more advanced — it is up to you) so that the agent can answer questions based on the rulebook.

The user will provide a situation about which they are unsure whether they played it according to the rules. The agent will provide a structured answer containing an explanation based on the rules, information on whether the user handled the situation correctly (basically a boolean) and a category label indicating which aspect of the rulebook was relevant: [BoardGameSetup, Combat, Scenario, Character].

II.

Give the agent the ability to browse the internet. If the agent cannot answer the question based on the rulebook, it should search for the answer online.

III.

Evaluate the accuracy of the Agent in terms of how accurately it can predict if a scenario was played in the right way. Create three examples of questions and answers. Then, based on these three use an LLM to create a synthetic dataset of 15 question-answer pairs. Use this during evaluation.

Tools and code

The final solution should be in .py files. Use the notebook to showcase its usage.

Structure the components into appropriate classes, maintain a logical file structure, and write clean code.

You can use any type of LLM and embedding model to solve the task; however, the notebook contains open-source models that can be run on Google Colab. Due to size

restrictions, these models have limited ability; the accuracy of the final solution is irrelevant in case of any LLM used.

To reduce embedding time, you can modify the PDF so that it only contains a few pages.

If there is a problem with the agent libraries using async that results in an exception in Collab, which would take time to debug, ignore it. The functionality should be coded to work in a local Python environment. However, if it doesn't work in Collab due to this kinds of bug, that is not a problem.

```
%pip install llama-index-embeddings-huggingface llama-index-llms-huggingface ai
```

```
from llama_index.embeddings.huggingface import HuggingFaceEmbedding
from llama_index.llms.huggingface import HuggingFaceLLM
from llama_index.core import set_global_tokenizer
from transformers import AutoTokenizer
```

```
embed_model = HuggingFaceEmbedding(model_name="Qwen/Qwen3-Embedding-0.6B")
embeddings = embed_model.get_text_embedding("Hello World!")
print(len(embeddings))
print(embeddings[:5])
```

```
model_name = "Qwen/Qwen3-1.7B"
locally_run = HuggingFaceLLM(model_name=model_name, tokenizer_name=model_name)
```

```
locally_run.complete("Hello world").text
```

