Alberto Andrés Valdés González.
Degree: Mathematical Engineer.
Work position: ML-Engineer.
Mail: anvaldes@uc.cl/alberto.valdes.gonzalez.96@gmail.com
Location: Santiago, Chile.

Parsing LLM

Large language models (LLMs) are highly versatile tools that perform various tasks, including language translation, question answering, and code generation. Chat tools like ChatGPT, Claude, and Gemini have significantly improved conversational AI by enabling users to engage with these tools in a manner that feels as natural as conversing with another human. The use of these LLMs can be further improved by building assistant-like tools where users can instruct in natural language to complete certain tasks. This would significantly enhance productivity and efficiency, making these tools invaluable.

However, integrating LLM output with other tools has its challenges. Even though LLMs are powerful tools capable of many things, general LLMs are often trained to respond conversationally. Due to the conversational nature of the responses, it is hard to get a consistent response from an LLM.

Consider the following scenario. A restaurant's user reviews must be analyzed. Information from the review must be extracted and fed into an analytics engine to further analyze and visualize the data.

If we use ChatGPT to do this, we could write a prompt like the one below.

Extract the following information from the provided restaurant review:

Determine the overall sentiment of the review (positive, negative, or neutral). Assign a score from 1 to 10, indicating how likely the reviewer would recommend the restaurant to their friends and family based on the content of the review. Identify and extract the key themes or aspects discussed in the review.

Review:

Burgers and Bites offers a mixed experience. The staff is exceptionally amicable, and the food is equally tasteful but time-consuming. Unfortunately, one will find a limited variety of vegetarian dishes and no vegan options. Parking can be tricky, especially during peak hours. Nevertheless, the quality of the burgers compensates for these shortcomings—literally.

Output:

1. Overall sentiment of the review: Positive.

2. Score (1 to 10, likelihood of recommendation): 7

3. Key themes the review talks about:

a. Staff friendliness: "The staff is exceptionally amicable"

b. Food quality: "the food is equally tasteful"

c. Service time: "time-consuming"

d. Vegetarian and vegan options: "limited variety of vegetarian dishes and no vegan options"

e. Parking difficulties: "The parking is tricky, particularly if the store is bustling with customers"

f. Burger quality: "the quality of the burgers makes up for it"

Comments:

Even though this is a well-formatted response, parsing this programmatically will be difficult. Also, even when the same review is passed to the ***LLM*** *multiple times, it will generate slightly different outputs each time, as LLMs are* ***generative****, not* ***deterministic****.*

How can we solve it?

```
from langchain_core.output_parsers import JsonOutputParser
from langchain_core.prompts import PromptTemplate
from langchain_core.pydantic_v1 import BaseModel
from langchain_openai import ChatOpenAI

class Review(BaseModel):
    sentiment: str
    score: int
    themes: list[str]

parser = JsonOutputParser(pydantic_object=Review)

prompt_template = PromptTemplate(
    template="""Extract the following information from the provided restaurant review:
```

```
    Determine the overall sentiment of the review (positive, negative, or neutral).
    Assign a score from 1 to 10, indicating how likely the reviewer would recommend the res
    Identify and extract the key themes or aspects discussed in the review.
    n{format_instructions}nn
    Review: {review}n""",
    input_variables=["review"],
    partial_variables={"format_instructions": parser.get_format_instructions()},
)

llm = ChatOpenAI(model="gpt-3.5-turbo", temperature=0)
chain = prompt_template | llm | parser
print(chain.invoke({"review": <Review to be analyzed>}))
```

Output:

```
    {
    'sentiment': 'positive',
    'score': 8,
    'themes':
        ['Staff friendliness',
        'Food quality',
        'Limited vegetarian options',
        'No vegan options',
        'Parking challenges during peak hours']
    }
```
