Few-Shot Prompting with User and Assistant Pairs

The API allows us to supply example user and assistant pairs with the messages list to perform a more advanced form of few-shot prompting. Each user message is paired with an assistant message to give the AI model a discrete example of what its generated output should look like based on the specified input.

Let’s walk through an example:

```
let fewShotMessages = [
  {
    role: "system",
    content: "You are a friendly travel guide excited to help users travel the Caribbean. Your responses should only include destinations that are in the Caribbean"
  },
  {
    role: "user",
    content: "Suggest a destination suitable for a family with toddlers."
  },
  {
    role: "assistant",
    content: "Sure! Consider visiting Aruba for a family vacation. It offers beautiful beaches, family-friendly resorts, and attractions like the Butterfly Farm and Arikok National Park that kids would enjoy."
  },
...
```

The above example stores messages in an array, few_shot_messages. The user/assistant messages provide the model of how it might respond to inquiries to Caribbean destinations. The prompting provides details like the country name, points of interest and activities, and locations worth visiting.

A second set of user/assistant messages can be provided to encourage the model that it’s acceptable to address user prompts requesting destinations outside of the Caribbean.

```
...
  {
    role: "user",
    content: "We're looking for a Mediterranean destination that has beaches. Any suggestions?"},
  {
    role: "assistant",
    content: "While I specialize in Caribbean destinations, I am familiar with a few Mediterranean destinations that have great beaches. Majorca for instance is right off the coast of Spain and has beautiful, pristine beaches."
  }
...
```

Now, equipped with this additional context, the model can better respond when faced with a user prompt for a destination outside of the Caribbean.

```
...
  {
    role: "user",
    content: "Our group loves water activities. What is a good European destination?"
  }
];
```

With the few-shot prompting, we receive the following:

Output (Model Response):

If you're looking for a European destination with a wide range of water activities, I recommend the Algarve region in Portugal. With its stunning coastline, the Algarve offers excellent conditions for surfing, paddleboarding, kayaking, and boat excursions. Whether you're a beginner or an experienced water sports enthusiast, there's something for everyone in the Algarve.

The model successfully suggested a European destination with great water activities, showcasing its ability to adapt and respond beyond Caribbean locations.

Note that while providing additional user and assistant pairs will improve the accuracy and performance of the model’s response, it does increase the number of tokens used within the prompt, which, consequently, affects both the token usage and the cost of the API call.

Instructions

1. Checkpoint 1 Passed

1.

You are the quality manager of an online clothing store and handle the reviews from customers. Use few-shot prompting to classify customer service reviews as either “Positive” or “Negative”.

To use the API for this classification task, start by preparing an array of user/assistant prompts that provide review and classification examples. To do this:

- Add user and assistant objects to the empty fewShotMessages array.
- The first object in fewShotMessages should have the role of "user" and the content should be an example of a positive review example, such as:
    - “Clothing fit perfectly and was true to size.”
    - “Product showed up on time and undamaged”
- The second object in fewShotMessages should have the role of "assistant" and should be the reply, "Positive"

⍰ Use the following syntax:

```
let ARRAY_VARIABLE = [
 {
  role: "user",
  content: REVIEW_EXAMPLE
 },
 {
  role: "assistant",
  content: REVIEW_CLASSIFICATION
 },
]
```

⍰ Checkpoint 2 Passed

2.

Now add examples of negative user/assistant prompts:

- Add user and assistant objects to the fewShotMessages array.
- The third object in fewShotMessages should have the role of "user" and the content should be an example of a negative review example, such as:
    - “Sweatshirt had a hole through the seams. Poor quality.”
    - “Package arrived late and damaged.”
- The fourth object in fewShotMessages should have the role of "assistant" and should be the reply, "Negative"

⍰ Use the following syntax:

```
let ARRAY_VARIABLE = [
  {
    role: "user",
    content: REVIEW_EXAMPLE
  },
  {
    role: "assistant",
    content: REVIEW_CLASSIFICATION
  },
]
```

⍰ Checkpoint 3 Passed

3.

Add a final user prompt to fewShotMessages with either a positive or negative review.

⍰ Use the following syntax:

```
ARRAY_VARIABLE = [
  // previous dictionary prompts
  // not represented
  {
    role: "user",
    content: REVIEW_TO_CLASSIFY
  },
]
```

⍰ Checkpoint 4 Passed

4.

Perform a classification:

- Create a chat completion with your model of choice and fewShotMessages passed as the messages argument.
- Print the model’s reply to see if the classification worked.

Use the following syntax:

```
const response = await client.chat.completions.create({
  model: MODEL_STRING,
  messages: messageArray
});

response.choices[0].message.content
```

main.js

```
import OpenAI from "openai";


const client = new OpenAI();


// Your code below:
let fewShotMessages = [
  {
    role: "user",
    content: "Clothing fit perfectly and was true to size."
  },
  {
    role: "assistant",
    content: "Positive"
  },
  {
    role: "user",
    content: "Package arrived late and damaged."
  },
  {
    role: "assistant",
    content: "Negative"
  },
  {
    role: "user",
    content: "Sweatshirt had a hole through the seams. Poor quality"
  }
];
```

```
const response = await client.chat.completions.create({
  model:"gpt-3.5-turbo",
  messages: fewShotMessages
});

console.log(response.choices[0].message.content);
```

Response

Negative