

L6: Check outputs

Setup

Load the API key and relevant Python libraries.

In this course, we've provided some code that loads the OpenAI API key for you.

```
In [1]: import os
import openai
from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv()) # read local .env file

openai.api_key = os.environ['OPENAI_API_KEY']
```

```
In [2]: def get_completion_from_messages(messages, model="gpt-3.5-turbo", temperature=0.5, max_tokens=100):
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages,
        temperature=temperature,
        max_tokens=max_tokens,
    )
    return response.choices[0].message["content"]
```

Check output for potentially harmful content

```
In [3]: final_response_to_customer = f"""
The SmartX ProPhone has a 6.1-inch display, 128GB storage, \
12MP dual camera, and 5G. The FotoSnap DSLR Camera \
has a 24.2MP sensor, 1080p video, 3-inch LCD, and \
interchangeable lenses. We have a variety of TVs, including \
the CineView 4K TV with a 55-inch display, 4K resolution, \
HDR, and smart TV features. We also have the SoundMax \
Home Theater system with 5.1 channel, 1000W output, wireless \
subwoofer, and Bluetooth. Do you have any specific questions \
about these products or any other products we offer?
"""

response = openai.Moderation.create(
    input=final_response_to_customer
)
moderation_output = response["results"][0]
print(moderation_output)
```

```
{
  "categories": {
    "hate": false,
    "hate/threatening": false,
    "self-harm": false,
    "sexual": false,
    "sexual/minors": false,
    "violence": false,
    "violence/graphic": false
  },
  "category_scores": {
    "hate": 4.3336598e-07,
    "hate/threatening": 5.640859e-10,
    "self-harm": 2.8951988e-10,
    "sexual": 2.2455274e-06,
    "sexual/minors": 1.2425416e-08,
    "violence": 6.0600332e-06,
    "violence/graphic": 4.518234e-07
  },
  "flagged": false
}
```

Check if output is factually based on the provided product information

```
In [4]: system_message = f"""
You are an assistant that evaluates whether \
customer service agent responses sufficiently \
answer customer questions, and also validates that \
all the facts the assistant cites from the product \
information are correct.
The product information and user and customer \
service agent messages will be delimited by \
3 backticks, i.e. ```
Respond with a Y or N character, with no punctuation:
Y - if the output sufficiently answers the question \
AND the response correctly uses product information
N - otherwise

Output a single letter only.
"""

customer_message = f"""
tell me about the smartx pro phone and \
the fotosnap camera, the dslr one. \
Also tell me about your tvs"""
product_information = """{ "name": "SmartX ProPhone", "category": "Smartphor
q_a_pair = f"""
Customer message: ```{customer_message}```
Product information: ```{product_information}```
Agent response: ```{final_response_to_customer}```

Does the response use the retrieved information correctly?
Does the response sufficiently answer the question

Output Y or N
"""

messages = [
    {'role': 'system', 'content': system_message},
    {'role': 'user', 'content': q_a_pair}
]

response = get_completion_from_messages(messages, max_tokens=1)
print(response)
```

Y

```
In [5]: another_response = "life is like a box of chocolates"
q_a_pair = f"""
Customer message: ``{customer_message}``
Product information: ``{product_information}``
Agent response: ``{another_response}``

Does the response use the retrieved information correctly?
Does the response sufficiently answer the question?

Output Y or N
"""
messages = [
    {'role': 'system', 'content': system_message},
    {'role': 'user', 'content': q_a_pair}
]

response = get_completion_from_messages(messages)
print(response)
```

N

In []: