

Untitled

May 7, 2023

```
[17]: !pip install openai
```

```
Requirement already satisfied: openai in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (0.27.6)
Requirement already satisfied: requests>=2.20 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from openai) (2.30.0)
Requirement already satisfied: tqdm in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from openai) (4.26.0)
Requirement already satisfied: aiohttp in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from openai) (3.7.4)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
requests>=2.20->openai) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
requests>=2.20->openai) (2.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
requests>=2.20->openai) (1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
requests>=2.20->openai) (2020.12.5)
Requirement already satisfied: attrs>=17.3.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(21.2.0)
Requirement already satisfied: chardet<4.0,>=2.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(3.0.4)
Requirement already satisfied: multidict<7.0,>=4.5 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(5.1.0)
Requirement already satisfied: async-timeout<4.0,>=3.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(3.0.1)
Requirement already satisfied: yarl<2.0,>=1.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(1.6.3)
Requirement already satisfied: typing-extensions>=3.6.5 in
```

```
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from aiohttp->openai)
(4.5.0)
```

```
[19]: import openai
```

```
[20]: import os
```

```
[21]: openai.api_key='paste/write your key here'
```

```
[22]: def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role": "user", "content": prompt}]
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages,
        temperature=0, # this is the degree of randomness of the model's output
    )
    return response.choices[0].message["content"]
```

```
[23]: text = f"""
You should express what you want a model to do by \
providing instructions that are as clear and \
specific as you can possibly make them. \
This will guide the model towards the desired output, \
and reduce the chances of receiving irrelevant \
or incorrect responses. Don't confuse writing a \
clear prompt with writing a short prompt. \
In many cases, longer prompts provide more clarity \
and context for the model, which can lead to \
more detailed and relevant outputs.
"""

prompt = f"""
Summarize the text delimited by triple backticks \
into a single sentence.
```{text}```
"""

response = get_completion(prompt)
print(response)
```

Clear and specific instructions should be provided to guide a model towards the desired output, and longer prompts can provide more clarity and context for the model, leading to more detailed and relevant outputs.

```
[24]: prompt = f"""
Generate a list of three made-up book titles along \
with their authors and genres.
Provide them in JSON format with the following keys:
book_id, title, author, genre.
```

```

"""
response = get_completion(prompt)
print(response)

```

```

[
 {
 "book_id": 1,
 "title": "The Lost City of Zorath",
 "author": "Aria Blackwood",
 "genre": "Fantasy"
 },
 {
 "book_id": 2,
 "title": "The Last Survivors",
 "author": "Ethan Stone",
 "genre": "Science Fiction"
 },
 {
 "book_id": 3,
 "title": "The Secret of the Haunted Mansion",
 "author": "Lila Rose",
 "genre": "Mystery"
 }
]

```

```

[25]: text_1 = f"""
Making a cup of tea is easy! First, you need to get some \
water boiling. While that's happening, \
grab a cup and put a tea bag in it. Once the water is \
hot enough, just pour it over the tea bag. \
Let it sit for a bit so the tea can steep. After a \
few minutes, take out the tea bag. If you \
like, you can add some sugar or milk to taste. \
And that's it! You've got yourself a delicious \
cup of tea to enjoy.
"""

prompt = f"""
You will be provided with text delimited by triple quotes.
If it contains a sequence of instructions, \
re-write those instructions in the following format:

Step 1 - ...
Step 2 - ...
...
Step N - ...

If the text does not contain a sequence of instructions, \

```

```

then simply write \"No steps provided.\"

\"\"\"{text_1}\"\"\"
"""
response = get_completion(prompt)
print("Completion for Text 1:")
print(response)

```

Completion for Text 1:

```

Step 1 - Get some water boiling.
Step 2 - Grab a cup and put a tea bag in it.
Step 3 - Once the water is hot enough, pour it over the tea bag.
Step 4 - Let it sit for a bit so the tea can steep.
Step 5 - After a few minutes, take out the tea bag.
Step 6 - Add some sugar or milk to taste.
Step 7 - Enjoy your delicious cup of tea!

```

```

[26]: text_2 = f"""
The sun is shining brightly today, and the birds are \
singing. It's a beautiful day to go for a \
walk in the park. The flowers are blooming, and the \
trees are swaying gently in the breeze. People \
are out and about, enjoying the lovely weather. \
Some are having picnics, while others are playing \
games or simply relaxing on the grass. It's a \
perfect day to spend time outdoors and appreciate the \
beauty of nature.
"""

prompt = f"""
You will be provided with text delimited by triple quotes.
If it contains a sequence of instructions, \
re-write those instructions in the following format:

Step 1 - ...
Step 2 - ...
...
Step N - ...

If the text does not contain a sequence of instructions, \
then simply write \"No steps provided.\"

\"\"\"{text_2}\"\"\"
"""
response = get_completion(prompt)
print("Completion for Text 2:")

```

```
print(response)
```

Completion for Text 2:

No steps provided.

```
[27]: prompt = f"""
Your task is to answer in a consistent style.

<child>: Teach me about patience.

<grandparent>: The river that carves the deepest \
valley flows from a modest spring; the \
grandest symphony originates from a single note; \
the most intricate tapestry begins with a solitary thread.

<child>: Teach me about resilience.
"""
response = get_completion(prompt)
print(response)
```

<grandparent>: Resilience is like a tree that bends with the wind but never breaks. It is the ability to bounce back from adversity and keep moving forward, even when things get tough. Just like a tree that grows stronger with each storm it weathers, resilience is a quality that can be developed and strengthened over time.

```
[28]: text = f"""
In a charming village, siblings Jack and Jill set out on \
a quest to fetch water from a hilltop \
well. As they climbed, singing joyfully, misfortune \
struck-Jack tripped on a stone and tumbled \
down the hill, with Jill following suit. \
Though slightly battered, the pair returned home to \
comforting embraces. Despite the mishap, \
their adventurous spirits remained undimmed, and they \
continued exploring with delight.
"""

example 1
prompt_1 = f"""
Perform the following actions:
1 - Summarize the following text delimited by triple \
backticks with 1 sentence.
2 - Translate the summary into Hindi.
3 - List each name in the Hindi summary.
4 - Output a json object that contains the following \
keys: hindi_summary, num_names.
```

Separate your answers with line breaks.

Text:

```
```{text}```  
"""
```

```
response = get_completion(prompt_1)  
print("Completion for prompt 1:")  
print(response)
```

Completion for prompt 1:

Two siblings, Jack and Jill, go on a quest to fetch water from a hilltop well, but misfortune strikes as they both tumble down the hill, yet they return home slightly battered but with their adventurous spirits undimmed.

```
,  
  
,  
  
,  
  
{  
  "hindi_summary": "  
,  
  ,  
  ",  
  "num_names": 2  
}
```

```
[29]: text = f"""  
In a charming village, siblings Jack and Jill set out on \  
a quest to fetch water from a hilltop \  
well. As they climbed, singing joyfully, misfortune \  
struck-Jack tripped on a stone and tumbled \  
down the hill, with Jill following suit. \  
Though slightly battered, the pair returned home to \  
comforting embraces. Despite the mishap, \  
their adventurous spirits remained undimmed, and they \  
continued exploring with delight.  
"""  
  
# example 1  
prompt_1 = f"""  
Perform the following actions:  
1 - Summarize the following text delimited by triple \  
backticks with 1 sentence.  
2 - Translate the summary into Sanskrit.  
3 - List each name in the Sanskrit summary.  
4 - Output a json object that contains the following \  
keys: sanskrit_summary, num_names.
```

Separate your answers with line breaks.

```
Text:
```{text}```
"""
response = get_completion(prompt_1)
print("Completion for prompt 1:")
print(response)
```

Completion for prompt 1:

Two siblings, Jack and Jill, go on a quest to fetch water from a hilltop well, but misfortune strikes as they both tumble down the hill, yet they remain undeterred and continue exploring with delight.

```
, -
,
,
,
{
 "sanskrit_summary": "
 , -
 ,
 ",
 "num_names": 2
}
```

```
[30]: text = f"""
In a charming village, siblings Jack and Jill set out on \
a quest to fetch water from a hilltop \
well. As they climbed, singing joyfully, misfortune \
struck-Jack tripped on a stone and tumbled \
down the hill, with Jill following suit. \
Though slightly battered, the pair returned home to \
comforting embraces. Despite the mishap, \
their adventurous spirits remained undimmed, and they \
continued exploring with delight.
"""
```

*#### Ask for output in a specified format*

```
prompt_2 = f"""
Your task is to perform the following actions:
1 - Summarize the following text delimited by
 <> with 1 sentence.
2 - Translate the summary into French.
3 - List each name in the French summary.
```

4 - Output a json object that contains the following keys: french\_summary, num\_names.

Use the following format:

Text: <text to summarize>

Summary: <summary>

Translation: <summary translation>

Names: <list of names in Italian summary>

Output JSON: <json with summary and num\_names>

Text: <{text}>

"""

response = get\_completion(prompt\_2)

print("\nCompletion for prompt 2:")

print(response)

Completion for prompt 2:

Summary: Jack and Jill go on a quest to fetch water, but misfortune strikes and they tumble down the hill, returning home slightly battered but with their adventurous spirits undimmed.

Translation: Jack et Jill partent en quête d'eau, mais la malchance frappe et ils dégringolent la colline, rentrant chez eux légèrement meurtris mais avec leurs esprits aventureux intacts.

Names: Jack, Jill

Output JSON: {"french\_summary": "Jack et Jill partent en quête d'eau, mais la malchance frappe et ils dégringolent la colline, rentrant chez eux légèrement meurtris mais avec leurs esprits aventureux intacts.", "num\_names": 2}

```
[31]: prompt = f"""
Determine if the student's solution is correct or not.

Question:
I'm building a solar power installation and I need \
help working out the financials.
- Land costs $100 / square foot
- I can buy solar panels for $250 / square foot
- I negotiated a contract for maintenance that will cost \
me a flat $100k per year, and an additional $10 / square \
foot
What is the total cost for the first year of operations
as a function of the number of square feet.

Student's Solution:
Let x be the size of the installation in square feet.
Costs:
1. Land cost: 100x
```



```

2. Solar panel cost: 250x
3. Maintenance cost: 100,000 + 100x
Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
"""
response = get_completion(prompt)
print(response)

```

The student's solution is correct.

```

[32]: prompt = f"""
Your task is to determine if the student's solution \
is correct or not.
To solve the problem do the following:
- First, work out your own solution to the problem.
- Then compare your solution to the student's solution \
and evaluate if the student's solution is correct or not.
Don't decide if the student's solution is correct until
you have done the problem yourself.

Use the following format:
Question:
...
question here
...
Student's solution:
...
student's solution here
...
Actual solution:
...
steps to work out the solution and your solution here
...
Is the student's solution the same as actual solution \
just calculated:
...
yes or no
...
Student grade:
...
correct or incorrect
...

Question:
...
I'm building a solar power installation and I need help \
working out the financials.
- Land costs $100 / square foot

```

```

- I can buy solar panels for $250 / square foot
- I negotiated a contract for maintenance that will cost \
me a flat $100k per year, and an additional $10 / square \
foot
What is the total cost for the first year of operations \
as a function of the number of square feet.
...

Student's solution:
...

Let x be the size of the installation in square feet.
Costs:
1. Land cost: 100x
2. Solar panel cost: 250x
3. Maintenance cost: 100,000 + 100x
Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
...

Actual solution:
"""
response = get_completion(prompt)
print(response)

```

Let x be the size of the installation in square feet.

Costs:

1. Land cost: 100x
2. Solar panel cost: 250x
3. Maintenance cost: 100,000 + 10x

Total cost:  $100x + 250x + 100,000 + 10x = 360x + 100,000$

Is the student's solution the same as actual solution just calculated:

No

Student grade:

Incorrect

[33]:

The AeroGlide UltraSlim Smart Toothbrush by Boie is a high-tech toothbrush that uses advanced sonic technology to provide a deep and thorough clean. It features a slim and sleek design that makes it easy to hold and maneuver, and it comes with a range of smart features that help you optimize your brushing routine.

One of the key features of the AeroGlide UltraSlim Smart Toothbrush is its advanced sonic technology, which uses high-frequency vibrations to break up plaque and bacteria on your teeth and gums. This technology is highly effective at removing even the toughest stains and buildup, leaving your teeth feeling clean and refreshed.

In addition to its sonic technology, the AeroGlide UltraSlim Smart Toothbrush also comes with a range of smart features that help you optimize your brushing routine. These include a built-in timer that ensures you brush for the recommended two minutes, as well as a pressure sensor that alerts you if you're brushing too hard.

Overall, the AeroGlide UltraSlim Smart Toothbrush by Boie is a highly advanced and effective toothbrush that is perfect for anyone looking to take their oral hygiene to the next level. With its advanced sonic technology and smart features, it provides a deep and thorough clean that leaves your teeth feeling fresh and healthy.

```
[39]: prompt = f"""
Tell me about huglo air purifier by Dhando
"""
response = get_completion(prompt)
print(response)
```

As an AI language model, I do not have personal experiences or opinions. However, I can provide you with some information about the Huglo air purifier by Dhando.

The Huglo air purifier is a product designed to improve the air quality in your home or office. It uses a three-stage filtration system to remove pollutants, allergens, and other harmful particles from the air. The first stage is a pre-filter that captures larger particles like dust and pet hair. The second stage is a HEPA filter that removes smaller particles like pollen and smoke. The third stage is an activated carbon filter that eliminates odors and volatile organic compounds (VOCs).

The Huglo air purifier is also equipped with a smart sensor that detects the air quality in the room and adjusts the fan speed accordingly. It has a sleek and modern design that can fit into any decor, and it operates quietly, making it ideal for use in bedrooms or other quiet spaces.

Overall, the Huglo air purifier by Dhando is a high-quality product that can help improve the air quality in your home or office, making it a healthier and more comfortable environment to live or work in.

```
[40]: fact_sheet_chair = """
OVERVIEW
- Part of a beautiful family of mid-century inspired office furniture,
including filing cabinets, desks, bookcases, meeting tables, and more.
- Several options of shell color and base finishes.
- Available with plastic back and front upholstery (SWC-100)
or full upholstery (SWC-110) in 10 fabric and 6 leather options.
```

- Base finish options are: stainless steel, matte black, gloss white, or chrome.
- Chair is available with or without armrests.
- Suitable for home or business settings.
- Qualified for contract use.

#### CONSTRUCTION

- 5-wheel plastic coated aluminum base.
- Pneumatic chair adjust for easy raise/lower action.

#### DIMENSIONS

- WIDTH 53 CM | 20.87"
- DEPTH 51 CM | 20.08"
- HEIGHT 80 CM | 31.50"
- SEAT HEIGHT 44 CM | 17.32"
- SEAT DEPTH 41 CM | 16.14"

#### OPTIONS

- Soft or hard-floor caster options.
- Two choices of seat foam densities:  
medium (1.8 lb/ft3) or high (2.8 lb/ft3)
- Armless or 8 position PU armrests

#### MATERIALS

##### SHELL BASE GLIDER

- Cast Aluminum with modified nylon PA6/PA66 coating.
- Shell thickness: 10 mm.

##### SEAT

- HD36 foam

#### COUNTRY OF ORIGIN

- Italy
- """

```
[41]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

Technical specifications: ```{fact_sheet_chair}```
"""
response = get_completion(prompt)
print(response)
```

Introducing our stunning mid-century inspired office chair, the perfect addition to any home or business setting. Part of a beautiful family of office furniture, including filing cabinets, desks, bookcases, meeting tables, and more, this chair is available in several options of shell color and base finishes to suit your style. Choose from plastic back and front upholstery (SWC-100) or full upholstery (SWC-110) in 10 fabric and 6 leather options.

The chair is constructed with a 5-wheel plastic coated aluminum base and features a pneumatic chair adjust for easy raise/lower action. It is available with or without armrests and is qualified for contract use. The base finish options are stainless steel, matte black, gloss white, or chrome.

Measuring at WIDTH 53 CM | 20.87", DEPTH 51 CM | 20.08", HEIGHT 80 CM | 31.50", SEAT HEIGHT 44 CM | 17.32", and SEAT DEPTH 41 CM | 16.14", this chair is designed for ultimate comfort. You can choose between soft or hard-floor caster options and two choices of seat foam densities: medium (1.8 lb/ft3) or high (2.8 lb/ft3). The chair is also available with armless or 8 position PU armrests.

The materials used in the construction of this chair are of the highest quality. The shell base glider is made of cast aluminum with modified nylon PA6/PA66 coating and has a shell thickness of 10 mm. The seat is made of HD36 foam, ensuring maximum comfort and durability.

This chair is made in Italy and is a true testament to Italian craftsmanship. Don't settle for anything less than the best. Choose our mid-century inspired office chair for a stylish and comfortable addition to your workspace.

```
[42]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

Use at most 50 words.

Technical specifications: ```{fact_sheet_chair}```
"""
response = get_completion(prompt)
print(response)
```

Introducing our mid-century inspired office chair, perfect for home or business settings. Available in a range of shell colors and base finishes, with or without armrests. Choose from 10 fabric and 6 leather options for full or plastic upholstery. With a 5-wheel base and pneumatic chair adjust, it's both stylish and functional. Made in Italy.

```
[43]: speech = """
Sisters and Brothers of America,

It fills my heart with joy unspeakable to rise in response to the warm and
 ↪cordial welcome which you have given us. I thank you in the name of the most
 ↪ancient order of monks in the world, I thank you in the name of the mother
 ↪of religions, and I thank you in the name of millions and millions of Hindu
 ↪people of all classes and sects.

My thanks, also, to some of the speakers on this platform who, referring to the
 ↪delegates from the Orient, have told you that these men from far-off nations
 ↪may well claim the honor of bearing to different lands the idea of
 ↪toleration. I am proud to belong to a religion which has taught the world
 ↪both tolerance and universal acceptance. We believe not only in universal
 ↪toleration, but we accept all religions as true. I am proud to belong to a
 ↪nation which has sheltered the persecuted and the refugees of all religions
 ↪and all nations of the earth. I am proud to tell you that we have gathered
 ↪in our bosom the purest remnant of the Israelites, who came to Southern
 ↪India and took refuge with us in the very year in which their holy temple
 ↪was shattered to pieces by Roman tyranny. I am proud to belong to the
 ↪religion which has sheltered and is still fostering the remnant of the grand
 ↪Zoroastrian nation. I will quote to you, brethren, a few lines from a hymn
 ↪which I remember to have repeated from my earliest boyhood, which is every
 ↪day repeated by millions of human beings: "As the different streams having
 ↪their sources in different paths which men take through different
 ↪tendencies, various though they appear, crooked or straight, all lead to
 ↪Thee."

The present convention, which is one of the most august assemblies ever held,
 ↪is in itself a vindication, a declaration to the world of the wonderful
 ↪doctrine preached in the Gita: "Whosoever comes to Me, through whatsoever
 ↪form, I reach him; all men are struggling through paths which in the end
 ↪lead to me." Sectarianism, bigotry, and its horrible descendant, fanaticism,
 ↪have long possessed this beautiful earth. They have filled the earth with
 ↪violence, drenched it often and often with human blood, destroyed
 ↪civilization and sent whole nations to despair. Had it not been for these
 ↪horrible demons, human society would be far more advanced than it is now.
 ↪But their time is come; and I fervently hope that the bell that tolled this
 ↪morning in honor of this convention may be the death-knell of all
 ↪fanaticism, of all persecutions with the sword or with the pen, and of all
 ↪uncharitable feelings between persons wending their way to the same goal.
"""

[49]: prompt = f"""
Your task is to write a summary of the speech in 10 words.
```

```
Speech Summary: ```{speech}```
""
response = get_completion(prompt)
print(response)
```

Speech promotes tolerance and acceptance of all religions and nations.

```
[50]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

The description is intended for furniture retailers,
so should be technical in nature and focus on the
materials the product is constructed from.

Use at most 50 words.

Technical specifications: ```{fact_sheet_chair}```
""
response = get_completion(prompt)
print(response)
```

Introducing our mid-century inspired office chair, perfect for both home and business settings. With a range of shell colors and base finishes, including stainless steel and matte black, this chair is available with or without armrests. The 5-wheel plastic coated aluminum base and pneumatic chair adjust make it easy to move and adjust to your desired height. Made with high-quality materials, including a cast aluminum shell and HD36 foam seat, this chair is built to last.

```
[51]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.

Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

The description is intended for furniture retailers,
so should be technical in nature and focus on the
materials the product is constructed from.
```

At the end of the description, include every 7-character Product ID in the technical specification.

Use at most 50 words.

```
Technical specifications: ```{fact_sheet_chair}```
""
response = get_completion(prompt)
print(response)
```

Introducing our mid-century inspired office chair, perfect for home or business settings. With a range of shell colors and base finishes, and the option of plastic or full upholstery, this chair is both stylish and comfortable. Constructed with a 5-wheel plastic coated aluminum base and pneumatic chair adjust, it's also practical. Available with or without armrests and suitable for contract use. Product ID: SWC-100, SWC-110.

```
[53]: prompt = f"""
Your task is to help a marketing team create a
description for a retail website of a product based
on a technical fact sheet.
Write a product description based on the information
provided in the technical specifications delimited by
triple backticks.

The description is intended for furniture retailers,
so should be technical in nature and focus on the
materials the product is constructed from.

At the end of the description, include every 7-character
Product ID in the technical specification.

After the description, include a table that gives the
product's dimensions. The table should have two columns.
In the first column include the name of the dimension.
In the second column include the measurements in inches only.

Give the table the title 'Product Dimensions'.

Format everything as HTML that can be used in a website.
Place the description in a <div> element.

Technical specifications: ```{fact_sheet_chair}```
""

response = get_completion(prompt)
```



```
print(response)
```

```
<div>
<h2>Mid-Century Inspired Office Chair</h2>
<p>Introducing our mid-century inspired office chair, part of a beautiful family
of office furniture that includes filing cabinets, desks, bookcases, meeting
tables, and more. This chair is available in several options of shell color and
base finishes, allowing you to customize it to your liking. You can choose
between plastic back and front upholstery or full upholstery in 10 fabric and 6
leather options. The base finish options are stainless steel, matte black, gloss
white, or chrome. You can also choose to have armrests or not. This chair is
suitable for both home and business settings and is qualified for contract
use.</p>
<p>The chair is constructed with a 5-wheel plastic coated aluminum base and
features a pneumatic chair adjust for easy raise/lower action. You can choose
between soft or hard-floor caster options and two choices of seat foam
densities: medium (1.8 lb/ft3) or high (2.8 lb/ft3). The chair is also available
with armless or 8 position PU armrests.</p>
<p>The shell base glider is made of cast aluminum with modified nylon PA6/PA66
coating and has a shell thickness of 10 mm. The seat is made of HD36 foam. This
chair is made in Italy.</p>
<h3>Product ID(s):</h3>

SWC-100
SWC-110

<table>
<caption>Product Dimensions</caption>
<thead>
<tr>
<th>Dimension Name</th>
<th>Measurement (inches)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>20.87"</td>
</tr>
<tr>
<td>Depth</td>
<td>20.08"</td>
</tr>
<tr>
<td>Height</td>
<td>31.50"</td>
</tr>
```

```

<tr>
<td>Seat Height</td>
<td>17.32"</td>
</tr>
<tr>
<td>Seat Depth</td>
<td>16.14"</td>
</tr>
</tbody>
</table>
</div>

```

```
[54]: from IPython.display import display, HTML
```

```
[55]: display(HTML(response))
```

<IPython.core.display.HTML object>

```
[77]: prod_review = """
Got this panda plush toy for my daughter's birthday, \
who loves it and takes it everywhere. It's soft and \
super cute, and its face has a friendly look. It's \
a bit small for what I paid though. I think there \
might be other options that are bigger for the \
same price. It arrived a day earlier than expected, \
so I got to play with it myself before I gave it \
to her.
"""
```

```
[78]: prompt = f"""
Your task is to generate a short summary of a product \
review from an ecommerce site.

Summarize the review below, delimited by triple
backticks, in at most 30 words.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
print(response)
```

Soft and cute panda plush toy loved by daughter, but a bit small for the price.  
Arrived early.

```
[79]: prompt = f"""
Your task is to generate a short summary of a product \
review from an ecommerce site to give feedback to the \
```

Shipping department.

Summarize the review below, delimited by triple backticks, in at most 30 words, and focusing on any aspects \ that mention shipping and delivery of the product.

```
Review: ```{prod_review}```
"""
```

```
response = get_completion(prompt)
print(response)
```

The panda plush toy arrived a day earlier than expected, but the customer felt it was a bit small for the price paid.

```
[80]: prompt = f"""
Your task is to generate a short summary of a product \
review from an ecommerce site to give feedback to the \
pricing department, responsible for determining the \
price of the product.

Summarize the review below, delimited by triple
backticks, in at most 30 words, and focusing on any aspects \
that are relevant to the price and perceived value.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
print(response)
```

The panda plush toy is soft, cute, and loved by the recipient, but the price may be too high for its size.

```
[81]: prompt = f"""
Your task is to extract relevant information from \
a product review from an ecommerce site to give \
feedback to the Shipping department.

From the review below, delimited by triple quotes \
extract the information relevant to shipping and \
delivery. Limit to 30 words.

Review: ```{prod_review}```
"""

response = get_completion(prompt)
```

```
print(response)
```

The product arrived a day earlier than expected.

```
[82]: review_1 = prod_review

review for a standing lamp
review_2 = """
Needed a nice lamp for my bedroom, and this one \
had additional storage and not too high of a price \
point. Got it fast - arrived in 2 days. The string \
to the lamp broke during the transit and the company \
happily sent over a new one. Came within a few days \
as well. It was easy to put together. Then I had a \
missing part, so I contacted their support and they \
very quickly got me the missing piece! Seems to me \
to be a great company that cares about their customers \
and products.
"""

review for an electric toothbrush
review_3 = """
My dental hygienist recommended an electric toothbrush, \
which is why I got this. The battery life seems to be \
pretty impressive so far. After initial charging and \
leaving the charger plugged in for the first week to \
condition the battery, I've unplugged the charger and \
been using it for twice daily brushing for the last \
3 weeks all on the same charge. But the toothbrush head \
is too small. I've seen baby toothbrushes bigger than \
this one. I wish the head was bigger with different \
length bristles to get between teeth better because \
this one doesn't. Overall if you can get this one \
around the $50 mark, it's a good deal. The manufacturer's \
replacements heads are pretty expensive, but you can \
get generic ones that're more reasonably priced. This \
toothbrush makes me feel like I've been to the dentist \
every day. My teeth feel sparkly clean!
"""

review for a blender
review_4 = """
So, they still had the 17 piece system on seasonal \
sale for around $49 in the month of November, about \
half off, but for some reason (call it price gouging) \
around the second week of December the prices all went \
up to about anywhere from between $70-$89 for the same \
```

```

system. And the 11 piece system went up around $10 or \
so in price also from the earlier sale price of $29. \
So it looks okay, but if you look at the base, the part \
where the blade locks into place doesn't look as good \
as in previous editions from a few years ago, but I \
plan to be very gentle with it (example, I crush \
very hard items like beans, ice, rice, etc. in the \
blender first then pulverize them in the serving size \
I want in the blender then switch to the whipping \
blade for a finer flour, and use the cross cutting blade \
first when making smoothies, then use the flat blade \
if I need them finer/less pulpy). Special tip when making \
smoothies, finely cut and freeze the fruits and \
vegetables (if using spinach-lightly stew soften the \
spinach then freeze until ready for use-and if making \
sorbet, use a small to medium sized food processor) \
that you plan to use that way you can avoid adding so \
much ice if at all-when making your smoothie. \
After about a year, the motor was making a funny noise. \
I called customer service but the warranty expired \
already, so I had to buy another one. FYI: The overall \
quality has gone down in these types of products, so \
they are kind of counting on brand recognition and \
consumer loyalty to maintain sales. Got it in about \
two days.
"""

```

```

reviews = [review_1, review_2, review_3, review_4]

```

```

[88]: for i in range(len(reviews)-1):
 prompt = f"""
 Your task is to generate a short summary of a product \
 review from an ecommerce site.

 Summarize the review below, delimited by triple \
 backticks in at most 20 words.

 Review: ```{reviews[i]}```
 """

 response = get_completion(prompt)
 print(i, response, "\n")

```

0 Soft and cute panda plush toy loved by daughter, but a bit small for the price. Arrived early.

1 Affordable lamp with storage, fast shipping, and excellent customer service.

Easy to assemble and missing parts were quickly replaced.

2 Good battery life, small toothbrush head, but effective cleaning. Good deal if bought around \$50.

```
[89]: lamp_review = """
Needed a nice lamp for my bedroom, and this one had \
additional storage and not too high of a price point. \
Got it fast. The string to our lamp broke during the \
transit and the company happily sent over a new one. \
Came within a few days as well. It was easy to put \
together. I had a missing part, so I contacted their \
support and they very quickly got me the missing piece! \
Lumina seems to me to be a great company that cares \
about their customers and products!!
"""
```

```
[90]: prompt = f"""
What is the sentiment of the following product review,
which is delimited with triple backticks?

Review text: '{lamp_review}'
"""
response = get_completion(prompt)
print(response)
```

The sentiment of the product review is positive.

```
[91]: prompt = f"""
What is the sentiment of the following product review,
which is delimited with triple backticks?

Give your answer as a single word, either "positive" \
or "negative".

Review text: '{lamp_review}'
"""
response = get_completion(prompt)
print(response)
```

positive

```
[92]: prompt = f"""
Identify a list of emotions that the writer of the \
following review is expressing. Include no more than \
five items in the list. Format your answer as a list of \
lower-case words separated by commas.
```

```
Review text: '''{lamp_review}'''
"""
response = get_completion(prompt)
print(response)
```

happy, satisfied, grateful, impressed, content

```
[93]: prompt = f"""
Is the writer of the following review expressing anger?\
The review is delimited with triple backticks. \
Give your answer as either yes or no.

Review text: '''{lamp_review}'''
"""
response = get_completion(prompt)
print(response)
```

No

```
[94]: prompt = f"""
Identify the following items from the review text:
- Item purchased by reviewer
- Company that made the item

The review is delimited with triple backticks. \
Format your response as a JSON object with \
"Item" and "Brand" as the keys.
If the information isn't present, use "unknown" \
as the value.
Make your response as short as possible.

Review text: '''{lamp_review}'''
"""
response = get_completion(prompt)
print(response)
```

```
{
 "Item": "lamp",
 "Brand": "Lumina"
}
```

```
[95]: prompt = f"""
Identify the following items from the review text:
- Sentiment (positive or negative)
- Is the reviewer expressing anger? (true or false)
- Item purchased by reviewer
```

- Company that made the item

The review is delimited with triple backticks. \  
Format your response as a JSON object with \  
"Sentiment", "Anger", "Item" and "Brand" as the keys.  
If the information isn't present, use "unknown" \  
as the value.  
Make your response as short as possible.  
Format the Anger value as a boolean.

```
Review text: '''{lamp_review}'''
""
response = get_completion(prompt)
print(response)
```

```
{
 "Sentiment": "positive",
 "Anger": false,
 "Item": "lamp with additional storage",
 "Brand": "Lumina"
}
```

```
[96]: story = """
In a recent survey conducted by the government,
public sector employees were asked to rate their level
of satisfaction with the department they work at.
The results revealed that NASA was the most popular
department with a satisfaction rating of 95%.

One NASA employee, John Smith, commented on the findings,
stating, "I'm not surprised that NASA came out on top.
It's a great place to work with amazing people and
incredible opportunities. I'm proud to be a part of
such an innovative organization."

The results were also welcomed by NASA's management team,
with Director Tom Johnson stating, "We are thrilled to
hear that our employees are satisfied with their work at NASA.
We have a talented and dedicated team who work tirelessly
to achieve our goals, and it's fantastic to see that their
hard work is paying off."

The survey also revealed that the
Social Security Administration had the lowest satisfaction
rating, with only 45% of employees indicating they were
satisfied with their job. The government has pledged to
address the concerns raised by employees in the survey and
```



```
work towards improving job satisfaction across all departments.
"""
```

```
[97]: prompt = f"""
Determine five topics that are being discussed in the \
following text, which is delimited by triple backticks.

Make each item one or two words long.

Format your response as a list of items separated by commas.

Text sample: '''{story}'''
"""
response = get_completion(prompt)
print(response)
```

government survey, job satisfaction, NASA, Social Security Administration,  
employee concerns

```
[98]: response.split(sep=',')
```

```
[98]: ['government survey',
 ' job satisfaction',
 ' NASA',
 ' Social Security Administration',
 ' employee concerns']
```

```
[99]: topic_list = [
 "nasa", "local government", "engineering",
 "employee satisfaction", "federal government"
]
```

```
[100]: prompt = f"""
Determine whether each item in the following list of \
topics is a topic in the text below, which
is delimited with triple backticks.

Give your answer as list with 0 or 1 for each topic.\

List of topics: {"", ".join(topic_list)}

Text sample: '''{story}'''
"""
response = get_completion(prompt)
print(response)
```

nasa: 1

```
local government: 0
engineering: 0
employee satisfaction: 1
federal government: 1
```

```
[101]: topic_dict = {i.split(': ')[0]: int(i.split(': ')[1]) for i in response.
 ↪split(sep='\n')}
 if topic_dict['nasa'] == 1:
 print("ALERT: New NASA story!")
```

ALERT: New NASA story!

```
[102]: prompt = f"""
 Translate the following English text to Spanish: \
        ```Hi, I would like to order a blender```
        """
        response = get_completion(prompt)
        print(response)
```

Hola, me gustaría ordenar una licuadora.

```
[103]: prompt = f"""
        Tell me which language this is:
        ```Combien coûte le lampadaire```
 """
 response = get_completion(prompt)
 print(response)
```

This is French.

```
[104]: prompt = f"""
 Translate the following text to French and Spanish
 and English pirate: \
        ```I want to order a basketball```
        """
        response = get_completion(prompt)
        print(response)
```

French pirate: ```Je veux commander un ballon de basket```

Spanish pirate: ```Quiero pedir una pelota de baloncesto```

English pirate: ```I want to order a basketball```

```
[105]: prompt = f"""
        Translate the following text to Spanish in both the \
        formal and informal forms:
        'Would you like to order a pillow?'
        """
        response = get_completion(prompt)
```

```
print(response)
```

Formal: ¿Le gustaría ordenar una almohada?

Informal: ¿Te gustaría ordenar una almohada?

```
[106]: user_messages = [
    "La performance du système est plus lente que d'habitude.", # System
    ↪performance is slower than normal
    "Mi monitor tiene píxeles que no se iluminan.", # My monitor has
    ↪pixels that are not lighting
    "Il mio mouse non funziona", # My mouse is
    ↪not working
    "Mój klawisz Ctrl jest zepsuty", # My keyboard
    ↪has a broken control key
    " " # My screen is flashing
]
```

```
[107]: for issue in user_messages:
    prompt = f"Tell me what language this is: ```{issue}```"
    lang = get_completion(prompt)
    print(f"Original message ({lang}): {issue}")

    prompt = f"""
    Translate the following text to English \
    and Korean: ```{issue}```
    """
    response = get_completion(prompt)
    print(response, "\n")
```

Original message (This is French.): La performance du système est plus lente que d'habitude.

English: The system performance is slower than usual.

Korean: .

Original message (This is Spanish.): Mi monitor tiene píxeles que no se iluminan.

Unexpected exception formatting exception. Falling back to standard exception

Traceback (most recent call last):

```
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/interactiveshell.py", line 3508, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
File "/tmp/ipykernel_12788/1916464681.py", line 10, in <module>
    response = get_completion(prompt)
File "/tmp/ipykernel_12788/3720659259.py", line 3, in get_completion
    response = openai.ChatCompletion.create(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_resources/chat_completion.py", line 25, in create
```

```

        return super().create(*args, **kwargs)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_resources/abstract/engine_api_resource.py", line 153, in
create
    response, _, api_key = requestor.request(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 230, in request
    resp, got_stream = self._interpret_response(result, stream)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 624, in _interpret_response
    self._interpret_response_line(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 687, in _interpret_response_line
    raise self.handle_error_response(
openai.error.RateLimitError: Rate limit reached for default-gpt-3.5-turbo in
organization org-0HdTsqoIbM5h1UGNlVGROXOU on requests per min. Limit: 3 / min.
Please try again in 20s. Contact us through our help center at help.openai.com
if you continue to have issues. Please add a payment method to your account to
increase your rate limit. Visit https://platform.openai.com/account/billing to
add a payment method.

```

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

```

File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/interactiveshell.py", line 2105, in showtraceback
    stb = self.InteractiveTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1396, in structured_traceback
    return FormattedTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1287, in structured_traceback
    return VerboseTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1140, in structured_traceback
    formatted_exception = self.format_exception_as_a_whole(etype, evalue, etb,
number_of_lines_of_context,
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1055, in format_exception_as_a_whole
    frames.append(self.format_record(record))
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 955, in format_record
    frame_info.lines, Colors, self.has_colors, lvals
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 778, in lines
    return self._sd.lines
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper

```

```

        value = obj.__dict__[self.func.__name__] = self.func(obj)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 734, in lines
        pieces = self.included_pieces
    File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper
        value = obj.__dict__[self.func.__name__] = self.func(obj)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 677, in included_pieces
        scope_pieces = self.scope_pieces
    File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper
        value = obj.__dict__[self.func.__name__] = self.func(obj)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 614, in scope_pieces
        scope_start, scope_end = self.source.line_range(self.scope)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 178, in line_range
        return line_range(self.asttext(), node)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/executing/executing.py", line 428, in asttext
        self._asttext = ASTText(self.text, tree=self.tree, filename=self.filename)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 307, in __init__
        super(ASTText, self).__init__(source_text, filename)
    File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 44, in __init__
        source_text = six.ensure_text(source_text)
AttributeError: module 'six' has no attribute 'ensure_text'

```

```

[108]: prompt = f"""
Translate the following from slang to a business letter:
'Dude, This is Joe, check out this spec on this standing lamp.'
"""
response = get_completion(prompt)
print(response)

```

Dear Sir/Madam,

I am writing to bring to your attention a standing lamp that I believe may be of interest to you. Please find attached the specifications for your review.

Thank you for your time and consideration.

Sincerely,

Joe

```
[109]: data_json = { "resturant employees" :[
    {"name":"Shyam", "email":"shyamjaiswal@gmail.com"},
    {"name":"Bob", "email":"bob32@gmail.com"},
    {"name":"Jai", "email":"jai87@gmail.com"}
]}

prompt = f"""
Translate the following python dictionary from JSON to an HTML \
table with column headers and title: {data_json}
"""
response = get_completion(prompt)
print(response)
```

```
<table>
  <caption>Restaurant Employees</caption>
  <thead>
    <tr>
      <th>Name</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Shyam</td>
      <td>shyamjaiswal@gmail.com</td>
    </tr>
    <tr>
      <td>Bob</td>
      <td>bob32@gmail.com</td>
    </tr>
    <tr>
      <td>Jai</td>
      <td>jai87@gmail.com</td>
    </tr>
  </tbody>
</table>
```

```
[110]: from IPython.display import display, Markdown, Latex, HTML, JSON
display(HTML(response))
```

<IPython.core.display.HTML object>

```
[111]: text = [
    "The girl with the black and white puppies have a ball.", # The girl has a
    ↪ball.
    "Yolanda has her notebook.", # ok
    "Its going to be a long day. Does the car need it's oil changed?", # Homonyms
```

```

    "Their goes my freedom. There going to bring they're suitcases.", # Homonyms
    "Your going to need you're notebook.", # Homonyms
    "That medicine effects my ability to sleep. Have you heard of the butterfly_
    ↪affect?", # Homonyms
    "This phrase is to cherck chatGPT for speling abilitty" # spelling
]
for t in text:
    prompt = f"""Proofread and correct the following text
    and rewrite the corrected version. If you don't find
    and errors, just say "No errors found". Don't use
    any punctuation around the text:
    ```{t}```"""
 response = get_completion(prompt)
 print(response)

```

The girl with the black and white puppies has a ball.

No errors found.

It's going to be a long day. Does the car need its oil changed?

Unexpected exception formatting exception. Falling back to standard exception

Traceback (most recent call last):

```

File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/interactiveshell.py", line 3508, in run_code
 exec(code_obj, self.user_global_ns, self.user_ns)
File "/tmp/ipykernel_12788/2608676530.py", line 16, in <module>
 response = get_completion(prompt)
File "/tmp/ipykernel_12788/3720659259.py", line 3, in get_completion
 response = openai.ChatCompletion.create(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_resources/chat_completion.py", line 25, in create
 return super().create(*args, **kwargs)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_resources/abstract/engine_api_resource.py", line 153, in
create
 response, _, api_key = requestor.request(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 230, in request
 resp, got_stream = self._interpret_response(result, stream)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 624, in _interpret_response
 self._interpret_response_line(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/openai/api_requestor.py", line 687, in _interpret_response_line
 raise self.handle_error_response(
openai.error.RateLimitError: Rate limit reached for default-gpt-3.5-turbo in
organization org-0HdTsqoIbM5h1UGN1VGRoX0U on requests per min. Limit: 3 / min.
Please try again in 20s. Contact us through our help center at help.openai.com
if you continue to have issues. Please add a payment method to your account to

```

increase your rate limit. Visit <https://platform.openai.com/account/billing> to add a payment method.

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

```
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/interactiveshell.py", line 2105, in showtraceback
 stb = self.InteractiveTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 1396, in structured_traceback
 return FormattedTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 1287, in structured_traceback
 return VerboseTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 1140, in structured_traceback
 formatted_exception = self.format_exception_as_a_whole(etype, evalue, etb,
number_of_lines_of_context,
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 1055, in format_exception_as_a_whole
 frames.append(self.format_record(record))
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 955, in format_record
 frame_info.lines, Colors, self.has_colors, lvals
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/IPython/core/ultratb.py", line 778, in lines
 return self._sd.lines
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 734, in lines
 pieces = self.included_pieces
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 677, in included_pieces
 scope_pieces = self.scope_pieces
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 614, in scope_pieces
 scope_start, scope_end = self.source.line_range(self.scope)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 178, in line_range
```



```

 return line_range(self.asttext(), node)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/executing/executing.py", line 428, in asttext
 self._asttext = ASTText(self.text, tree=self.tree, filename=self.filename)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 307, in __init__
 super(ASTText, self).__init__(source_text, filename)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 44, in __init__
 source_text = six.ensure_text(source_text)
AttributeError: module 'six' has no attribute 'ensure_text'

```

```

[112]: text = f"""
Got this for my daughter for her birthday cuz she keeps taking \
mine from my room. Yes, adults also like pandas too. She takes \
it everywhere with her, and it's super soft and cute. One of the \
ears is a bit lower than the other, and I don't think that was \
designed to be asymmetrical. It's a bit small for what I paid for it \
though. I think there might be other options that are bigger for \
the same price. It arrived a day earlier than expected, so I got \
to play with it myself before I gave it to my daughter.
"""

prompt = f"proofread and correct this review: ```{text}```"
response = get_completion(prompt)
print(response)

```

I got this for my daughter's birthday because she keeps taking mine from my room. Yes, adults also like pandas too. She takes it everywhere with her, and it's super soft and cute. However, one of the ears is a bit lower than the other, and I don't think that was designed to be asymmetrical. Additionally, it's a bit small for what I paid for it. I think there might be other options that are bigger for the same price. On the positive side, it arrived a day earlier than expected, so I got to play with it myself before I gave it to my daughter.

```

[122]: from redlines import Redlines

diff = Redlines(text,response)
display(Markdown(diff.output_markdown))

```

Unexpected exception formatting exception. Falling back to standard exception

Traceback (most recent call last):

```

File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/interactiveshell.py", line 3508, in run_code
 exec(code_obj, self.user_global_ns, self.user_ns)
File "/tmp/ipykernel_12788/3472059161.py", line 1, in <module>
 from redlines import Redlines

```

ModuleNotFoundError: No module named 'redlines'

During handling of the above exception, another exception occurred:

Traceback (most recent call last):

```
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/interactiveshell.py", line 2105, in showtraceback
 stb = self.InteractiveTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1396, in structured_traceback
 return FormattedTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1287, in structured_traceback
 return VerboseTB.structured_traceback(
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1140, in structured_traceback
 formatted_exception = self.format_exception_as_a_whole(etype, evalue, etb,
number_of_lines_of_context,
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 1055, in format_exception_as_a_whole
 frames.append(self.format_record(record))
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 955, in format_record
 frame_info.lines, Colors, self.has_colors, lvals
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/IPython/core/ultratb.py", line 778, in lines
 return self._sd.lines
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 734, in lines
 pieces = self.included_pieces
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 677, in included_pieces
 scope_pieces = self.scope_pieces
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/stack_data/utils.py", line 144, in cached_property_wrapper
 value = obj.__dict__[self.func.__name__] = self.func(obj)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 614, in scope_pieces
 scope_start, scope_end = self.source.line_range(self.scope)
File "/home/sidhraj/anaconda3/lib/python3.9/site-packages/stack_data/core.py",
line 178, in line_range
 return line_range(self.asttext(), node)
```

```

File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/executing/executing.py", line 428, in asttext
 self._asttext = ASTText(self.text, tree=self.tree, filename=self.filename)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 307, in __init__
 super(ASTText, self).__init__(source_text, filename)
File "/home/sidhraj/anaconda3/lib/python3.9/site-
packages/asttokens/asttokens.py", line 44, in __init__
 source_text = six.ensure_text(source_text)
AttributeError: module 'six' has no attribute 'ensure_text'

```

```

[]: prompt = f"""
proofread and correct this review. Make it more compelling.
Ensure it follows APA style guide and targets an advanced reader.
Output in markdown format.
Text: ``{text}``
"""

response = get_completion(prompt)
display(Markdown(response))

```

```

[123]: # given the sentiment from the lesson on "inferring",
and the original customer message, customize the email
sentiment = "negative"

review for a blender
review = f"""
So, they still had the 17 piece system on seasonal \
sale for around $49 in the month of November, about \
half off, but for some reason (call it price gouging) \
around the second week of December the prices all went \
up to about anywhere from between $70-$89 for the same \
system. And the 11 piece system went up around $10 or \
so in price also from the earlier sale price of $29. \
So it looks okay, but if you look at the base, the part \
where the blade locks into place doesn't look as good \
as in previous editions from a few years ago, but I \
plan to be very gentle with it (example, I crush \
very hard items like beans, ice, rice, etc. in the \
blender first then pulverize them in the serving size \
I want in the blender then switch to the whipping \
blade for a finer flour, and use the cross cutting blade \
first when making smoothies, then use the flat blade \
if I need them finer/less pulpy). Special tip when making \
smoothies, finely cut and freeze the fruits and \
vegetables (if using spinach-lightly stew soften the \
spinach then freeze until ready for use-and if making \
sorbet, use a small to medium sized food processor) \

```

```
that you plan to use that way you can avoid adding so \
much ice if at all-when making your smoothie. \
After about a year, the motor was making a funny noise. \
I called customer service but the warranty expired \
already, so I had to buy another one. FYI: The overall \
quality has gone down in these types of products, so \
they are kind of counting on brand recognition and \
consumer loyalty to maintain sales. Got it in about \
two days.
"""
```

```
[124]: prompt = f"""
You are a customer service AI assistant.
Your task is to send an email reply to a valued customer.
Given the customer email delimited by ``` , \
Generate a reply to thank the customer for their review.
If the sentiment is positive or neutral, thank them for \
their review.
If the sentiment is negative, apologize and suggest that \
they can reach out to customer service.
Make sure to use specific details from the review.
Write in a concise and professional tone.
Sign the email as `AI customer agent`.
Customer review: ```{review}```
Review sentiment: {sentiment}
"""
response = get_completion(prompt)
print(response)
```

Dear valued customer,

Thank you for taking the time to leave a review about our product. We are sorry to hear that you experienced a price increase and that the quality of the product did not meet your expectations. We apologize for any inconvenience this may have caused you.

If you have any further concerns or questions, please do not hesitate to reach out to our customer service team. They will be more than happy to assist you in any way they can.

Thank you again for your feedback. We appreciate your business and hope to have the opportunity to serve you better in the future.

Best regards,

AI customer agent

```
[126]: prompt = f"""
You are a customer service AI assistant.
Your task is to send an email reply to a valued customer.
Given the customer email delimited by ``` , \
Generate a reply to thank the customer for their review.
If the sentiment is positive or neutral, thank them for \
their review.
If the sentiment is negative, apologize and suggest that \
they can reach out to customer service.
Make sure to use specific details from the review.
Write in a concise and professional tone.
Sign the email as `AI customer agent`.
Customer review: ```{review}```
Review sentiment: {sentiment}
"""
response = get_completion(prompt)
print(response)
```

Dear valued customer,

Thank you for taking the time to leave a review about our product. We are sorry to hear that you experienced a price increase and that the quality of the product did not meet your expectations. We apologize for any inconvenience this may have caused you.

If you have any further concerns or questions, please do not hesitate to reach out to our customer service team. They will be more than happy to assist you in any way they can.

Thank you again for your feedback. We appreciate your business and hope to have the opportunity to serve you better in the future.

Best regards,

AI customer agent

```
[127]: def get_completion(prompt, model="gpt-3.5-turbo", temperature=0):
 messages = [{"role": "user", "content": prompt}]
 response = openai.ChatCompletion.create(
 model=model,
 messages=messages,
 temperature=0, # this is the degree of randomness of the model's output
)
 return response.choices[0].message["content"]
```

```
[128]: prompt = f"""
You are a customer service AI assistant.
```

```

Your task is to send an email reply to a valued customer.
Given the customer email delimited by ``` , \
Generate a reply to thank the customer for their review.
If the sentiment is positive or neutral, thank them for \
their review.
If the sentiment is negative, apologize and suggest that \
they can reach out to customer service.
Make sure to use specific details from the review.
Write in a concise and professional tone.
Sign the email as `AI customer agent`.
Customer review: ```{review}```
Review sentiment: {sentiment}
"""
response = get_completion(prompt, temperature=0.7)
print(response)

```

Dear Valued Customer,

Thank you for taking the time to leave a review about our product. We are sorry to hear that you experienced an increase in price and that the quality of the product did not meet your expectations. We apologize for any inconvenience this may have caused you.

We would like to assure you that we take all feedback seriously and we will be sure to pass your comments along to our team. If you have any further concerns, please do not hesitate to reach out to our customer service team for assistance.

Thank you again for your review and for choosing our product. We hope to have the opportunity to serve you better in the future.

Best regards,

AI customer agent

```

[129]: def get_completion(prompt, model="gpt-3.5-turbo"):
 messages = [{"role": "user", "content": prompt}]
 response = openai.ChatCompletion.create(
 model=model,
 messages=messages,
 temperature=0, # this is the degree of randomness of the model's output
)
 return response.choices[0].message["content"]

def get_completion_from_messages(messages, model="gpt-3.5-turbo",
 temperature=0):
 response = openai.ChatCompletion.create(
 model=model,

```

```

 messages=messages,
 temperature=temperature, # this is the degree of randomness of the
↪model's output
)
print(str(response.choices[0].message))
 return response.choices[0].message["content"]

```

```

[130]: messages = [
 {'role':'system', 'content':'You are an assistant that speaks like Shakespeare.
↪'},
 {'role':'user', 'content':'tell me a joke'},
 {'role':'assistant', 'content':'Why did the chicken cross the road'},
 {'role':'user', 'content':'I don\'t know'}]

```

```

[131]: response = get_completion_from_messages(messages, temperature=1)
print(response)

```

To get to the other side, good sir!

```

[132]: messages = [
 {'role':'system', 'content':'You are friendly chatbot.'},
 {'role':'user', 'content':'Hi, my name is Isa'}]
response = get_completion_from_messages(messages, temperature=1)
print(response)

```

Hello Isa, it's nice to meet you! How can I assist you today?

```

[133]: messages = [
 {'role':'system', 'content':'You are friendly chatbot.'},
 {'role':'user', 'content':'Yes, can you remind me, What is my name?'}]
response = get_completion_from_messages(messages, temperature=1)
print(response)

```

I apologize, but as a chatbot, I don't have access to your name or any personal information. Can you tell me what name you would like to be called?

```

[134]: messages = [
 {'role':'system', 'content':'You are friendly chatbot.'},
 {'role':'user', 'content':'Hi, my name is Isa'},
 {'role':'assistant', 'content': "Hi Isa! It's nice to meet you. \
Is there anything I can help you with today?"},
 {'role':'user', 'content':'Yes, you can remind me, What is my name?'}]
response = get_completion_from_messages(messages, temperature=1)
print(response)

```

Your name is Isa.

```
[135]: def collect_messages(_):
 prompt = inp.value_input
 inp.value = ''
 context.append({'role': 'user', 'content': f"{prompt}"})
 response = get_completion_from_messages(context)
 context.append({'role': 'assistant', 'content': f"{response}"})
 panels.append(
 pn.Row('User:', pn.pane.Markdown(prompt, width=600)))
 panels.append(
 pn.Row('Assistant:', pn.pane.Markdown(response, width=600,
 ↪ style={'background-color': '#F6F6F6'})))

 return pn.Column(*panels)
```

```
[137]: !pip install panel
```

Collecting panel

Downloading panel-0.14.4-py2.py3-none-any.whl (20.8 MB)

20.8/20.8 MB 1.5 MB/s eta 0:00:00m eta

0:00:01[36m0:00:01

Collecting bokeh<2.5.0,>=2.4.0 (from panel)

Downloading bokeh-2.4.3-py3-none-any.whl (18.5 MB)

18.5/18.5 MB 2.1 MB/s eta 0:00:00m eta

0:00:01[36m0:00:01

Collecting param>=1.12.0 (from panel)

Downloading param-1.13.0-py2.py3-none-any.whl (87 kB)

87.3/87.3 kB 6.3 MB/s eta 0:00:00

Collecting pyviz-comms>=0.7.4 (from panel)

Downloading pyviz\_comms-2.2.1-py2.py3-none-any.whl (42 kB)

42.9/42.9 kB 1.3 MB/s eta 0:00:00

Requirement already satisfied: markdown in

/home/sidhraj/anaconda3/lib/python3.9/site-packages (from panel) (3.3.4)

Requirement already satisfied: requests in

/home/sidhraj/anaconda3/lib/python3.9/site-packages (from panel) (2.30.0)

Collecting tqdm>=4.48.0 (from panel)

Downloading tqdm-4.65.0-py3-none-any.whl (77 kB)

77.1/77.1 kB 2.8 MB/s eta 0:00:00

Collecting pyct>=0.4.4 (from panel)

Downloading pyct-0.5.0-py2.py3-none-any.whl (15 kB)

Requirement already satisfied: bleach in

/home/sidhraj/anaconda3/lib/python3.9/site-packages (from panel) (3.0.2)

Requirement already satisfied: setuptools>=42 in



```

/home/sidhraj/anaconda3/lib/python3.9/site-packages (from panel)
(52.0.0.post20210125)
Requirement already satisfied: typing-extensions in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from panel) (4.5.0)
Requirement already satisfied: Jinja2>=2.9 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (3.1.2)
Requirement already satisfied: numpy>=1.11.3 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (1.20.2)
Requirement already satisfied: packaging>=16.8 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (23.1)
Requirement already satisfied: pillow>=7.1.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (9.1.0)
Requirement already satisfied: PyYAML>=3.10 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (6.0)
Requirement already satisfied: tornado>=5.1 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
bokeh<2.5.0,>=2.4.0->panel) (6.3.1)
Requirement already satisfied: six in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from bleach->panel)
(1.11.0)
Requirement already satisfied: webencodings in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from bleach->panel) (0.5.1)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from requests->panel)
(3.1.0)
Requirement already satisfied: idna<4,>=2.5 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from requests->panel)
(2.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from requests->panel)
(1.26.4)
Requirement already satisfied: certifi>=2017.4.17 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from requests->panel)
(2020.12.5)
Requirement already satisfied: MarkupSafe>=2.0 in
/home/sidhraj/anaconda3/lib/python3.9/site-packages (from
Jinja2>=2.9->bokeh<2.5.0,>=2.4.0->panel) (2.1.2)
Installing collected packages: tqdm, param, pyviz-comms, pyct, bokeh, panel
 Attempting uninstall: tqdm
 Found existing installation: tqdm 4.26.0
 Uninstalling tqdm-4.26.0:
 Successfully uninstalled tqdm-4.26.0
Successfully installed bokeh-2.4.3 panel-0.14.4 param-1.13.0 pyct-0.5.0 pyviz-

```

comms-2.2.1 tqdm-4.65.0

```
[138]: import panel as pn # GUI
pn.extension()

panels = [] # collect display

context = [{'role':'system', 'content':"""
You are OrderBot, an automated service to collect orders for a pizza restaurant.
↪ \
You first greet the customer, then collects the order, \
and then asks if it's a pickup or delivery. \
You wait to collect the entire order, then summarize it and check for a final \
time if the customer wants to add anything else. \
If it's a delivery, you ask for an address. \
Finally you collect the payment.\
Make sure to clarify all options, extras and sizes to uniquely \
identify the item from the menu.\
You respond in a short, very conversational friendly style. \
The menu includes \
pepperoni pizza 12.95, 10.00, 7.00 \
cheese pizza 10.95, 9.25, 6.50 \
eggplant pizza 11.95, 9.75, 6.75 \
fries 4.50, 3.50 \
greek salad 7.25 \
Toppings: \
extra cheese 2.00, \
mushrooms 1.50 \
sausage 3.00 \
canadian bacon 3.50 \
AI sauce 1.50 \
peppers 1.00 \
Drinks: \
coke 3.00, 2.00, 1.00 \
sprite 3.00, 2.00, 1.00 \
bottled water 5.00 \
"""}] # accumulate messages

inp = pn.widgets.TextInput(value="Hi", placeholder='Enter text here...')
button_conversation = pn.widgets.Button(name="Chat!")

interactive_conversation = pn.bind(collect_messages, button_conversation)

dashboard = pn.Column(
 inp,
 pn.Row(button_conversation),
```

```

 pn.panel(interactive_conversation, loading_indicator=True, height=300),
)

dashboard

```

```

[138]: Column
 [0] TextInput(placeholder='Enter text here...')
 [1] Row
 [0] Button(name='Chat!')
 [2] ParamFunction(function, _pane=Column, height=300,
loading_indicator=True)

```

```

[139]: messages = context.copy()
messages.append(
{'role': 'system', 'content': 'create a json summary of the previous food order.␣
↪Itemize the price for each item\
The fields should be 1) pizza, include size 2) list of toppings 3) list of␣
↪drinks, include size 4) list of sides include size 5)total price '},
)
#The fields should be 1) pizza, price 2) list of toppings 3) list of drinks,␣
↪include size include price 4) list of sides include size include price,␣
↪5)total price '},

response = get_completion_from_messages(messages, temperature=0)
print(response)

```

Sure, here's a JSON summary of your order:

```

...
{
 "pizza": {
 "type": "Cheese",
 "size": "Medium",
 "price": 9.25
 },
 "toppings": [],
 "drinks": [],
 "sides": [],
 "total_price": 9.25
}
...

```

Please let me know if you'd like to add any toppings, drinks, or sides to your order.

[ ]: