# openai api usage

- 整理：黄剑
- 仓库：https://github.com/MDGSF/openai_usage

依赖库：

- `pip install -i https://pypi.tuna.tsinghua.edu.cn/simple openai`
- `pip install -i https://pypi.tuna.tsinghua.edu.cn/simple python-dotenv`
- `pip install -i https://pypi.tuna.tsinghua.edu.cn/simple redlines`

In [4]:

```python
import openai
import os
from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv())

openai.api_key = os.getenv("OPENAI_API_KEY")
openai.proxy = {
    "http": "http://127.0.0.1:1080",
    "https": "http://127.0.0.1:1080"
}
model='gpt-3.5-turbo'
```

In [5]:

```python
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,
    )
    return response.choices[0].message["content"]
```

In [6]:

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

## Principles of Prompting

- Principle 1: Write clear and specific instructions, clear != short.
- Principle 2: Give the model time to think.

### Tactic 1: Use delimiters

- Triple quotes: """
- Triple backticks: ```
- Triple dashes: ---
- Angle brackets: < >

- XML tags: `<tag> </tag>`

In [4]:
~~~
text = f"""
Attention once again turned to the US-Mexico border this week as a pandemic-era poli

Large numbers of migrants and refugees rushed to the border in hopes of seeking prot

At the same time, President Joe Biden’s administration had dispatched additional tr

Here are some of the stories that have marked the past few days along the 3,140km (1
"""
prompt = f"""
Summarize the text delimited by triple backticks \
into a single sentence.
```{text}```
"""
response = get_completion(prompt)
print (response)
~~~

```
The expiration of the pandemic-era policy in the US that allowed authorities to rapi
dly expel most asylum seekers under the pretext of public health has led to large nu
mbers of migrants and refugees rushing to the US-Mexico border in hopes of seeking p
rotection, while President Joe Biden’s administration dispatched additional troops a
nd resources to brace for an influx of arrivals.
```

## Tactic 2: Ask for structured output

HTML or JSON

In [5]:
```
prompt = f"""
Generate a list of three made-up book titles along \
with their authors and genres.
Privide then 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 Life of Bees",
    "author": "Lila Rose",
    "genre": "Romance"
  }
]
```

## Tactic 3: Check whether conditions are satisfied

Check assumptions required to do the task

In [6]:

```
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 sequences of instructions, \
re-write those instructions in the following format:

Step 1 = ...
Step 2 = ...
...
Step N = ...

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

\"\"\"{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 = If you like, add some sugar or milk to taste.
Step 7 = Enjoy your delicious cup of tea!
```

## Tactic 4: Few-shot prompting

Give successful examples of completing tasks. Then ask model to perform the task.

In [7]:

```
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 nost 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 t
hings get tough. Just like a tree, we must have strong roots and a solid foundation
to weather any storm that comes our way.
```

# Principle 2

Give the model time to think

## Tactic 1: Specify the steps to complete a task

```
Step 1: ...
Step 2: ...
...
Step N: ...
```

In [8]:

~~~
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 nishap, \
their adventurous spirits remained undinned, and they \
continued exploring with delight.
"""

prompt = f"""
Perform the following actions:
1 - Summarize the following text delimited by triple \
backticks 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.

Separate your answers with line breaks.

Text:
```{text}```
"""

response = get_completion(prompt)
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 well on a hill, but
misfortune strikes when Jack trips and tumbles down the hill, with Jill following su
it, yet they return home slightly battered but with their adventurous spirits still
intact.

Deux frères et sœurs, Jack et Jill, partent en quête d'eau d'un puits sur une collin
e, mais la malchance frappe quand Jack trébuche et dégringole la colline, avec Jill
qui suit, mais ils rentrent chez eux légèrement meurtris mais avec leurs esprits ave
ntureux toujours intacts.

Jack, Jill.

{
  "french_summary": "Deux frères et sœurs, Jack et Jill, partent en quête d'eau d'un
puits sur une colline, mais la malchance frappe quand Jack trébuche et dégringole la
colline, avec Jill qui suit, mais ils rentrent chez eux légèrement meurtris mais ave
c leurs esprits aventureux toujours intacts.",
  "num_names": 2
}
```

In [9]:

```
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 nishap, \
their adventurous spirits remained undinned, and they \
continued exploring with delight.
"""

prompt2 = 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 to summarize: <{text}>
"""

response = get_completion(prompt2)
print ("Completion 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 a hill before returning home with their adventurous spirits still intac
t.
Translation: Jack et Jill partent en quête d'eau, mais un malheur frappe et ils tomb
ent d'une colline avant de rentrer chez eux avec leurs esprits aventureux toujours i
ntacts.
Names: Jack, Jill
Output JSON: {"french_summary": "Jack et Jill partent en quête d'eau, mais un malheu
r frappe et ils tombent d'une colline avant de rentrer chez eux avec leurs esprits a
venturex toujours intacts.", "num_names": 2}
```

## Tactic 2: Instruct the model to work out its own solution before rushing to a conclusion

In [4]:

```
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: 100,00 + 100x
Total cost: 100x + 250x + 100,000 + 100x = 450x + 100,000
"""

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

```
The student's solution is correct.
```

In [6]:

~~~
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 the 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: 100,00 + 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 the actual solution just calculated:
No

student grade:
Incorrect
```

# Model Limitations

- Hallucination
- Makes statements that sound plausible but are not true
- Reducing hallucinations:
- First find relevant information,

- then answer the question
- based on the relevant information.

In [7]:

```
prompt = f"""
Tell me about AeroGlide UltraSlim Smart Toothbrush by Boie
"""


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

```
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 a
nd sleek design that makes it easy to hold and maneuver, and it comes with a range o
f 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 bacter
ia 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 c
omes with a range of smart features that help you optimize your brushing routine. Th
ese 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 e
ffective 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.
```

# Iterative Prompt Development

Prompt guidelines:

- Be clear and specific
- Analyze why result does not give desired output.
- Refine the idea and the prompt
- Repeat

Iterative Process:

- Try something
- Analyze where the result does not give what your want
- Clarify instructions, give more time to think
- Refine prompts with a batch of examples

# Summarize

总结文本

In [8]:

```
Prod_review = f"""
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 frendly look. It's \
a bit samll 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.
"""

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. Arri
ved early.
```

In [10]:

````
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 wa
s a bit small for the price paid.
```

In [11]:

````
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)
````

~~~
```Soft and cute panda plush toy, loved by daughter. Friendly face. Small for the pr
ice, other options may offer better value. Arrived early.```
~~~

In [12]:

````
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." (11 words)
```

# Inferring

推理

### 示例一：判断是积极情绪还是消极情绪

In [13]:

```
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!!
"""



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
```

### 示例二：提取情绪

In [14]:

```
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)
```

```
satisfied, grateful, impressed, content, pleased
```

### 示例三：判断是否生气

In [15]:

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

Review text: '''{lamp_review}'''
"""

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

```
No
```

**示例四：提取商品和品牌**

In [16]:

```python
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 with additional storage",
  "Brand": "Lumina"
}
```

**示例五：同时提取多个信息**

In [17]:

```python
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"
}
```

### 示例六：提取主题词语

In [18]:

```
story = """
In a recent survey conducted by the goverment,
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.
Web 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 department.
"""


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)
print (response.split(sep=','))
```

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

### 示例七：判断文章和给定主题是否相关

In [19]:

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

```
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
```

# Transforming

Translating to a different language, and more!

#### 示例一：一种语言翻译为另一种语言

In [20]:

~~~
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.
```

#### 示例二：判断给定文本是什么语言

In [21]:

~~~
prompt = f"""
Tell me which language this is:
```Hola, me gustaría ordenar una licuadora.```
"""

response = get_completion(prompt)
print (response)
~~~

```
This is Spanish.
```

#### 示例三：一种语言翻译为其它多种语言

In [22]:

~~~
prompt = f"""
Translate the following text to Chinese, French, Spanish and English: \
```I want to order a basketball```
"""

response = get_completion(prompt)
print (response)
~~~

```
Chinese: 我想订购一个篮球 (Wǒ xiǎng dìnggòu yīgè lánqiú)
French: Je veux commander un ballon de basket (Jeuh vuh koh-mahn-deh uhn bah-lawn duh bah-sket)
Spanish: Quiero ordenar una pelota de baloncesto (Kee-eh-roh or-deh-nahr oo-nah peh-loh-tah deh bah-lon-ces-toh)
English: I want to order a basketball
```

### 示例四：翻译为正式和非正式

In [23]:

```python
prompt = f"""
Translate the following English 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?
```

### 示例五：翻译

In [ ]:

```python
user_messages = [
  "My screen is flickering.",
  "Mi pantalla está parpadeando.",
  "我的屏幕在闪烁"
]

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)
```

### 示例六：把口语转换为正式表达

In [25]:

```python
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
```

### 示例七：把JSON转换为HTML

In [26]:
```
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 HTML \
table with column headers and title: {data_json}
"""

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

```
<table>
  <caption>Resturant 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>
```

In [28]:
```
from IPython.core.display import HTML
print (display(HTML(response)))
```

Resturant Employees

| Name | Email |
|---|---|
| Shyam | shyamjaiswal@gmail.com |
| Bob | bob32@gmail.com |
| Jai | jai87@gmail.com |

```
None
```

### 示例八：检查语法错误，并修改为正确的

In [ ]:
```
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 your're notebook.", # Homonyms
  "That medicine effects my ability to sleep. Have your heard of the butterfly affec
  "This phrase is to cherck chatGPT for speling abilitty" # spelling
]
```

```
for t in text:
  prompt = f"Proofread and correct: ```{t}```"
  response = get_completion(prompt)
  print (response)
```

**示例九：检查语法错误，并修改为正确的**

In [ ]:

~~~
for t in text:
  prompt = f"""
  Proofread and correct the following text
  and rewrite the corrected version. If you don't find
  any errors, just say "No errors found":
  ```{t}```
  """
  response = get_completion(prompt)
  print (response)
~~~

**示例十：检查语法错误，并修改为正确的**

In [29]:

```
text = f"""
Got this for my daughter for her birthday cuz she keeps taking \
mine from my froom. 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 that 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. Y
es, adults also like pandas too. She takes it everywhere with her, and it's super so
ft and cute. However, one of the ears is a bit lower than the other, and I don't thi
nk that was designed to be asymmetrical. Additionally, it's a bit small for what I p
aid 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."
```

In [32]:

```
from redlines import Redlines
from IPython.core.display import Markdown
diff = Redlines(text, response)
display(Markdown(diff.output_markdown))
```

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

**示例十一：检查语法错误，并修改为正确的**

In [33]:

```
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)
print (response)
```

```
# Review of a Panda Plush Toy

I purchased this panda plush toy as a birthday gift for my daughter, who has a pench
ant for taking my belongings from my room. As an adult, I also have a soft spot for
pandas, and I must say that this toy did not disappoint.

My daughter takes it everywhere with her, and I can attest to its softness and cuten
ess. However, I did notice that one of the ears is slightly lower than the other, wh
ich I don't believe was intended to be asymmetrical.

While I am pleased with the quality of the toy, I do feel that it is a bit small for
the price I paid. I suspect that there may be other options available that are large
r for the same cost.

Despite this, I was pleasantly surprised to receive the toy a day earlier than expec
ted, which gave me the opportunity to play with it myself before giving it to my dau
ghter. Overall, I would recommend this panda plush toy to anyone who loves these ado
rable creatures, but be aware of its size and slight asymmetry.
```

In [34]:

```
display(Markdown(response))
```

# Review of a Panda Plush Toy

I purchased this panda plush toy as a birthday gift for my daughter, who has a penchant for taking my belongings from my room. As an adult, I also have a soft spot for pandas, and I must say that this toy did not disappoint.

My daughter takes it everywhere with her, and I can attest to its softness and cuteness. However, I did notice that one of the ears is slightly lower than the other, which I don't believe was intended to be asymmetrical.

While I am pleased with the quality of the toy, I do feel that it is a bit small for the price I paid. I suspect that there may be other options available that are larger for the same cost.

Despite this, I was pleasantly surprised to receive the toy a day earlier than expected, which gave me the opportunity to play with it myself before giving it to my daughter. Overall, I would recommend this panda plush toy to anyone who loves these adorable creatures, but be aware of its size and slight asymmetry.

## Expanding

Expand a shorter text to a longer text (email, essay)

In [37]:

```
sentiment = "negative"
```

```
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 \
play to be very gentle with it (example, I crush \
very hard items like beans, ics, 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 fruite 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 done 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.
"""

prompt = f"""
Your 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)
print (response)
```

```
Dear valued customer,

Thank you for taking the time to leave a review about your recent purchase. 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.

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
```

## Temperature

temperature 是一个随机数， 数字越小，表示结果越确定， 数字越大，表示结果越随机。

### example

my favorite food is xxx.

- pizza: 53%
- sushi: 30%
- tacos: 5%

temperature = 0 时，输出可能是：

- my favorite food is pizza
- my favorite food is pizza
- my favorite food is pizza

temperature = 0.3 时，输出可能是：

- my favorite food is pizza
- my favorite food is sushi
- my favorite food is pizza

temperature = 0.7 时，输出可能是：

- my favorite food is tacos
- my favorite food is sushi
- my favorite food is pizza

**Note:**

- for tasks that require reliability, predictability，set temperature = 0
- for tasks that require variety, use temperature higher
- 当 temperature = 0 时，每次都是相同的输出。
- 当 temperature = 0.7 时，每次都会得到不同的输出。

# Chatbots

In [42]:

```
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 dont know'}
]

response = get_completion_from_messages(messages, temperature=1.)
print (response)
```

```
To get to yonder fairest hen, forsooth!
```

In [43]:

```
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?
```

In [44]:

```
messages = [
  {'role':'system', 'content':'You are friendly chatbot.'},
  {'role':'user', 'content':'Yes, you can remind me, What is my name?'},
]

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

```
I'm sorry, but as a chatbot, I don't have access to your personal information, so I
don't know your name. However, I'm happy to chat with you and answer any questions y
ou may have.
```

In [45]:

```
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)
```

```
Of course! Your name is Isa.
```

In [10]:

```
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)))
    return pn.Column(*panels)


import panel as pn
```

```
pn.extension()

panels = []

context = [
  {
    'role':'system',
    'content':"""
You are OrderBot, an automated service to collect orders for a pizza restaurant. \
Your 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 \
"""
  }
]

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)
)

dashboard
```

Out[10]:

Enter text here...

Chat!

User:

Assistant: Hello! Welcome to our pizza restaurant. How can I assist you with your order today?

In [ ]: