# Transforming

In this notebook, we will explore how to use Large Language Models for text transformation tasks such as language translation spelling and grammar checking, tone adjustment, and format conversion.

## Setup

In [1]:
```
import openai
import os

from dotenv import load_dotenv, find_dotenv
_ = load_dotenv(find_dotenv()) # read local .env file

openai.api_key  = os.getenv('OPENAI_API_KEY')
```

In [2]:
```
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=temperature,
    )
    return response.choices[0].message["content"]
```

## Translation

ChatGPT is trained with sources in many languages. This gives the model the ability to do translation. Here are some example of how to use this capability.

In [3]:
```
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.
```

In [4]:
```
prompt = f"""
Tell me which language this is:
```Combien coûte le lampadaire?```
"""
response = get_completion(prompt)
print(response)
```

```
This is French.
```

In [5]:
```
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```
```

In [6]:
```
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?
```

## Universal Translator

Imagine you are in charge of IT at a large multinational e-commerce company. Users are messaging you with IT issues in all their native languages. Your staff is from all over the world and speaks only their native languages. You need a universal translator!

In [7]:
```
user_messages = [
  "La performance du système est plus lente que d'habitude.",  # System performance is s
  "Mi monitor tiene píxeles que no se iluminan.",              # My monitor has pixels t
  "Il mio mouse non funziona",                                 # My mouse is not working
  "Mój klawisz Ctrl jest zepsuty",                             # My keyboard has a broke
  "我的屏幕在闪烁"                                              # My screen is flashing
]
```
```

In [8]:
```python
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'habitu
de.
English: The system performance is slower than usual.
Korean: 시스템 성능이 평소보다 느립니다.

Original message (This is Spanish.): Mi monitor tiene píxeles que no se iluminan.
English: My monitor has pixels that don't light up.
Korean: 내 모니터에는 불이 켜지지 않는 픽셀이 있습니다.

Original message (This is Italian.): Il mio mouse non funziona
English: My mouse is not working.
Korean: 내 마우스가 작동하지 않습니다.

Original message (This is Polish.): Mój klawisz Ctrl jest zepsuty
English: My Ctrl key is broken.
Korean: 제 Ctrl 키가 고장 났어요.

Original message (This is Chinese (Simplified).): 我的屏幕在闪烁
English: My screen is flickering.
Korean: 내 화면이 깜빡입니다.
```

## Try it yourself!

Try some translations on your own!

In [9]:
```
text = f"""
London is a beautiful city
"""
prompt = f"""
Given a text, identify the source language.
Then translate the source text into Spanish, Chinese, and Malayalam.
The text is delimited with triple backticks.
The response should follow the format:
Source Language:
Spanish:
Chinese:
Malayalam:

Given text - ```{text}```
"""
response = get_completion(prompt)
print(response)
```

```
Source Language: English
Spanish: Londres es una ciudad hermosa
Chinese: 伦敦是一个美丽的城市 (Lúndūn shì yīgè měilì de chéngshì)
Malayalam: ലണ്ടൻ ഒരു സുന്ദരമായ നഗരമാണ് (Laṇṭaṇ oru sundaramāya nagaramāṇ)
```

## Tone Transformation

Writing can vary based on the intended audience. ChatGPT can produce different tones.

In [10]:
```
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
```

## Format Conversion

ChatGPT can translate between formats. The prompt should describe the input and output formats.

In [11]:
```
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>
```

In [12]:
```
from IPython.display import display, Markdown, Latex, HTML, JSON
display(HTML(response))
```

Restaurant Employees

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

## Spellcheck/Grammar check.

Here are some examples of common grammar and spelling problems and the LLM's response.

To signal to the LLM that you want it to proofread your text, you instruct the model to 'proofread' or 'proofread and correct'.

In [13]:
```
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?",
  "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?
Their goes my freedom. There going to bring they're suitcases.

Corrected: There goes my freedom. They're going to bring their suitcases.
You're going to need your notebook.
That medicine affects my ability to sleep. Have you heard of the butterfly effect?
This phrase is to check ChatGPT for spelling ability.
```

In [14]:
```
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, ad
ults also like pandas too. She takes it everywhere with her, and it's super soft and cut
e. However, one of the ears is a bit lower than the other, and I don't think that was des
igned 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.
```
```

In [15]:
```
from redlines import Redlines

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 ~~room.~~ **room.** Yes, adults also like pandas ~~too.~~ **too.** She takes it everywhere with her, and it's super soft and ~~cute. One~~ **cute. 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~~ **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.~~ **daughter.**

In [16]:
```
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))
```

Title: A Soft and Cute Panda Plushie for All Ages

As an adult, I can attest that pandas are not just for kids. That's why I got this adorable panda plushie for my daughter's birthday, after she kept taking mine from my room. And let me tell you, it was a hit!

The plushie is super soft and cuddly, making it the perfect companion for my daughter. She takes it everywhere with her, and it has quickly become her favorite toy. However, I did notice that one of the ears is a bit lower than the other, which I don't think was designed to be asymmetrical. But that doesn't take away from its cuteness.

The only downside is that it's a bit small for the price I paid. I think there might be other options that are bigger for the same price. But overall, I'm happy with my purchase.

One thing that surprised me was that it arrived a day earlier than expected. This gave me the chance to play with it myself before giving it to my daughter. And I have to say, I was impressed with the quality and attention to detail.

In conclusion, if you're looking for a soft and cute panda plushie for yourself or a loved one, this is definitely a great option. Just be aware that it might be a bit smaller than expected.

## Try it yourself!

Try changing the instructions to form your own review.

In [20]:
```
starting_eleven = ['Ter Stegen', 'Arajo', 'Christeinsen', 'Balde', 'Cancelo', 'FDJ', 'Ki
```

Thanks to the following sites:

https://writingprompts.com/bad-grammar-examples/ (https://writingprompts.com/bad-grammar-examples/)

In [21]:
```
prompt = f"""
Given a list of players, create a starting eleven indicating the jersey number and posit
Follow the standard representation of football starting eleven.
The given text is in the form a list delimited with tripple backticks: ```{starting_elev
"""
response = get_completion(prompt)
print(response)
```

```
Here's a possible starting eleven:

```
1 - Ter Stegen (GK)
2 - Arajo (CB)
3 - Christeinsen (CB)
4 - Balde (CB)
5 - Cancelo (RB)
6 - FDJ (CDM)
7 - Kimmich (CM)
8 - Pedri (CAM)
9 - Gundagon (CM)
10 - Lewendoswki (ST)
11 - Dembele (LW)
```
```

In [22]:
```
prompt = f"""
Given a list of players, create a starting eleven indicating the jersey number and posit
Follow the standard representation of football starting eleven in a field pictioral view
Follow a 4-4-2 formation.
The given text is in the form a list delimited with tripple backticks: ```{starting_elev
"""
response = get_completion(prompt)
print(response)
```

```
```
Starting Eleven:

      1
   Arajo  3
2          4
      5
   FDJ   6
7          8
      9
  Lewandowski  11
         10

Jersey Numbers and Positions:
1 - Ter Stegen (GK)
2 - Cancelo (RB)
3 - Arajo (CB)
4 - Christeinsen (CB)
5 - Balde (LB)
6 - FDJ (CDM)
7 - Kimmich (RM)
8 - Pedri (CM)
9 - Gundagon (LM)
10 - Dembele (ST)
11 - Lewandowski (ST)
```
```

In [ ]: