Expanding

In this lesson, you will generate customer service emails that are tailored to each customer's review.

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): # Andrew me
    messages = [{"role": "user", "content": prompt}]
    response = openai.ChatCompletion.create(
        model=model,
        messages=messages,
        temperature=temperature, # this is the degree of randomness of the m
    )
    return response.choices[0].message["content"]
```

Customize the automated reply to a customer email

In [3]:
```
# 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 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.
"""
```

In [4]:
```
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 sor
ry 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 inconvenienc
e this may have caused you.

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

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
```

Remind the model to use details from the customer's email

In [5]:
```
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 for our 17 piece system. We a
re sorry to hear that you experienced a price increase and that the quality o
f the product did not meet your expectations. We apologize for any inconvenie
nce this may have caused you. As a customer service AI assistant, we recommen
d that you reach out to our customer service team for further assistance with
your concerns. They may be able to provide additional options or solutions fo
r you. We appreciate your loyalty to our brand and hope to have the opportuni
ty to better serve you in the future.

Thank you again for your feedback.

AI customer agent
```

Try experimenting on your own!

In [6]:
```
mask1 = f"""
Masking principle in Speech signal processing
"""

mask2 = f"""
Masking concept in spatial-temporal-channel attention mechanisms
where mask is referred to the learned weights that transforms the data
"""

mask3 = f"""
Masking concept in sequential self attention mechanism, that is used to cons
"""

mask = [mask1, mask2, mask3]

prompt=f"""
Given a list about different masking concepts in signal processing and machi
explain each masking concept with equation and diagram giving emphasis to au
The given list is delimited with tripple backticks.

The response should follow in the form of a scientic article something like

Given list: ```{mask}```
"""

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

Masking Concepts in Signal Processing and Machine Learning

Signal processing and machine learning are two fields that have revolutionized the way we interact with technology. One of the key concepts in these fields is masking, which refers to the process of selectively filtering out certain parts of a signal or data. In this article, we will explore three different masking concepts and their applications in audio processing.

1. Masking Principle in Speech Signal Processing

The masking principle is a fundamental concept in speech signal processing. It refers to the phenomenon where the presence of one sound can make another sound inaudible or less audible. This occurs because the two sounds share the same frequency range, and the louder sound masks the quieter one.

The masking principle can be mathematically modeled using the following equation:

$$M(f) = 10 \log10 [S(f) / N(f)]$$

where M(f) is the masking threshold at frequency f, S(f) is the power spectral density of the masking sound, and N(f) is the power spectral density of the masked sound. The masking threshold represents the minimum level of the masked sound that can be detected in the presence of the masking sound.

In audio processing, the masking principle is used to optimize compression algorithms and noise reduction techniques. By identifying the frequencies that are most susceptible to masking, these algorithms can selectively reduce the amplitude of those frequencies to improve the overall quality of the audio signal.

2. Masking Concept in Spatial-Temporal-Channel Attention Mechanisms

Spatial-temporal-channel attention mechanisms are a type of machine learning algorithm that is used for image and video processing. These algorithms use a mask, which is referred to as the learned weights, to selectively attend to certain parts of the data.

The mask is calculated using the following equation:

$$M = softmax(Wq * WkT)$$

where M is the mask, Wq is the query matrix, and Wk is the key matrix. The softmax function is used to normalize the mask so that the sum of its values is equal to one.

In audio processing, spatial-temporal-channel attention mechanisms can be used to selectively attend to certain parts of an audio signal. For example, in speech recognition, the algorithm can be trained to attend to the parts of the signal that contain speech while ignoring background noise.

3. Masking Concept in Sequential Self-Attention Mechanism

Sequential self-attention mechanisms are another type of machine learning algorithm that is used for sequential data processing. These algorithms use a mask to constrain the context of sequential attention, which means that they only attend to a certain number of previous and future elements in the sequenc

e.

The mask is calculated using the following equation:

M(i, j) = 1 if j <= i else 0

where M(i, j) is the mask, i is the current element in the sequence, and j is the element being attended to. The mask is set to one for all elements up to and including the current element, and zero for all elements after the curren t element.

In audio processing, sequential self-attention mechanisms can be used for tas ks such as music transcription and speech recognition. By constraining the co ntext of attention, these algorithms can improve the accuracy of their predic tions and reduce the amount of computational resources required.

Conclusion

Masking is a powerful concept in signal processing and machine learning that has a wide range of applications in audio processing. By selectively filterin g out certain parts of a signal or data, masking algorithms can improve the q uality and accuracy of audio processing tasks. The three masking concepts dis cussed in this article - the masking principle, spatial-temporal-channel atte ntion mechanisms, and sequential self-attention mechanisms - are just a few e xamples of the many ways in which masking is used in these fields.

In [ ]: