Negative Prompting and Avoiding Undesired Outputs

Overview

This tutorial explores the concept of negative prompting and techniques for avoiding undesired outputs when working with large language models. We'll focus on using OpenAI's GPT models and the LangChain library to implement these strategies.

Motivation

As AI language models become more powerful, it's crucial to guide their outputs effectively. Negative prompting allows us to specify what we don't want in the model's responses, helping to refine and control the generated content. This approach is particularly useful when dealing with sensitive topics, ensuring factual accuracy, or maintaining a specific tone or style in the output.

Key Components

1. Using negative examples to guide the model
2. Specifying exclusions in prompts
3. Implementing constraints using LangChain
4. Evaluating and refining negative prompts

Method Details

We'll start by setting up our environment with the necessary libraries. Then, we'll explore different techniques for negative prompting:

1. Basic negative examples: We'll demonstrate how to provide examples of undesired outputs to guide the model.
2. Explicit exclusions: We'll use prompts that specifically state what should not be included in the response.
3. Constraint implementation: Using LangChain, we'll create more complex prompts that enforce specific constraints on the output.
4. Evaluation and refinement: We'll discuss methods to assess the effectiveness of our negative prompts and iteratively improve them.

Throughout the tutorial, we'll use practical examples to illustrate these concepts and provide code snippets for implementation.

Conclusion

By the end of this tutorial, you'll have a solid understanding of negative prompting techniques and how to apply them to avoid undesired outputs from language models. These skills will enable you to create more controlled, accurate, and appropriate AI-generated content for various applications.

Setup

First, let's import the necessary libraries and set up our environment.

In [4]:

```
import os
import re
from langchain_openai import ChatOpenAI
from langchain.prompts import PromptTemplate

# Load environment variables
from dotenv import load_dotenv
load_dotenv()

# Set up OpenAI API key
os.environ["OPENAI_API_KEY"] = os.getenv('OPENAI_API_KEY')

# Initialize the language model
llm = ChatOpenAI(model="gpt-4o-mini")

def get_response(prompt):
    """Helper function to get response from the language model."""
    return llm.invoke(prompt).content
```

1. Using Negative Examples

Let's start with a simple example of using negative examples to guide the model's output.

In [5]:

```
negative_example_prompt = PromptTemplate(
    input_variables=["topic"],
    template="""Provide a brief explanation of {topic}.
    Do NOT include any of the following in your explanation:
    - Technical jargon or complex terminology
    - Historical background or dates
    - Comparisons to other related topics
    Your explanation should be simple, direct, and focus only on the cor
)

response = get_response(negative_example_prompt.format(topic="photosynth
print(response)
```

```
Photosynthesis is the process by which green plants, algae, and some bacte
ria convert sunlight into energy. They take in carbon dioxide from the air
and water from the soil. Using sunlight, they transform these ingredients
into glucose, a type of sugar that provides energy for growth and developm
ent. As a byproduct, they release oxygen into the air, which is essential
for many living beings.
```

2. Specifying Exclusions

Now, let's explore how to explicitly specify what should be excluded from the response.

In [6]:

```python
exclusion_prompt = PromptTemplate(
    input_variables=["topic", "exclude"],
    template="""Write a short paragraph about {topic}.
    Important: Do not mention or reference anything related to {exclude}
)

response = get_response(exclusion_prompt.format(
    topic="the benefits of exercise",
    exclude="weight loss or body image"
))
print(response)
```

```
Exercise offers a multitude of benefits that extend beyond physical appear
ance. Engaging in regular physical activity enhances cardiovascular healt
h, strengthens muscles, and improves flexibility, contributing to overall
physical well-being. Additionally, exercise is known to boost mood and red
uce symptoms of anxiety and depression through the release of endorphins,
fostering a sense of happiness and mental clarity. It also promotes better
sleep quality, increases energy levels, and enhances cognitive function, l
eading to improved focus and productivity in daily tasks. Ultimately, inco
rporating exercise into one's routine cultivates a healthier, more vibrant
lifestyle.
```

3. Implementing Constraints

Let's use LangChain to create more complex prompts that enforce specific constraints on the output.

In [7]:

```python
constraint_prompt = PromptTemplate(
    input_variables=["topic", "style", "excluded_words"],
    template="""Write a {style} description of {topic}.
    Constraints:
    1. Do not use any of these words: {excluded_words}
    2. Keep the description under 100 words
    3. Do not use analogies or metaphors
    4. Focus only on factual information"""
)

response = get_response(constraint_prompt.format(
    topic="artificial intelligence",
    style="technical",
    excluded_words="robot, human-like, science fiction"
))
```

```
print(response)
```

```
Artificial intelligence (AI) refers to the simulation of cognitive process
es by computer systems. This includes the ability to learn from data, reco
gnize patterns, make decisions, and perform tasks that typically require i
ntelligence. AI encompasses various subfields such as machine learning, na
tural language processing, and computer vision. Algorithms are designed to
analyze large datasets, enabling systems to improve performance over time.
AI applications range from data analysis and image recognition to autonomo
us systems and decision support tools. The development of AI involves inte
rdisciplinary techniques, including mathematics, statistics, and computer
programming.
```

4. Evaluation and Refinement

To evaluate and refine our negative prompts, we can create a function that checks if the output adheres to our constraints.

In [8]:

```
def evaluate_output(output, constraints):
    """Evaluate if the output meets the given constraints."""
    results = {}
    for constraint, check_func in constraints.items():
        results[constraint] = check_func(output)
    return results

# Define some example constraints
constraints = {
    "word_count": lambda x: len(x.split()) <= 100,
    "no_excluded_words": lambda x: all(word not in x.lower() for word in
    "no_analogies": lambda x: not re.search(r"\b(as|like)\b", x, re.IGNO

}

# Evaluate the previous output
evaluation_results = evaluate_output(response, constraints)
print("Evaluation results:", evaluation_results)

# If the output doesn't meet all constraints, we can refine our prompt
if not all(evaluation_results.values()):
    refined_prompt = constraint_prompt.format(
        topic="artificial intelligence",
        style="technical and concise",  # Added 'concise' to address wor
        excluded_words="robot, human-like, science fiction, like, as"  #
    )
    refined_response = get_response(refined_prompt)
    print("\nRefined response:\n", refined_response)

    # Evaluate the refined output
    refined_evaluation = evaluate_output(refined_response, constraints)
    print("\nRefined evaluation results:", refined_evaluation)
```

```
Evaluation results: {'word_count': True, 'no_excluded_words': True, 'no_an
alogies': False}

Refined response:
 Artificial intelligence (AI) refers to the simulation of cognitive proces
ses by computer systems. It encompasses various subfields, including machi
ne learning, natural language processing, and computer vision. AI systems
analyze data, recognize patterns, and make decisions based on algorithms.
They can perform tasks such as speech recognition, image analysis, and pre
dictive modeling. AI applications are utilized in industries such as finan
ce, healthcare, and autonomous systems, enhancing efficiency and enabling
advanced problem-solving capabilities. The development of AI relies on lar
ge datasets, computational power, and sophisticated algorithms to improve
accuracy and performance over time.

Refined evaluation results: {'word_count': True, 'no_excluded_words': Tru
e, 'no_analogies': False}
```