

## 02-Principios basicos de prompting

May 12, 2024

### 0.1 Principios básicos del prompting

1. Los prompts deben ser sencillos y claros. No es necesario escribir un párrafo entero para obtener una respuesta.
2. Dale al modelo tiempo para “pensar” en la respuesta. No le des un prompt y esperes una respuesta inmediata.

```
[ ]: import openai
import os
from IPython.display import Markdown

from dotenv import load_dotenv
load_dotenv()

openai.api_key = os.getenv("OPENAI_API_KEY")

[ ]: client = openai.OpenAI()

def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role": "user", "content": prompt}]
    response = client.chat.completions.create(
        model=model,
        messages=messages,
        temperature=0.7,
        max_tokens=256,
        top_p=1,
        frequency_penalty=0,
        presence_penalty=0
    )
    return response.choices[0].message.content
```

#### 0.1.1 Tácticas

1. Utiliza delimitadores para indicar cómo distinguir partes del input. Delimitadores pueden ser “, ””, < >, ‘, etc.

```
[ ]: # Definimos el texto
```

```

text = """En esencia, a menor temperatura en un Modelo de Lenguaje de Gran
↳Escala (LLM), los resultados tienden a ser más deterministas. Esto implica
↳que el modelo optará con más frecuencia por el siguiente token de mayor
↳probabilidad. Incrementar la temperatura introduce mayor variabilidad en las
↳respuestas, lo que promueve resultados más variados o creativos.
↳Básicamente, al elevar la temperatura, se incrementa la posibilidad de
↳elegir diferentes tokens como opciones. Desde un punto de vista práctico, en
↳tareas como preguntas y respuestas basadas en información objetiva, se
↳sugiere utilizar un valor de temperatura reducido para lograr respuestas más
↳certeras y breves. Sin embargo, en la creación de poesía o en otras
↳actividades creativas, puede resultar ventajoso aumentar el valor de la
↳temperatura. El aumento en la temperatura puede generar alucinaciones."""

# Definimos el prompt
prompt = f"Resume el texto delimitado por las triples comillas en una sola
↳oración: {text}"

# Obtenemos la respuesta
response = get_completion(prompt)
display(Markdown(f"**Respuesta:** {response}"))

```

**Respuesta:** La temperatura en un Modelo de Lenguaje de Gran Escala (LLM) influye en la determinación de los resultados, siendo menor temperatura más determinista y mayor temperatura más variabilidad y creatividad, beneficiando la elección de diferentes tokens y favoreciendo respuestas certeras en tareas objetivas con temperatura reducida, mientras que en actividades creativas como la poesía se recomienda aumentar la temperatura, aunque un incremento excesivo puede generar alucinaciones.

2. JSON, HTML, XML, YAML, etc. son formatos que el modelo puede entender y que pueden ayudar a estructurar el input.

```

[ ]: prompt = """Genera una lista de tres títulos de libros inventados junto con sus
↳autores y géneros. Proporciónelos en formato JSON con las siguientes claves:
↳book_id, título, autor, género"""
response = get_completion(prompt)
print(response)

{
  "libro1": {
    "título": "El misterio del faro oscuro",
    "autor": "Isabel Mendoza",
    "género": "Misterio"
  },
  "libro2": {
    "título": "La estrella fugaz",
    "autor": "Javier García",
    "género": "Ciencia ficción"
  },
}

```

```

"libro3": {
    "título": "El jardín secreto de las hadas",
    "autor": "María Pérez",
    "género": "Fantasía"
}
}

```

3. Pídele al modelo que verifique si se están cumpliendo ciertas condiciones.

```

[ ]: text_1 = """Preparar una exquisita taza de café en una máquina espresso es un
    ↪proceso sencillo. En primer lugar, asegúrate de que la máquina esté
    ↪encendida y lista para usar. Luego, muele los granos de café según el grosor
    ↪deseado y colócalos en el portafiltro. Compacta el café con el tampador para
    ↪garantizar una extracción uniforme. Ajusta la cantidad de café según tu
    ↪preferencia y coloca el portafiltro en la máquina. Activa la máquina y
    ↪espera a que el café se extraiga lentamente, disfrutando del aroma que llena
    ↪el aire. ¡Y voilà! Disfruta de tu aromático y delicioso café espresso."""

prompt = f"""
Vas a recibir un texto delimitado por triples comillas.
Si contiene una secuencia de instrucciones, reescribe esas instrucciones en el
    ↪siguiente formato:

Paso 1 - ...
Paso 2 - ...
...
Paso N - ...

Si el texto no contiene una secuencia de instrucciones, entonces simplemente
    ↪escribe "No se proporcionaron pasos."

Texto:
```{text_1}```
"""

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

```

Respuesta:

Paso 1 - Asegúrate de que la máquina esté encendida y lista para usar.  
Paso 2 - Muele los granos de café según el grosor deseado y colócalos en el portafiltro.  
Paso 3 - Compacta el café con el tampador para garantizar una extracción uniforme.  
Paso 4 - Ajusta la cantidad de café según tu preferencia y coloca el portafiltro en la máquina.  
Paso 5 - Activa la máquina y espera a que el café se extraiga lentamente,

disfrutando del aroma que llena el aire.

Paso 6 - ¡Y voilà! Disfruta de tu aromático y delicioso café espresso.

```
[ ]: text_2 = """El sol brilla intensamente hoy y los pájaros cantan. Es un hermoso_
    ↪ día para dar un paseo por el parque. Las flores están floreciendo y los_
    ↪ árboles se balancean suavemente con la brisa. La gente está fuera_
    ↪ disfrutando del buen tiempo. Algunos están haciendo picnics, mientras que_
    ↪ otros están jugando o simplemente relajándose en el césped. Es un día_
    ↪ perfecto para pasar tiempo al aire libre y apreciar la belleza de la_
    ↪ naturaleza."""

prompt = f"""
Te proporcionarán un texto delimitado por triples comillas.
Si contiene una secuencia de instrucciones, reescribe esas instrucciones en el_
    ↪ siguiente formato:

Paso 1 - ...
Paso 2 - ...
...
Paso N - ...

Si el texto no contiene una secuencia de instrucciones, simplemente escribe "No_
    ↪ se proporcionaron pasos."

```{text_2}```
"""

response = get_completion(prompt)
print("Resultado para Texto 2:")
print(response)
```

Resultado para Texto 2:

No se proporcionaron pasos.

4. Few-shot prompting: Utiliza ejemplos para que el modelo entienda mejor lo que quieres.

```
[ ]: prompt = """
Tu tarea es responder de manera consistente.

<hijo>: Enséñame sobre la paciencia.

<abuelo>: El río que talla el valle más profundo fluye desde una modesta fuente;
    ↪ la sinfonía más grandiosa se origina a partir de una sola nota; el tapiz_
    ↪ más intrincado comienza con un hilo solitario.

<hijo>: Enséñame sobre la resiliencia.
"""

response = get_completion(prompt)
```

```
display(Markdown(f"**Respuesta:** {response}"))
```

**Respuesta:** : La resiliencia es como el bambú, fuerte y flexible. A pesar de las tormentas y los vientos fuertes, el bambú se dobla pero no se rompe. Se adapta y se recupera, siempre encontrando la manera de seguir creciendo. Es importante aprender a ser como el bambú, resistente ante las adversidades y capaz de superar cualquier obstáculo que se presente en nuestro camino.

Principio 2: Darle tiempo al modelo para “pensar” en la respuesta.

Táctica: Especifica los pasos requeridos para completar la tarea.

```
[ ]: text = """
En un tranquilo bosque, los hermanos Mario y Sara se aventuraron en busca de un
↳ antiguo tesoro perdido. Mientras caminaban entre los árboles, riendo y
↳ compartiendo historias, un ruido extraño los detuvo de repente. MARIO,
↳ intrigado, se acercó para investigar y accidentalmente activó una trampa
↳ oculta. Una red se desplegó atrapándolo, y Sara, al intentar liberarlo, cayó
↳ en un foso cercano.

Afortunadamente, ninguno de los dos resultó gravemente herido. Juntos,
↳ encontraron la manera de salir de sus respectivas trampas y regresaron a
↳ casa con algunas raspaduras y moretones. Sin embargo, en lugar de
↳ desanimarse, se abrazaron con alivio y se rieron de su mala suerte.

Decidieron que la próxima vez serían más cuidadosos, pero su determinación por
↳ descubrir el tesoro perdido no disminuyó. Con una nueva perspectiva sobre la
↳ aventura, continuaron explorando el bosque con entusiasmo renovado,
↳ emocionados por lo que el destino les depararía.
"""

prompt_1 = f"""
Realiza las siguientes acciones:
1 - Resumir el siguiente texto delimitado por triple comillas con 1 frase.
2 - Traduzca el resumen al francés.
3 - Listar cada nombre en el resumen en francés.
4 - Generar un objeto json que contenga lo siguiente
claves: french_summary, num_names.

Separa tus respuestas con saltos de línea.

Texto:
```{text}```
"""

response = get_completion(prompt_1)
print("Respuesta para el prompt 1:")
print(response)
```

Respuesta para el prompt 1:

1 - Mario y Sara se aventuran en busca de un tesoro perdido en un bosque,

enfrentando obstáculos pero manteniendo su determinación.

2 - Mario et Sara se lancent à la recherche d'un trésor perdu dans une forêt, affrontant des obstacles mais gardant leur détermination.

3 - Mario, Sara

```
4 - {  
  "french_summary": "Mario et Sara se lancent à la recherche d'un trésor perdu  
dans une forêt, affrontant des obstacles mais gardant leur détermination.",  
  "num_names": 2  
}
```

Pide el resultado en un formato específico.

```
[ ]: prompt_2 = f"""  
tu tarea consiste en realizar las siguientes acciones:  
1 - Resumir el siguiente texto delimitado por <> con 1 frase.  
2 - Traducir el resumen al francés.  
3 - Listar cada nombre en el resumen en francés.  
4 - Dar salida a un objeto json que contenga las siguientes claves:┐  
  ↳french_summary, num_names.  
  
Utilice el siguiente formato:  
Texto: <texto a resumir>  
Resumen: <resumen>  
Traducción: <traducción del resumen>  
Nombres: <lista de nombres en resumen>  
JSON de salida: <json con resumen y número_nombres>  
  
Texto: <{text}>  
"""  
response = get_completion(prompt_2)  
print("\nRespuesta para el prompt 2:")  
print(response)
```

Respuesta para el prompt 2:

Resumen: En un bosque tranquilo, los hermanos Mario y Sara se aventuran en busca de un antiguo tesoro perdido, pero se encuentran con trampas inesperadas que los hacen más determinados que nunca.

Traducción: Dans une forêt tranquille, les frères Mario et Sara se lancent à la recherche d'un ancien trésor perdu, mais se retrouvent confrontés à des pièges inattendus qui les rendent plus déterminés que jamais.

Nombres: Mario, Sara

JSON de salida: {"french\_summary": "Dans une forêt tranquille, les frères Mario et Sara se lancent à la recherche d'un ancien trésor perdu, mais se retrouvent confrontés à des pièges inattendus qui les rendent plus déterminés que jamais.",

```
"num_names": 2}
```

Táctica 2: Pedir al modelo que elabore su propia solución antes de llegar a una conclusión precipitada.

## 0.2 Limitaciones del modelo: Alucinaciones

Apple es una compañía de tecnología real pero iCan no es un producto real de Apple. Sin embargo, el modelo puede alucinar que iCan es un producto real de Apple y devolvernos una respuesta que no es cierta.

```
[ ]: prompt = """
Háblame sobre el iCan, la última innovación en tecnología de dispositivos_
↳portátiles de Apple.
"""
response = get_completion(prompt)
display(Markdown(f"**Respuesta:** {response}"))
```

**Respuesta:** El iCan es la última innovación de Apple en tecnología de dispositivos portátiles. Este dispositivo revolucionario combina las funcionalidades de un reloj inteligente, un teléfono móvil y un asistente personal en un solo dispositivo compacto y elegante.

El iCan cuenta con una pantalla táctil de alta resolución que permite a los usuarios acceder a sus aplicaciones favoritas, recibir notificaciones en tiempo real y realizar llamadas y mensajes de texto de forma rápida y sencilla. Además, el dispositivo cuenta con un sistema de reconocimiento de voz avanzado que permite a los usuarios interactuar con el iCan de forma natural y fluida.

Otras características destacadas del iCan incluyen un sensor de ritmo cardíaco integrado, un monitor de sueño, seguimiento de actividad física y GPS incorporado. Todo esto hace que el iCan sea el compañero perfecto para aquellos que buscan estar siempre conectados y al tanto de su salud y bienestar.

En resumen, el iCan es la última palabra en tecnología de dispositivos portátiles y representa la visión futurista de Apple en cuanto a la interacción entre humanos y tecn

## 0.3 Tu turno de experimentar!

```
[ ]:
```