01-Iniciacion

May 12, 2024

1 Iniciación al Prompt Engineering

Los Modelos de Lenguaje de Gran Escala (LLM) son sistemas de inteligencia artificial diseñados para generar texto humano coherente y contextualmente relevante. Para generar el siguiente token en una secuencia, los LLM utilizan una serie de parámetros que influyen en la selección del token adecuado. Conocer y ajustar estos parámetros es crucial para obtener resultados óptimos y adaptados a diferentes contextos y tareas.

Algunos de los parámetros clave en la generación de tokens incluyen la temperatura, que determina el grado de aleatoriedad en la selección de tokens, y el Top P, que controla la probabilidad acumulada de los tokens considerados. Además, la longitud máxima y las secuencias de parada ayudan a controlar la extensión y estructura de las respuestas generadas. También existen penalizaciones por frecuencia y presencia para evitar la repetición innecesaria de palabras o frases.

Entender y ajustar estos parámetros permite a los usuarios adaptar el comportamiento del LLM a diferentes aplicaciones, como responder preguntas basadas en hechos, generar contenido creativo o realizar traducciones automáticas, entre otras. Por lo tanto, es esencial familiarizarse con estos parámetros para aprovechar al máximo las capacidades de un LLM.

1.1 El papel de los parámetros en los LLM

Temperatura (Temperature): 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.

Top P: Esta es una técnica de muestreo en conjunto con la temperatura, conocida como muestreo de núcleo, que te permite controlar cuán determinista es el modelo. Si buscas respuestas exactas y basadas en hechos, es aconsejable mantener un valor bajo de Top P. Si, por el contrario, deseas respuestas más variadas, puedes aumentar este valor. Utilizar Top P significa que solo se consideran para las respuestas los tokens que constituyen la masa de probabilidad top_p, por lo que un valor bajo de top_p seleccionará las respuestas más seguras. Un valor alto de top_p, en cambio, permitirá que el modelo considere un rango más amplio de palabras posibles, incluyendo aquellas

menos probables, lo que conduce a resultados más variados. La recomendación general es modificar la temperatura o el Top P, pero no ambos al mismo tiempo.

El **Top K** es otro parámetro importante en la generación de texto por parte de los Modelos de Lenguaje de Gran Escala (LLM). Este parámetro se refiere a la selección de los K tokens más probables como candidatos para el siguiente token en la secuencia. Al ajustar el valor de K, los usuarios pueden controlar el grado de determinismo en las respuestas generadas por el modelo. Un valor de K más pequeño resultará en respuestas más deterministas y enfocadas, mientras que un valor de K más grande permitirá una mayor diversidad y creatividad en las respuestas al considerar un rango más amplio de tokens.

La diferencia principal entre el Top P y el Top K radica en la forma en que ambos parámetros restringen el conjunto de tokens considerados para la generación del siguiente token en un Modelo de Lenguaje de Gran Escala (LLM).

Top P se basa en la probabilidad acumulada de los tokens. En lugar de seleccionar un número fijo de tokens más probables (como hace el Top K), el Top P considera todos los tokens cuya probabilidad acumulada sea igual o inferior al valor de P establecido. Esto significa que el número de tokens considerados puede variar, dependiendo de las probabilidades asignadas a cada token por el modelo.

Por otro lado, el Top K selecciona un número fijo de tokens más probables (K) como candidatos para el siguiente token. Al ajustar el valor de K, se limita el conjunto de tokens candidatos a los K tokens con mayor probabilidad.

Longitud Máxima(Max Length): Puedes gestionar el número de tokens que el modelo genera ajustando la longitud máxima. Especificar una longitud máxima te ayuda a evitar respuestas demasiado largas o irrelevantes y a controlar los costos asociados.

Secuencias de Parada(Stop Sequences): Una secuencia de parada es una cadena de texto que detiene la generación de tokens por parte del modelo. Especificar secuencias de parada es otra manera de controlar la longitud y estructura de las respuestas del modelo. Por ejemplo, puedes indicar al modelo que genere listas que no tengan más de 10 elementos añadiendo “11” como secuencia de parada.

Penalización por Frecuencia(Frequency Penalty): Esta penalización se aplica al siguiente token proporcionalmente a la cantidad de veces que ese token ya ha aparecido en la respuesta y en la indicación. Cuanto mayor sea la penalización por frecuencia, menos probable será que una palabra aparezca de nuevo. Esta configuración reduce la repetición de palabras en la respuesta del modelo al aplicar una penalización mayor a los tokens que aparecen con más frecuencia.

Penalización por Presencia(Presence Penalty): Al igual que la penalización por frecuencia, la penalización por presencia se aplica a los tokens repetidos, pero a diferencia de la primera, la penalización es la misma para todos los tokens repetidos, independientemente de la frecuencia de su aparición. Esto evita que el modelo repita frases con demasiada frecuencia en su respuesta. Si deseas que el modelo genere texto diverso o creativo, podrías optar por usar una penalización por presencia más alta. Por otro lado, si necesitas que el modelo mantenga el enfoque, podrías probar con una penalización por presencia más baja.

Al igual que con la temperatura y el Top P, la recomendación general es modificar la penalización por frecuencia o por presencia, pero no ambas al mismo tiempo.

1.2 Setup del entorno y de la apikey de OpenAI

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

from dotenv import load_dotenv
load_dotenv()

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

Durante todo el curso vamos a utilizar GPT-3.5-turbo de OpenAI. Para poder utilizarlo necesitamos una apikey que se obtiene a través de la web de OpenAI y que vamos a guardar en un archivo .env. Para ello, vamos a instalar la librería python-dotenv que nos permitirá cargar las variables de entorno desde un archivo .env. Dentro de ese archivo .env vamos a guardar la apikey de OpenAI de esta forma: OPENAI_API_KEY = nuestra_apikey

Vamos a crear también una función que nos ayude a la hora de usar los prompts para que sea más sencillo.

[ ]:
```python
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=1,
        max_tokens=256,
        top_p=1,
        frequency_penalty=0,
        presence_penalty=0
    )
    return response.choices[0].message.content
```

[ ]:
```python
# Definimos el prompt
prompt = f"Hola, qué puedes hacer por mí?"

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

Respuesta: ¡Hola! Puedo ayudarte con información, responder preguntas, ofrecerte consejos, jugar juegos, contar chistes y mucho más. ¿En qué puedo ayudarte hoy?