

Justificación de la Práctica Final IA3

Objetivo del desarrollo

El objetivo de la práctica es automatizar las respuestas a los emails de solicitud de devolución recibidos por la empresa Componentes Intergalácticos Industriales S.A., identificando el motivo de la solicitud, determinando si procede o no la devolución según criterios definidos en el prompt, y generando una respuesta formal con firma personalizada.

Estructura del flujo de trabajo

El flujo está compuesto por 3 pasos principales, tal como se requería en el enunciado:

1. Extracción: Se utiliza `PromptTemplate` junto con `LLMChain` para extraer del email el número de pedido, nombre del remitente y motivo principal de la solicitud. Se incluye una instrucción para evitar que el modelo devuelva la información en formato Markdown.
2. Evaluación: Se define un segundo `LLMChain` con un prompt que contiene las condiciones explícitas para aceptar o rechazar una devolución. Se espera que el modelo responda exclusivamente con 'ACEPTAR' o 'RECHAZAR'.
3. Generación de respuesta: Se utiliza una tercera `LLMChain` que genera una respuesta formal personalizada, adaptada a la decisión tomada. Se parametriza también la firma profesional mediante variables.

Justificación del uso de herramientas

- ◆ Se ha optado por utilizar `LLMChain` y `PromptTemplate` en lugar de `SimpleSequentialChain`, ya que cada paso del flujo requiere manipular múltiples variables de entrada y salida. `SimpleSequentialChain` solo permite una entrada y una salida por paso, lo que limita la flexibilidad necesaria en este caso.
- ◆ No se ha utilizado `HuggingFaceHub` ni modelos de Hugging Face porque el caso requiere precisión lingüística y contexto profundo, y `ChatOpenAI (gpt-4o)` ofrece mejores resultados para entender matices de correos formales y redactar respuestas empáticas. Además, su integración con LangChain es más directa para este tipo de flujos.
- ◆ Se ha implementado la función `crear_cadena()` para construir dinámicamente cada prompt y evitar repetición innecesaria de código.
- ◆ También se ha añadido `convertir_a_diccionario()`, una función para convertir el texto generado por el modelo en un diccionario de Python. Esta función limpia posibles etiquetas Markdown no deseadas y asegura un procesamiento robusto.

Casos implementados

Se han implementado dos escenarios diferenciados:

- Caso 1: Solicitud RECHAZADA por daños durante el transporte (Darth Márquez)
- Caso 2: Solicitud ACEPTADA por defecto de fábrica (Lucía Robles)

Dependencias requeridas

pip install langchain langchain-openai python-dotenv openai