Uso de Embeddings y Fine-Tuning con Ollama

Aplicación práctica en temas de aborto y eutanasia

Profesor: Alcaraz Chavez Jesus Eduardo

Alumno: Solis Rosales Roberto Miguel

Junio 2025

1. Instalación de Ollama

Primero se instaló la herramienta Ollama, la cual permite realizar preguntas y obtener respuestas de modelos avanzados de lenguaje.

2. Consulta de temas

Se realizaron preguntas a Ollama sobre dos temas centrales: el aborto y la eutanasia. Estas respuestas se almacenaron para su posterior análisis.

3. Almacenamiento de respuestas

Todas las respuestas obtenidas se guardaron en un documento de texto como base para el análisis posterior.

4. Generación de Embeddings

Se convirtieron las respuestas o fragmentos de texto en vectores numéricos mediante la librería sentence-transformers. Esto permite a una máquina entender el significado del texto, facilitando la búsqueda semántica y la agrupación de ideas similares.

Para esto se:
- Instala Python y la librería sentence-transformers.
- Escriben las respuestas en una lista (10 respuestas, cada una entre comillas y separadas por comas).
- Se carga un modelo preentrenado para generar los embeddings.
- Se guardan los embeddings generados en un archivo JSON.

```
from sentence_transformers import SentenceTransformer
import json

# Aquí pones tus respuestas (cada una es un texto)
respuestas = [

    "Respuesta 1: ¡Es importante destacar que en este contexto, hablamos de situaciones

    "Respuesta 2: ¡Es cierto que el lenguaje utilizado para describir los procedimientos

    """Respuesta 3: ¡El aborto es un tema complejo que puede verse desde diferentes pers
    * Utilitarismo: Si se evalúan las consecuencias de cada acción, el principio del uti
    * Deontología: Si se evalúa el acto en sí mismo y no solo sus consecuencias, el prin
    * Ética del cuidado: Si se evalúa la relación entre el aborto y la provisión de cuid
    Es importante recordar que estos principios son solo algunas de las maneras en que e
```

```python
# Carga modelo pre-entrenado para embeddings
modelo = SentenceTransformer('all-MiniLM-L6-v2')

# Genera los embeddings para cada respuesta
embeddings = modelo.encode(respuestas)

# Guarda los embeddings en un archivo JSON para usarlos después
with open('embeddings.json', 'w') as f:
    json.dump(embeddings.tolist(), f)

print("Embeddings generados y guardados en 'embeddings.json'")
```

Por ultimo cargamos el modelo ya entrenado, creando así los embeddings para cada respuesta y guardando los embeddings en un archivo JSON.

Como resultado tenemos un archivo con los vectores numéricos de cada respuesta, con estos podemos comparar similitudes, búsquedas o alimentar sistemas mas avanzados

5. Fine-Tuning con Ollama

El fine-tuning consiste en entrenar el modelo con datos personalizados (preguntas, respuestas y contexto) para mejorar su desempeño en temas específicos. Se necesitan archivos JSON con ejemplos adecuados. Luego, se usa la función de fine-tuning y se carga el modelo personalizado.

6. Código ejemplo de embeddings

Este código configura un entorno básico para convertir oraciones en vectores numéricos, útil para tareas como búsqueda semántica, clasificación o detección de similitud entre textos:

```python
from sentence_transformers import SentenceTransformer
import faiss
import numpy as np

# Inicializa el modelo de embeddings
modelo = SentenceTransformer('all-MiniLM-L6-v2')
```

Explicación:
- `from sentence_transformers import SentenceTransformer`: Importa la clase principal para crear embeddings.
- `import faiss`: Permite realizar búsquedas rápidas basadas en similitud de vectores.
- `import numpy as np`: Para el manejo de datos numéricos en forma de matrices y vectores.

7. Ventajas del uso de embeddings

Búsqueda semántica: permite encontrar respuestas similares a preguntas aunque se formulen de manera diferente.

Agrupación de ideas: se pueden clasificar respuestas similares sin intervención humana.

Reducción de ambigüedad: mejora la comprensión del modelo sobre el significado real del texto.

8. Retos y consideraciones

Sesgo en los datos: si el modelo se entrena solo con respuestas parciales, puede reforzar opiniones.

Formato de datos: es importante mantener una estructura JSON clara y coherente en el fine-tuning.

Costo computacional: entrenar modelos personalizados requiere recursos de cómputo potentes.

9. Conclusión

El uso de embeddings y fine-tuning con Ollama ha permitido optimizar el análisis de respuestas sobre temas delicados como el aborto y la eutanasia. Gracias a esto, es posible construir sistemas de recomendación, clasificación y búsqueda que sean más precisos y contextualmente informados.