

AI mit Oracle

Wir bauen einen RAG-basierten Chat

Teil 3.3 – RAG & Chat mit KI: Integration in den Chat

Was bisher geschah...

- Teil 1: Kommunikation mit OpenAI-API via UTL_HTTP
- Teil 2: Chat mit seiner eigenen Historie als Wissensbasis
- Teil 3.1: Dokumentenaufbereitung und Chunking als Grundlage für RAG
- Teil 3.2: Vektorisierung von Texten zur Implementation einer semantischen Suche

Was kommt jetzt?

- Wir bauen alles zusammen für einen RAG-basierten Chat
- Nutzung des PL/SQL-Packages als Backend für eine APEX-Anwendung

RAG – Kontextunterstützung des Chats

- Grundsätzliche Vorüberlegungen
 - Zu übertragender Dokument-Content des Chat
 - Datenstruktur: Chat←Historie←RAG-Content
- Implementation des Messages-Array der JSON-API (für ChatGPT)
 1. Role „system“: grundsätzliche Prompts „You are a helpful assistant...“
 2. Role „assistant“: Dokument-Content (RAG)
 3. Roles „user“ und „assistant“ jew. paarweise: Chat-Historie
 4. Role „user“: aktueller Prompt
- Demo
 - PL/SQL-Package als Backend für eine APEX-Anwendung

RAG – Kontextunterstützung des Chats

- **Problematik Datenschutz**
 - **Lösungsmöglichkeiten:**
 1. **Lokales LLM**
 2. **Pseudonymisierung (umkehrbar), z.B. mit Microsoft Presidio**
- **Fazit**
 - **Struktur funktioniert für Detailabfragen, andere Implementation für z.B. Zusammenfassungen nötig**
 - **Datenschutzaspekte sind zu beachten!**

RAG – Kontextunterstützung des Chats

Ausblick

- Implementation eines Pseudonymisierers (Rest_Service)
- Implementation eines Generators für SQL-Abfragen für interaktive Reporting-Erstellung
- Implementation von Tools = Funktionen, deren Aufruf durch das LLM gesteuert wird



Graef Computer GmbH · Fallgatter 5 · 44369 Dortmund
Telefon: +49 231 222 429 - 99 · info@graef.com