

Enhancing LLM Prompt Engineering through Automated Prompt Management

Mit der zunehmenden Nutzung großer Sprachmodelle (LLMs) wird effektives Prompt-Management immer wichtiger, da gut gestaltete Prompts die Leistung erheblich verbessern können. Prompt-Engineering ist jedoch zeitaufwendig und schwer zu organisieren. Unser entwickeltes Tool unterstützt diese Prozesse, verbessert so die Effizienz und steigert die Leistung der Modelle.

In unserem Projekt haben wir ein System entwickelt, das das Management und die Optimierung von Prompts für große Sprachmodelle (LLMs) automatisiert. Dies soll die Effizienz und Leistung der Modelle verbessern. Die Aufgaben umfassen die Erstellung und Implementierung eines Systems zur Speicherung, Organisation und Versionierung von LLM-Prompts sowie die Entwicklung von Methoden zur Analyse der Effektivität der Prompts und zur Verbesserung basierend auf diesen Leistungskennzahlen.

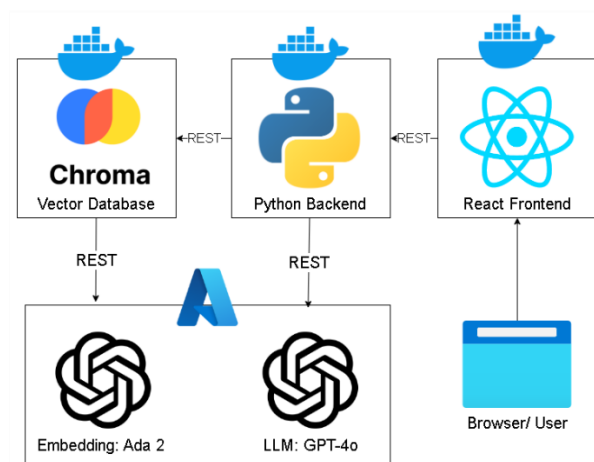


Abbildung 1: Architektur

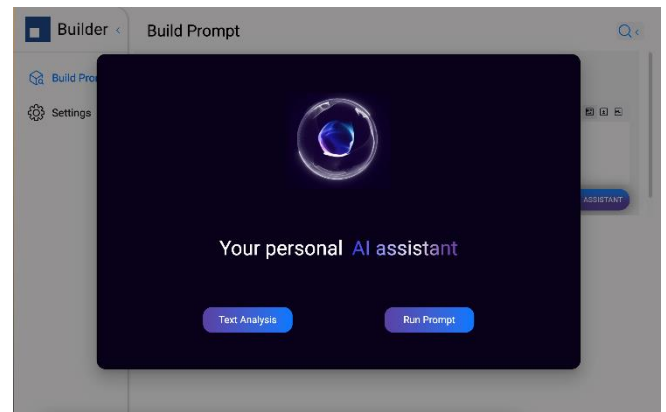


Abbildung 2: GUI Interface

Die Projektarchitektur besteht aus mehreren Schlüsselkomponenten. Der Nutzer greift über die GUI im Browser auf das System zu. Diese Benutzeroberfläche wird mithilfe von React Remix in einem Docker-Container bereitgestellt. Das Frontend kommuniziert über eine klar definierte REST-Schnittstelle, die eine Circuit Breaker Methodologie implementiert, mit dem in Python geschriebenen Backend. Über diese Schnittstelle können Prompts gespeichert (automatische Versionierung), mithilfe von Filtern gesucht oder anhand eines Suchtextes über Vektorsuche nach ähnlichen Prompts durchsucht werden. Die Prompts werden in einem weiteren Container von Chroma DB verwaltet, das für die Embeddings der Prompts das auf Azure gehostete OpenAI Embedding Modell Ada 2 verwendet.

Neben den Funktionen für das Promptmanagement bietet das System auch Möglichkeiten, Prompts durch ein LLM auszuführen, zu evaluieren und automatisch zu verbessern. Bei der Evaluation werden die Metriken Klarheit, Vollständigkeit und Struktur LLM-basiert bewertet, und es werden Tipps zur Verbesserung des Prompts gegeben. Diese Tipps werden bei der automatisierten Prompt-Verbesserung genutzt, um den Prompt mithilfe eines LLM zu optimieren.

Team: David Pospisil, Marius Riesle, Max Andris, Tobias Bohun, Tobias Moser