DHBW Stuttgart, Anwendungsaspekte des Machine Learning, WWI2022A/B

Assignment: Retrieval-Augmented Generation (RAG) und Agentensysteme

Zielsetzung

In diesem Assignment entwickeln Sie zwei eigenständige Systeme:

1. Ein **Retrieval-Augmented Generation (RAG)**-System mit einer selbstgewählten Wissensquelle
2. Ein **Agentensystem** mit erweiterten Fähigkeiten wie Tool-Nutzung, Routing und Orchestrierung

Sie können frei entscheiden, welche Frameworks oder Modelle Sie verwenden – ob Open Source (z. B. Haystack, LangChain, LlamaIndex, etc.) oder ein reines Python-Skript. Ziel ist es, sowohl die Konzepte zu verstehen als auch deren praktische Umsetzung zu beherrschen.

Teil 1: RAG-System

Anforderungen:

- Wählen Sie eine externe Wissensquelle:
  - PDF (max. **10 Seiten**) oder
  - Word-Dokument (max. **10 Seiten**) oder
  - Transkript eines YouTube-Videos (max. **15 Minuten Laufzeit**)
- Integrieren Sie ein Retrieval-System (z. B. Chunks + Embeddings)
- Verwenden Sie eine OpenAI- oder Open-Source-Sprachmodell-API, um Fragen zur Wissensquelle zu beantworten

Hinweise zur API-Nutzung (OpenAI):

- Verwenden Sie **sparsam Tokens** – vermeiden Sie große Prompts oder überlange Chunks
- Nutzen Sie bei OpenAI bevorzugt:
  - `gpt-3.5-turbo` (kostengünstig)
  - `text-embedding-3-small` (für Embeddings)
- Chunk-Größe: max. 300–500 Tokens
- Optional: evaluieren Sie verschiedene Modelle hinsichtlich Kosten & Performance

Teil 2: Agentensystem

Anforderungen:

Erstellen Sie ein System, das **mindestens fünf** der folgenden Fähigkeiten implementiert:

1. **Structured Output**: z. B. Ausgabe in JSON, Tabelle oder Datenbank
2. **Tool Use**: z. B. Zugriff auf Rechner, API oder Dateisystem
3. **Retrieval**: Integration externer Informationen wie in Teil 1
4. **Prompt Chaining**: sequentielle Verarbeitung mehrerer Prompts
5. **Routing**: dynamische Auswahl eines Agenten oder Moduls basierend auf Input
6. **Parallelization**: parallele Abfragen von LLMs oder Tools
7. **Orchestration**: Koordination mehrerer Schritte (z. B. mit einem Workflow-Manager)

Optional:

- Nutzung von Frameworks wie LangChain, CrewAI, AutoGen, oder pure Python
- Vergleich der Performance verschiedener Agentenansätze