Retrieval-Augmented Generation (RAG) und Agentensysteme – Theorie, Praxis und Transfer

Assignment
DHBW Stuttgart
Modul: Anwendungsaspekte des Machine Learning
Studiengruppe: WWI2022A/B
Abgabe: 15. August 2025, 23:59 Uhr
Einzelarbeit, Umfang ca. 6–10 Seiten + Code
Abgabeformat: frei wählbar, z.B. PDF-Dokument + lauffähiger Code (z. B. GitHub-Link oder ZIP), oder alles als ein Dokument
Selbstständigkeitserklärung: Bitte fügen Sie am Ende des Dokuments eine unterschriebene Erklärung hinzu, dass Sie die Arbeit selbstständig verfasst haben.

Zielsetzung

Dieses Assignment vertieft zwei zentrale Konzepte moderner KI-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 verbinden eigenständige Recherche, theoretische Einordnung, praktische Implementierung und kritische Reflexion.
Im Fokus steht, dass Sie nicht nur funktionierende Prototypen entwickeln, sondern auch die dahinterliegenden Konzepte verstehen, vergleichen und reflektieren. Besonders bei Agentensystemen ist eine vorgelagerte Analyse- und Konzeptionsphase vorgesehen.

Aufgabenübersicht

Sie bearbeiten **drei verpflichtende Teile**:

1. **Theoretische Recherche & Einordnung** (RAG & Agentensysteme)
2. **Implementierung eines RAG-Systems + Agentensystem**
3. **Transferanalyse und Reflexion**

Teil 1: Theoretische Recherche & Einordnung

1.1 RAG-Systeme (ca. 2-3 Seiten)

Recherchieren und beschreiben Sie:

- Die Architektur eines typischen RAG-Systems
- Unterschiede zu reinen generativen Modellen und klassischen QA-Systemen
- Die Rolle von:
  - **Embeddings & Vektordatenbanken**
  - **Prompting & Kontextkonstruktion**
  - **Modellwahl und Kostenaspekten**
- Typische Herausforderungen bei der Umsetzung (z. B. Kontextfenster, Latenz, Retrieval-Qualität)
- **Optional:** Bewertung und Vergleich zweier Open-Source-RAG-Frameworks (z. B. LangChain, LlamaIndex, Haystack)

Verwenden Sie mindestens **3 zitierfähige Quellen**, z. B. wissenschaftliche Paper, Entwicklerblogs oder offizielle Framework-Dokumentationen. Zitieren Sie sauber (APA, IEEE oder Fußnoten).

1.2 Agentensysteme (ca. 2-3 Seiten)

Recherchieren Sie aktuelle Ansätze zur Entwicklung von Agentensystemen mit LLMs.

Beschreiben Sie:

- Was ist ein Agent in diesem Kontext?
  Was unterscheidet ihn von einer einfachen Prompt-Chain? (z. B. Entscheidungslogik, Tool-Integration, Memory)
- Zentrale Fähigkeiten von Agentensystemen:
  - Tool-Nutzung (z. B. API-Zugriff, Dateizugriff, Rechner)
  - Orchestrierung (Planung/Koordination mehrerer Schritte)
  - Routing (Aufgabenverteilung an spezialisierte Subagenten)
  - Memory, Planning, Reaktion auf Nutzerkontext
- Typische Frameworks (z. B. LangChain Agents, AutoGen, CrewAI)

Vergleichen Sie zwei dieser Frameworks anhand von:

- Architektur
- Modularität
- Typischer Use Cases
- Technischer Einstiegshürden (z. B. Installation, Lernkurve, Dokumentation)

Verwenden Sie mindestens **3–4 aktuelle Quellen**, darunter **mindestens eine nicht-kommerzielle/wissenschaftliche Publikation** (z. B. Whitepaper, ArXiv, Konferenzpapier).

Teil 2: Implementierung eines KI-Systems mit RAG + Agentenkomponenten

2.1 RAG-System

Erstellen Sie ein funktionierendes Retrieval-Augmented Generation-System.

- **Wissensquelle:**
  - PDF- oder Word-Datei (max. 10 Seiten) **oder**
  - YouTube-Transkript (max. 15 Minuten)
- **Pipeline:**
  - Chunking → Embedding → semantisches Retrieval → LLM-Antwort
- **Tools:** frei wählbar (z. B. LangChain, Haystack, LlamaIndex oder eigene Implementierung)
- **Dokumentation:**
  - Architektur (inkl. **Diagramm**)
  - Begründung der Toolwahl
  - Modellkonfiguration (LLM, Embedding-Modell, Tokenlimits etc.)
  - Optional: Besonderheiten bei Datenvorverarbeitung

Achten Sie auf gute Codekommentare. Alle Abstrahierungen über Frameworks müssen erklärt werden, sprich:
Was wurde wegabstrahiert und wie funktioniert das?

2.2 Agentensystem

Bauen Sie Agentensysteme mit mindestens vier der sechs Komponenten:

- Tool-Nutzung (z. B. Rechner, Wetter-API, Dateizugriff)
- Prompt-Chaining (mehrstufige Aufgaben)
- Orchestrierung (Koordination von Teilaufgaben)
- Routing (verschiedene Aufgaben → verschiedene Subagenten)
- Structured Output (z. B. Ausgabe als Tabelle oder JSON)
- Parallelisierung (z. B. gleichzeitige Abfragen)

Die Agentenkomponenten **dürfen mit dem RAG-System interagieren** (z. B. Tool-Nutzung basierend auf abgerufenen Informationen).

Teil 3: Transferanalyse und Reflexion (ca. 2-3 Seiten)

Wählen Sie eine konkrete Anwendung Ihrer Wahl, z. B.:

- Nachhilfe- oder Lern-Chatbot
- Internes Wissenssystem in einer Firma
- Beratungstool (z. B. Recht, Finanzen, Medizin)

Reflektieren Sie:

- **Wie müsste Ihr System angepasst werden**, um in dieser Domäne produktiv eingesetzt zu werden?
  (z. B. hinsichtlich Datenbasis, Antwortgeschwindigkeit, Nutzerführung)
- **Welche ethischen und rechtlichen Fragestellungen** ergeben sich?
  (z. B. Datenschutz, Halluzinationen, Verantwortung für Handlungen)
- **Welche technischen oder konzeptionellen Grenzen** sehen Sie beim Einsatz von RAG + Agenten in Ihrer gewählten Domäne?