
How to Build a Chatbot On- & Offline (Fast)

Projektteam: Boleslav Glavatki, Yolanda Sommerfeldt, Sophie Thümmeler
Projektorganisation: Christian Kandels

Exploration: KI in Lehre und Forschung

Ziele

- Anwendungsfelder für genKI-Tools identifizieren
- praktische Einsatzmöglichkeiten erforschen und diskutieren
- Leitfaden für den Einsatz von genKI-Tools

Kontext

- Hype um genKI
- Veränderung von Lernpraktiken
- Warum selbst schreiben, wenn man prompten kann?
- Prüfungsformen auf den Prüfstand stellen

Projektgruppen

#1 Nutzung von KI-Tools in der qualitativen Sozialforschung

#2 genKI löst Aufgaben aus der Lehre: Mathematik

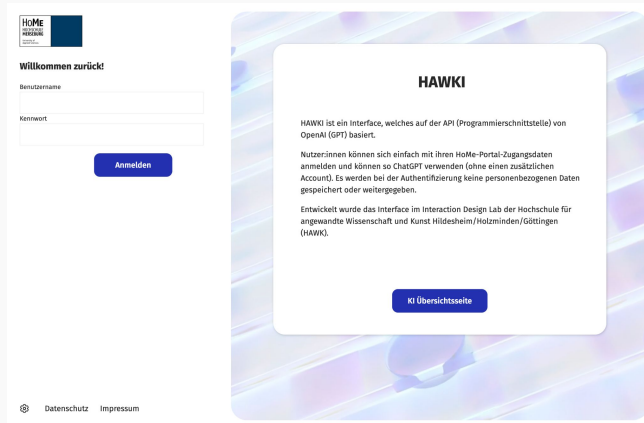
#3: Chatbots (ChatGPT vs. Open Source)

Sonstiges

hs-merseburg.de/sl2/ki

hs-merseburg.de/sl2/ki-pruefung

hs-merseburg.de/sl2/hawki



Ergebnisse

hs-merseburg.de/sl2/ki-exploration

02

Erstellung eines Chatbots mit ChatGPT Playgrounds

Yolanda Sommerfeldt

ChatGPT PLAYGROUNDS

- Veröffentlichung während unserer Projektarbeit
 - Enorme Erleichterung nach Versuch mit FineTuning von ChatGPT zu arbeiten

The screenshot displays the OpenAI Playground interface. At the top, it shows 'Personal' and 'Default project'. The main area is titled 'Assistants' and features a chat window for 'Chatbot-Assistant SL2 Kopie'. The chat history shows a user asking 'Was ist SL2?' and the assistant responding with a detailed explanation of SL² at Hochschule Merseburg. The interface includes a sidebar with various tools like 'File search', 'Code interpreter', and 'Vector store'. A 'Run' button is visible at the bottom of the chat input area.

Personal / Default project Playground Dashboard Docs API reference

Assistants Learn more

Chatbot-Assistant SL2 Kopie

THREAD thread_Y1BC2beCnwI1VOEYcZ0YqKus 686 tokens

Name: Chatbot-Assistant SL2 Kopie
asst_xNwCq4u1bByjkEFJQzrx5uvQ0

System instructions: Verständnis und verweise auf die entsprechenden Stellen. Bitte erwähne zu Beginn jeder Antwort, dass du ein Chatbot speziell für SL² an der Hochschule Merseburg bist.

Model: gpt-4-turbo

TOOLS: File search, Code interpreter, Vector store for Chatbot-Assistant SL2 Kop...

User: Was ist SL²?

Chatbot-Assistant SL2 Kopie: Ich bin ein Chatbot speziell für SL² an der Hochschule Merseburg. SL² steht für die Stärkung des Lehrens und Lernens. Es ist ein Team, das darauf fokussiert ist, die pädagogische Qualität und Effektivität im Bereich Lehre an der Hochschule zu verbessern und zu unterstützen. Das Hauptziel ist es, Lehrmethoden zu optimieren und Lernressourcen bereitzustellen, um sowohl Dozenten als auch Studenten in ihrem akademischen Alltag zu assistieren. Falls Sie spezifische Fragen oder Themen haben, die im Rahmen von SL² behandelt werden, stehe ich Ihnen gerne zur Verfügung.

Enter your message... Run

Playground messages can be viewed by anyone at your organization using the API.

SYSTEM INSTRUCTION



KRITERIEN FÜR EINE GUTE SYSTEM INSTRUCTION:

1. **Ziel und Zweck definieren**
 - Beschreibung des Assistenten: Erkläre, was der Assistent tun soll, wer die Zielgruppe ist und was der Bot abdeckt und was nicht
2. Sprachstil und Tonalität festlegen
 - Formalität: Soll der Bot formell, informell oder neutral kommunizieren?
 - **Empathie: hilfreiche, höfliche und unterstützende Antworten**
3. **Klarheit und Präzision in den Anweisungen**
 - Spezifische Aufgaben: Nenne explizit, welche Funktionen der Bot übernehmen soll
 - Handlungsanweisungen: Beschreibe, wie der Bot mit unklaren oder irrelevanten Fragen umgehen soll

System instruction

Du bist ein **hilfreicher Assistent** und beantwortest **nur Fragen zu SL²**, einem **Team der Hochschule Merseburg zur Stärkung des Lehrens und Lernens.** (...)

Antworte nur, wenn du dir sicher bist, ansonsten **verweise höflich darauf, dass ein anderes Team an der Hochschule weiterhelfen kann.**

SYSTEM INSTRUCTION



4. **Einschränkungen und Sicherheitsmechanismen**
 - Keine Spekulationen: Gib an, dass der Bot nur antworten soll, wenn er sich sicher ist
5. **Anpassung an spezifische Inhalte**
 - **Datensatz prüfen: Wechsle bei Bedarf zwischen präzisen Antworten und Verweisen auf andere Quellen.**

System instruction

(...)Wenn du eine Anfrage erhältst, frage bitte nach, ob du die hochgeladenen Dateien zum in der Anfrage enthaltenen Thema durchsuchen sollst. **Nutze ausschließlich die hochgeladenen Daten, um die Informationen zur Beantwortung der Fragen zu erhalten.** Antworte nur, wenn du dir sicher bist, ansonsten verweise höflich darauf, dass ein anderes Team an der Hochschule weiterhelfen kann.

SYSTEM INSTRUCTION



Edit system instructions

✦ Generate

Du bist ein hilfreicher Assistent und beantwortest nur Fragen zu SL², einem Team der Hochschule Merseburg zur Stärkung des Lehrens und Lernens. Wenn du eine Anfrage erhältst, frage bitte nach, ob du die hochgeladenen Dateien zum in der Anfrage enthaltenen Thema durchsuchen sollst. Nutze ausschließlich die hochgeladenen Daten, um die Informationen zur Beantwortung der Fragen zu erhalten. Antworte nur, wenn du dir sicher bist, ansonsten verweise höflich darauf, dass ein anderes Team an der Hochschule weiterhelfen kann. Bitte gib immer an, dass es sich um einen Chatbot für SL² handelt.

Außerdem stellst auf Nachfrage auch PDF-Formulare bereit. Die verfügbaren Formulare sind in der Datei `pdf_data.json` gespeichert.

- Wenn ein Benutzer nach einem Formular fragt, verwende die Funktion `findPdfLink`, um in der Datei nach dem passenden `formName` zu suchen.
- Gib den `link` sowie in pdfs steht` und die `description` des Formulars zurück.
- Falls kein passendes Formular gefunden wird, antworte höflich und schlage vor, dass der Benutzer den Namen des Formulars überprüft oder nach einer Liste aller verfügbaren Formulare fragt.

Wenn du einen Link zu einem Formular zurückgibst, stelle sicher, dass du den tatsächlichen Link aus der JSON-Datei verwendest und keine Beschreibung oder Platzhalter einfügst. Verwende das folgende Format:

Hier ist der Link zum [formName]:
[Formular herunterladen]([link])

Ersetze [formName] durch den tatsächlichen Namen des Formulars und [link] durch die URL, die in der JSON-Datei steht.

Wenn du einen Link zu einem Formular bereitstellst, stelle sicher, dass das HTML korrekt ist. Verwende immer das folgende Format:

<p>Hier muss das url</p>

Cancel

Save

Vector store for Chatbot-Assistant
SL2 Kopie

ID vs_6NZZBowcKp31tQSSErce5xLx

Expiration policy Never

Das Projekt SL ² _ Hochschule... .json	6.12.2024, 12:18	
Exploration_KI_ Hochschulejson	6.12.2024, 12:18	
E-Maker-Programm_ Hochsc... .json	6.12.2024, 12:18	
Studiengangskonferenzen_json	6.12.2024, 12:18	
Evaluationen an der Hochsch... .json	6.12.2024, 12:18	
Tag des Lehrens und des For... .json	6.12.2024, 12:18	
Lehre und Lernen im Blick_ H... .json	6.12.2024, 12:18	
Hochschul- und Mediendid... .json	6.12.2024, 12:18	
KI-Anwendungen_ Hochschu... .json	6.12.2024, 12:18	

Load more

+ Add

Detach from assistant Close

FILES

VORBEREITUNG:

- Umwandeln der Daten ins JSON Format
- Säubern der Daten (Entfernen von Klarnamen, etc.)

DURCHFÜHRUNG

- Upload der Daten im JSON Format

Assistants

Chatbot-Assistant SL2 Kopie

Model

gpt-4-turbo

TOOLS

File search + Files

Vector store for Chatbot-Assistant SL2 Kop... 168 KB
vs_6NZZBowcKp31tQSSErce5xLx

FUNCTIONS

Edit function

The model will intelligently decide to call functions based on input it receives from the user. [Learn more.](#)

Definition

 Generate Examples ▾

```
{
  "name": "addPdfFile",
  "description": "Diese Funktion gibt auf Anfrage den Link zu einer PDF-Datei zurück",
  "strict": false,
  "parameters": {
    "fileName": "Teilnahmeantrag_E-Maker_Programm_20240112.pdf",
    "filePath": "https://www.hs-merseburg.de/fileadmin/Hochschule/Projekte/SL2/E-Maker_Programm/Teilnahmeantrag_E-Maker_Programm_20240112.pdf",
    "category": "Teilnahme",
    "tags": [
      "Teilnahmeantrag",
      "Antrag",
      "Teilnahme",
      "E-Maker"
    ]
  },
  "type": "object",
  "properties": {},
  "required": []
}
```

Add "strict": true to ensure the model's response always follows this schema.



Cancel

Save

- Etablierung einer Funktion, die ermöglicht auf Nachfrage PDFs auszugeben

Assistants

 Chatbot-Assistant SL2 Kopie ▾

Code interpreter ⓘ

+ Files

Functions ⓘ

+ Functions

{f} addPdfFile

03

Erstellung eines Chatbots mit dem RAG-Ansatz

Boleslav Glavatki

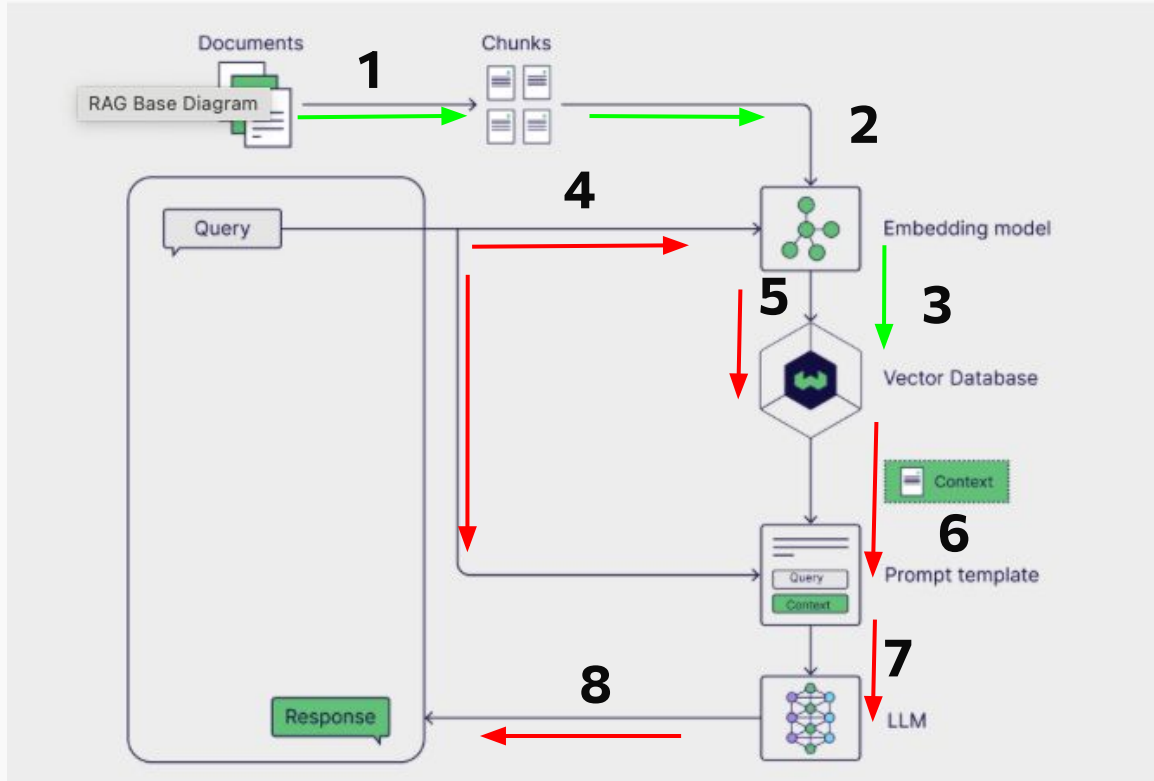
RAG

Retrieval-Augmented Generation (Abrufe-Gestützte Generierung)



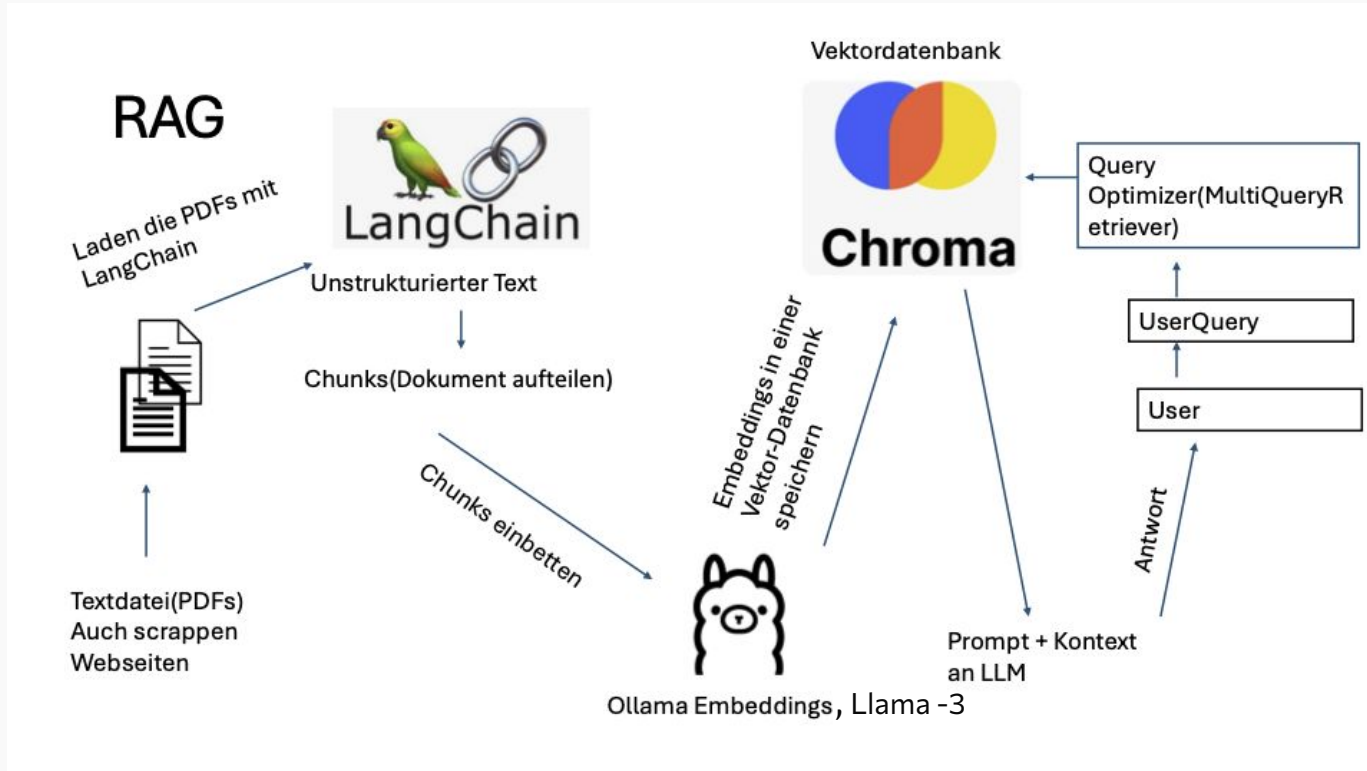
- **Kombination von Retrieval und Generierung (Abruf relevanter Informationen aus externen Wissensquellen (Datenbanken, Dokumente, APIs)).**
 - Retriever-Modul: Sucht und extrahiert relevante Daten
 - Generative Modell: Nutzt diese Daten als Kontext für die Antwortgenerierung.
- **Vorteile**
 - Erweiterung des Wissens ohne erneutes Training des Modells.
 - Aktuelle und faktenbasierte Antworten, da die Daten aus aktuellen Quellen stammen können.
 - Reduzierung von Halluzinationen des Modells.

RAG



1. **Dokumente werden in Chunks aufgeteilt**
2. **Embedding-Modell erstellt Vektoren**
3. **Speicherung in der Vektor-Datenbank**
4. **Abfrage (Query) wird verarbeitet**
5. **Abruf von relevanten Kontextinformationen**
6. **Prompt-Erstellung**
7. **Verarbeitung durch das LLM**
8. **Antwort**

RAG CODE-FIRST-ANSATZ



Bildquellen:
[1*MVJZLfszGGNiJ-UFK4U31A.png](#)
[Ollama.png](#)
[1*d2XUNgrLw7687CDfXx9-Dw.png](#)

RAG CODE-FIRST-ANSATZ



Benutzte Frameworks, Datenbanken und Programmiersprachen:

Frameworks:

- **Flask:** Web-Framework zur Erstellung der API und Benutzeroberfläche.
- **LangChain:** Zur Verarbeitung von Text und Generierung von Prompts.
- **Ollama:** Für Embeddings und Modellaufrufe.(Llama 3, nomic-embed-text:latest)

Datenbanken:

- **Chroma:** Vektor-Datenbank zur Speicherung und Suche von Embeddings.

Programmiersprachen:

- **Python:** Hauptsprache für die Backend-Logik (Datenverarbeitung, Anfragebearbeitung).
- **HTML/CSS/JavaScript:** Für die Benutzeroberfläche und Interaktionen (z. B. Formular für Datei-Uploads, Anzeige der Antworten).

Hardwareanforderungen

Llama 3 8B-Modell:

- **Arbeitsspeicher (RAM):** Mindestens 16 GB
- **Grafikkarte (GPU):** Leistungsstarke GPU mit mindestens 8 GB VRAM, vorzugsweise eine NVIDIA-GPU mit CUDA-Unterstützung
- **Festplattenspeicher:** Ungefähr 4 GB für das Modell

RAG CODE-FIRST-ANSATZ



python

```
app.py M X # style.css U script.js U index.html M
app.py
26 app = Flask(__name__)
27
28 # Logging aktivieren
29 logging.basicConfig(level=logging.DEBUG)
30
31 # Modell und Pfad zur PDF-Datei
32 local_model = "llama3.1:8b-instruct-q8_0"
33 llm = ChatOllama(model=local_model)
34 print("Verwendetes Modell:", llm.model_dump)
35 local_path = "KI_in_der_Lehre.pdf"
36
37 # Lade das PDF und bereite es vor
38 loader = UnstructuredPDFLoader(file_path=local_path)
39 data = loader.load()
40
41 # Split and chunk
42 text_splitter = RecursiveCharacterTextSplitter(chunk_size=700,
43 chunks = text_splitter.split_documents(data)
44
45 # Add to vector database
46 vector_db = Chroma.from_documents(
47 documents=chunks,
48 embedding=OllamaEmbeddings(model="nomic-embed-text:latest",
49 collection_name="local-rag")
50 )
51
52 # Multi-query retriever
53 QUERY_PROMPT = PromptTemplate(
54 input_variables=["question"],
55 template="""Du bist ein KI-Sprachmodell-Assistent. Deine A
56 Originalfrage: {question}""",
```

javascript

```
static > app.py M # style.css U script.js U X index.html M
static > script.js > selectFile
1 /* static/script.js */
2 async function sendMessage() {
3 const user_input = document.getElementById("user_input");
4 if (!user_input) return;
5
6 const response = await fetch('/chat', {
7 method: 'POST',
8 headers: {
9 'Content-Type': 'application/json'
10 },
11 body: JSON.stringify({ user_input: user_input })
12 }).then(res => res.json());
13
14 const chatWindow = document.getElementById("chat_window");
15
16 // Benutzer-Nachricht anzeigen
17 const userMessageDiv = document.createElement("div");
18 userMessageDiv.classList.add("chat-message", "user");
19 const userBubble = document.createElement("div");
20 userBubble.classList.add("chat-bubble", "user");
21 userBubble.innerText = user_input;
22 userMessageDiv.appendChild(userBubble);
23 chatWindow.appendChild(userMessageDiv);
24
25 // Antwort des ChatAssistenten anzeigen
26 if (response.response) {
27 const botMessageDiv = document.createElement("div");
28 botMessageDiv.classList.add("chat-message", "chatassistant");
29 const botBubble = document.createElement("div");
30 botBubble.classList.add("chat-bubble", "chatassistant");
```

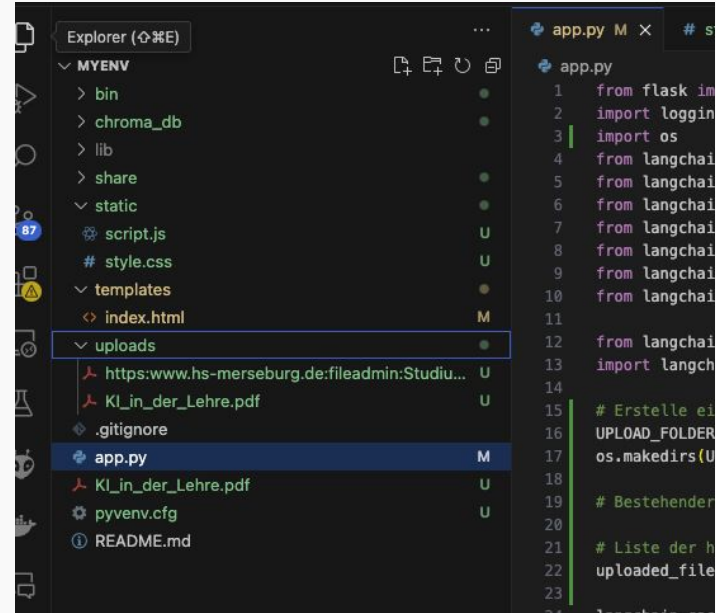
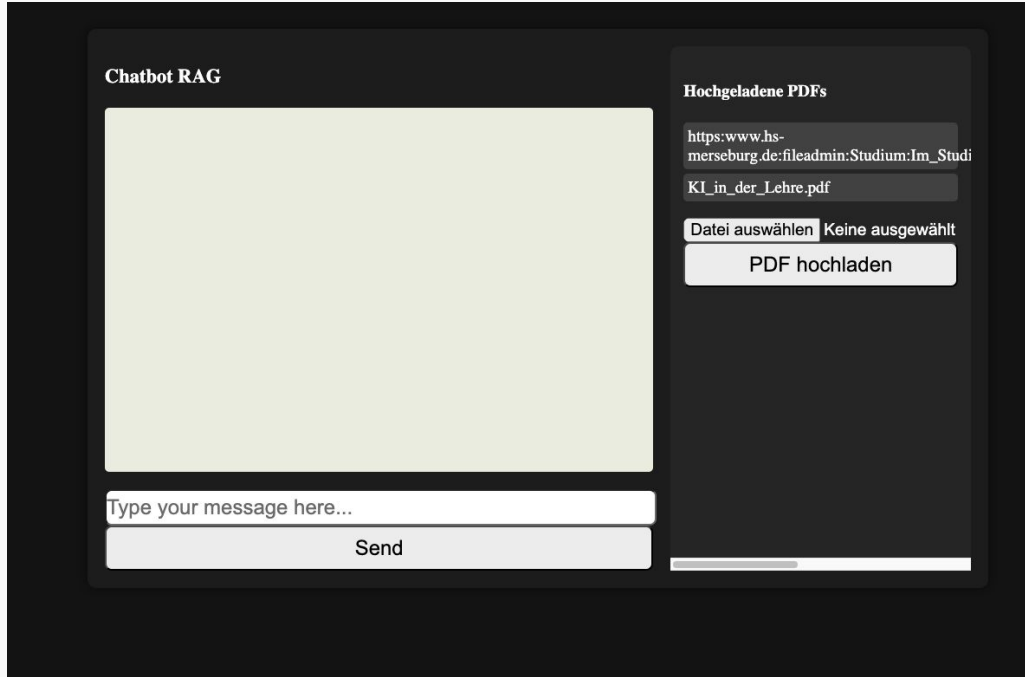
html

```
app.py M # style.css U script.js U index.html M X
templates > index.html > html > body > div.chat-container > div.side
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-
7 <title>Chat with LLM</title>
8 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/
9 integrity="sha384-whJ6K2fWJAwg1E+Vfrckps5BNty4qK5sIQZ
10 <link rel="stylesheet" href="{{ url_for('static', filename=
11 </head>
12
13 <body>
14 <div class="chat-container">
15 <div class="chat-window">
16 <h3 class="text-center">Chatbot RAG</h3>
17 <div id="chat_window"></div>
18 <div class="input-group">
19 <input type="text" id="user_input" class="form-
20 placeholder="Type your message here..">
21 <button class="btn btn-primary send-button" onc
22 </div>
23 </div>
24 <div class="sidebar">
25 <h4>Hochgeladene PDFs</h4>
26 <ul id="file_list" class="file-list">
27 {% for file in uploaded_files %}
28 <li data-file-name="{{ file }}" onclick="selectF
29 |<a href="/upload/{{ file }}" target="blank
30 </li>
```

CSS

```
app.py M # style.css U X script.js U index.html M
static > # style.css > body
1 /* Allgemeine Stile */
2 body {
3 background-color: #121212;
4 color: #ffffff;
5 font-size: 1.2rem;
6 }
7
8 .chat-container {
9 display: flex;
10 justify-content: space-between;
11 max-width: 1000px;
12 margin: 50px auto;
13 padding: 20px;
14 background-color: #1e1e1e;
15 border-radius: 10px;
16 box-shadow: 0 0 15px #000, 0 0, 0, 0.5;
17 }
18
19 .chat-window {
20 flex: 2;
21 margin-right: 20px;
22 }
23
24 .sidebar {
25 flex: 1;
26 background-color: #2a2a2a;
27 padding: 15px;
28 border-radius: 10px;
29 max-height: 100%;
30 overflow-y: auto;
```

RAG CODE-FIRST-ANSATZ



<https://github.com/BoGlavatkI/RAG-Py-LC-Chroma>

04

Vergleich der Ergebnisse

Boleslav Glavatki & Yolanda Sommerfeldt

VERGLEICH ChatGPT vs. OPEN SOURCE

Stellt SL2 einen
Plagiat-Scanner/Plagiat
-Tool zur Verfügung?

01

Kann mir SL2 zum
Thema KI in der
Lehre weiterhelfen?

02

Ich möchte gerne mit
meinen Kolleg:innen
Fußball spielen. Kann SL2
mir einen Sport-
platz reservieren?

03

VERGLEICH ChatGPT vs. OPEN SOURCE

ChatGPT

User

Kann mir SL2 zum Thema KI in der Lehre weiterhelfen?

Chatbot ChatGPT

Ja, SL² kann definitiv zum Thema KI in der Lehre weiterhelfen. Das Projekt "KI in Lehre und Forschung" an der Hochschule Merseburg konzentriert sich auf die Integration künstlicher Intelligenz in den Bildungsbereich. Es werden praktische Einsatzmöglichkeiten von textgenerierenden KI-Tools erforscht und diskutiert, und es wird angestrebt, konkrete Anwendungsfälle für generative KI-Tools wie Chatbots zu identifizieren und zu erproben[1].

Für weitere Informationen und Details zu den spezifischen Anwendungen und Themenfeldern von KI in der Lehre können Sie die entsprechende Webseite der Hochschule Merseburg besuchen: KI in der Lehre: Hochschule Merseburg[2].

Open Source

User

Kann mir SL2 zum Thema KI in der Lehre weiterhelfen?

Chatbot Open Source:

Ja, das Team von SL² bietet Unterstützung bei verschiedenen Themen rund um KI in der Lehre an. Dazu gehören unter anderem die Einführung und Umsetzung von KI-Anwendungen wie z.B. F4X und NoScribe, sowie die Beratung bei der Fragebogenerstellung und -auswertung. Durch die Nutzung der PlagAware-Software können Sie Ihre wissenschaftlichen Arbeiten auch auf Plagiatsfälle überprüfen lassen. Wenn Sie Interesse an unseren Angeboten haben, stehe ich Ihnen gerne zur Verfügung um gemeinsam das passende Angebot für Ihr Projekt zu finden.

VERGLEICH ChatGPT vs. OPEN SOURCE

ChatGPT

User

Stellt SL2 eine Plagiat-Tool zur Verfügung?

Chatbot ChatGPT

SL² an der Hochschule Merseburg bietet spezifische Unterstützung im Bereich des Lehrens und Lernens. Für Informationen über spezielle Tools wie ein Plagiat-Tool müsste ich die hochgeladenen Dateien durchsuchen. Möchten Sie, dass ich das tue, um Ihre Frage zu beantworten?

User

ja

Chatbot ChatGPT

Ja, SL² bietet ein Plagiat-Tool namens „Plagaware“ an. Dieses Tool ermöglicht es, wissenschaftliche Arbeiten automatisch auf Plagiate zu überprüfen. Lehrende und Studierende der Hochschule Merseburg können eine kostenlose Lizenz erhalten, um diese Software zu nutzen [1].

Open Source

User

Stellt SL2 eine Plagiat-Tool zur Verfügung?

Chatbot Open Source:

Ja, es gibt eine Webseite für PlagAware. Sie können unter plagiarismware.com/de weitere Informationen finden. Hier erhalten Sie detaillierte Informationen über die Funktionen und Möglichkeiten der Software, sowie Anleitungen und Videos zur Anwendung von PlagAware. Wenn Sie Fragen zu PlagAware haben oder Unterstützung benötigen, können Sie sich auch gerne an das Team von SL² wenden, das Ihnen bei der Nutzung der Software helfen wird. Die Kontaktinformationen sind auf der Webseite angegeben.

VERGLEICH ChatGPT vs. OPEN SOURCE

ChatGPT

User

Ich möchte gerne mit meinen Kolleg:innen Fußball spielen. Kann SL2 mir einen Sportplatz reservieren?

Chatbot ChatGPT

Ich bin ein Chatbot für SL² und kann spezifische Anfragen zum Thema Lehren und Lernen an der Hochschule Merseburg bearbeiten. Leider kann ich keine Reservierungen für Sportplätze vornehmen. Ich empfehle, sich direkt an die zuständige Abteilung oder das Sportzentrum der Hochschule Merseburg zu wenden, um Unterstützung bei der Reservierung eines Sportplatzes zu erhalten.

Open Source

User

Ich möchte gerne mit meinen Kolleg:innen Fußball spielen. Kann SL2 mir einen Sportplatz reservieren?

Chatbot Open Source:

Das ist keine Frage, die direkt zum bereitgestellten Kontext passt. SL² (Stärkung des Lehrens und Lernens) ist ein Projekt, das sich auf die Verbesserung der Lehre und des Lernens an der Hochschule konzentriert und bietet Unterstützung bei Fragen wie der Umsetzung von Lehr- und Lernkonzepten, der Durchführung von Online-Befragungen und der Verwendung von Software wie PlagAware.

Ich kann Ihnen nicht dabei helfen, einen Sportplatz zu reservieren. Für solche Anfragen wenden Sie sich bitte an das Hochschulressourcenzentrum oder die entsprechende Abteilung an der Hochschule.

VERGLEICH ChatGPT vs. OPEN SOURCE

ChatGPT

- Blackbox ChatGPT
- Sehr viel Entwicklung, eigenes Chat-Modell nah an state of the art
- Abhängigkeit von OpenAI
- Serverleistung, Kosten
- Benutzerfreundlich und intuitiv
- Weniger Aufwand und Vorwissen
- Nutzung multimodaler Modelle möglich (z.B. von Bildern)

Open Source

- Keine Kosten für Software
- Komplette lokale Anwendung
- Sehr sicher, Infos gehen nicht nach außen
- Volle Kontrolle über die Implementierung
- Feinheiten, Änderungen können gemacht werden
- Informationen können schnell verändert werden
- Update der Datenbank schnell möglich
- Komplexer Aufbau, Expertenwissen notwendig
- Hardware Anforderungen (Beschaffungskosten)
- Hoher Erstaufwand
- Kein Plug and Play
- Programmierkenntnisse notwendig
- Kein UI

04

Praktische Demonstration & Diskussion

Christian Kandels

Vielen Dank

FÜR IHRE AUFMERKSAMKEIT!

MÖGLICHE EINSATZFELDER



BILDUNGSKONTEXT

- als Ideengeber
- als Sparringpartner:in
- zur Analyse von qualitativen Interviews



VERWALTUNG

- für neue Mitarbeitende der HoMe
- für Lehrende
- als intranet alternative