Pietrusky, Stefan

KI und offene Bildung - Bildung ohne künstliche Barrieren. Mit KI-Kompetenz und Open Source für mehr Gerechtigkeit bei der individuellen Kompetenzentwicklung

Heidelberg 2025, 13 S.

KI und offene Bildung - Bildung ohne künstliche Barrieren: Mit KI-Kompetenz und Open Source für mehr Gerechtigkeit bei der individuellen Kompetenzentwicklung

Dr. Stefan Pietrusky[1]

[1]Heidelberg Center for Digital Humanities (HCDH), Heidelberg University
[1]stefan.pietrusky@uni-heidelberg.de
[1]Down Church Studios

16. Mai 2025

Abstract

Die zunehmende Verbreitung von KI im Bildungsbereich birgt Chancen und Herausforderungen. Während proprietäre Systeme den Zugang einschränken, bieten Open-Source-Modelle wie Llama Alternativen zur Förderung der Chancengleichheit. Der Beitrag stellt mit Open WebUI und dem Individual Educational Chatbot (IEC) zwei Ansätze vor, mit denen sich Lernassistenten lokal betreiben und individuell anpassen lassen. Zudem wird auf relevante KI-Kompetenzen und den Einfluss von Parametern auf LLM-Antworten eingegangen. Die Integration von Open-Source-KI in Bildungseinrichtungen erfordert didaktische Konzepte und kontinuierliche Weiterbildung, um die Technologie effektiv zu nutzen und Bildungsungleichheiten zu reduzieren.

Schlüsselbegriffe: KI, KI-Kompetenzen, Open-Source, Bildungsgerechtigkeit, IEC, Open WebUI

1 EINLEITUNG

Die rasante Entwicklung im Bereich der Künstlichen Intelligenz (KI) hat sowohl im Open-Source-Bereich, beispielhaft durch Plattformen wie Hugging Face, als auch im kommerziellen Sektor mittlerweile zur Entwicklung zahlreicher neuer Tools geführt [1]. Diese Tools nutzen die wissenschaftlichen Erkenntnisse des maschinellen Lernens und schaffen damit Anwendungen, die den Alltag der Menschen erleichtern sollen. Insbesondere im Bildungsbereich finden sich zunehmend KI-basierte Anwendungen, für die jedoch kostenpflichtige Abonnements erforderlich sind. Ohne Zahlung ist die Funktionalität stark eingeschränkt, wodurch diejenigen, ohne ausreichend finanzielle Mittel schnell benachteiligt werden [2] [3]. So können dann nur ältere und weniger leistungsfähige Modelle verwendet oder erst nach einem bestimmten Zeitintervall wieder Anfragen gestellt werden. Diese künstliche Ausgrenzung aufgrund wirtschaftlicher Barrieren widerspricht der Hoffnung, dass KI den Bildungszugang und die Bildungschancen für alle gleichermaßen verbessern kann. Von Anfang an spielen soziale Implikationen eine Rolle, wenn es um den Einsatz von KI in der Bildung geht. Das Problem ist, dass ohne eine gewisse technische Affinität oder KI-Kompetenz, die bereits jetzt vorhandenen Möglichkeiten im Bereich der offenen KI vielen verschlossen bleiben aufgrund eines zu schweren Einstiegs wie diese Technologie verwendet werden kann [4] [5].

Diese Unwissenheit führt dazu, dass akzeptiert wird, dass unzählige proprietäre KI-Anwendungen die Bildungsungerechtigkeit weiter verstärken. Es gibt bereits verschiedene Möglichkeiten, mit einfachen Mitteln einen eigenen, lokal betriebenen Lernassistenten einzurichten und weiterzuentwickeln. Um Lernchancen für alle zu verbessern und die individuelle Kompetenzentwicklung mithilfe von KI zu unterstützen, ist es erforderlich, offene Alternativen zu kommerziellen Lösungen sichtbar zu machen, um die Chancengleichheit im Bildungsbereich zu verbessern [6] [7]. Im Rahmen dieses Beitrags werden zwei Möglichkeiten aufgezeigt. Die erste Möglichkeit umfasst die Anwendung Open WebUI, eine benutzerfreundliche Schnittstelle zur Nutzung lokaler KI-Modelle. Die zweite Option beinhaltet den Individual Educational Chatbot (IEC) bei dem das Interface mithilfe von HTML, CSS und JavaScript erstellt wird. Die Modelle kommen in beiden Fällen von Ollama, einem System über das verschiedene Modelle heruntergeladen und lokal verwendet werden können. Im direkten Vergleich zwischen Open WebUI und IEC bietet die zweite Anwendungen eine niederschwelligere Möglichkeit hinsichtlich der individuellen Anpassung der Ausgaben des Lernassistenten.

Konkret dadurch, dass das Kompetenzniveau, also die Art, in der der Assistent antwortet, vom Nutzer selbst bestimmt werden kann. Die, für die Entwicklung erforderlichen, Werkzeuge stehen alle kostenlos zur Verfügung. Als Ergebnis hat man zwei Tools, die durch ihre Codestruktur einfach und schnell angepasst werden können. So können Modelle ausgetauscht werden, die in bestimmten Bereichen (Bsp. Programmierung; codellama) bessere Ergebnisse liefern. Bevor erklärt wird, wie man, die für die beiden Anwendungen erforderlichen, Modelle erhält, wird im folgenden Kapitel darauf eingegangen, welche Kompetenzen im KI-Bereich relevant sind.

2 KI-KOMPETENZEN

Das weltweit erste umfassende gesetzliche Rahmenwerk für KI ist der von der Europäischen Union verabschiedete AI Act [8]. Ziel ist es, die mit dieser Technologie einhergehenden Risiken zu adressieren und dadurch Europa eine führende Rolle zu sichern [9]. Jeder, der sich im KI-Bereich auskennt, weiß jedoch, dass Europa im Vergleich zu anderen Ländern bereits den Anschluss verloren hat und wahrscheinlich nur mit großen Investitionen aufholen kann. Ein aktuelles Beispiel zeigt hier aber auch deutliche Unterschiede. So plant Frankreich mehr als 100 Milliarden Euro in KI-Innovationen zu investieren und die USA, die bereits Vorreiter sind, planen im Rahmen des Stargate Projekts

Investitionen von 500 Milliarden Dollar.

Unabhängig davon werden durch den AI Act KI-Anwendungen je nach Risikoniveau in Kategorien eingeteilt [10]. KI-Systeme im Bereich soziale Bewertung und Manipulation werden als unvertretbar riskant eingestuft und sind dadurch verboten [8]. Videospiele, die mithilfe von KI arbeiten, sind hingegen erlaubt. Die Einhaltung der Vorschriften liegt in der Verantwortung der Entwickler [11]. Der AI Act definiert keine spezifischen Qualifikationen oder Zertifizierungen für Entwickler solcher Anwendungen. Um sichere und damit konforme Systeme zu entwickeln, benötigen die Entwickler aber spezifische Kompetenzen. Über welche Fähigkeiten muss man verfügen, um KI-Anwendungen zu entwickeln oder überhaupt einsetzen zu können? Die UNESCO hat einen Kompetenzrahmen entwickelt und es gibt das Kompetenzmodell AIComp. Die praktische Anwendung und die Fähigkeit zur kritischen Reflexion von KI steht im Fokus von AIComp [12]. Die UNESCO hingegen hat eher allgemeine digitale Kompetenzen im Umgang mit KI formuliert [13]. Eine Gemeinsamkeit beider Ansätze ist die Betonung der ethischen Verantwortung im Umgang mit KI. Auch sollen Informationen kritisch hinterfragt werden und fundierte Entscheidungen im Kontext von KI getroffen werden.

Um mit der schnellen Entwicklung Schritt halten zu können, ist es bei beiden Texten wichtig, dass man kontinuierlich neue Kenntnisse erwirbt, ganz im Sinne des lebenslangen Lernens. Ebenfalls ist es wichtig zu verstehen, wie KI-Systeme funktionieren und welche Auswirkungen sie haben können. Vor diesem Hintergrund gilt es folgende Fragen beantworten zu können. Wie funktionieren KI-Modelle und welche Daten nutzen sie? Welche Risiken und Verzerrungen bestehen bei Inhalten, die mithilfe von KI generiert werden? Wie kann man diese bewerten und hinterfragen? Und wie können KI-Anwendungen sinnvoll eingesetzt werden? Die Überschneidungen der beiden Ansätze zusammengefasst, setzt sich eine grundlegende KI-Kompetenz zusammen, aus einer ethischen Kompetenz, dem kritisches Denken, der Notwendigkeit des lebenslangen Lernens und einem grundlegenden Systemverständnis. Ziel ist es, Menschen auf die Chancen und Herausforderungen einer KI-geprägten Welt vorzubereiten. In folgendem Kapitel gehen wir auf eine leicht zu verwendende Quelle für Large Language Modelle (LLMs) ein, die für die zwei erwähnten Anwendungen benötigt wird.

3 EIN KÖNIGREICH FÜR OLLAMA

Inzwischen gibt es verschiedene Open-Source-Plattformen, mit denen man große Sprachmodelle lokal auf dem eigenen Rechner ausführen kann. Je nach System können die KI-Modelle auf CPUs- oder GPUs laufen. Modelle auf diese Art und Weise zu verwenden, bietet für Bildungseinrichtungen im Bereich Datenschutz einen entscheidenden Vorteil. Die Verwendung von LLM ohne Internetverbindung sorgt dafür, dass der Input, also die Daten, die vom Nutzer eingegeben werden, nicht auf irgendwelchen Servern gespeichert werden, die sich ggf. nicht an die DSGVO halten [14]. OpenAI speichert die Anfragen der Nutzer, um ihre Modelle zu verbessern. Im Bildungskontext wäre diese Vorgehensweise aufgrund der sensiblen Daten problematisch.

In diesem Beitrag werden die LLMs vom Modellmanagmentsystem Ollama geladen, da es die benutzerfreundlichste Lösung in diesem Bereich ist. Konkret ist Ollama für alle Betriebssysteme (Mac, Linux und Windows) optimiert und kann deshalb überall eingesetzt werden. Ebenso können bei einigen Modellen bestimmte Parameter angepasst werden, um die Ausgaben der LLMs zu steuern [14]. Alternativen zu Ollama sind LM Studio, GPT4All, LM Deploy, llama.cpp, vLLM und OpenWebUI auf das später noch eingegangen wird. Die Anwendung wird von der offiziellen Seite (https://ollama.com/) heruntergeladen und installiert. Nach der Installation muss das System ggf. neu gestartet werden. Wenn Ollama installiert ist, geht man über die Startseite der Seite in den

Bereich „Models“. In der Übersicht sieht man die aktuell verfügbaren Modelle. Auch das viel diskutierte Modell R1 von Deepseek ist nutzbar. Die Modelle werden zudem in drei Kategorien (Vision, Embedding und Tool) unterteilt. Vision-Modelle (Bsp. LLaVA) sind darauf spezialisiert, visuelle Daten (Bilder) zu verarbeiten und zu interpretieren. Sie werden für Aufgaben im Bereich der Objekterkennung, Bildbeschreibung und Texterkennung in Bildern verwendet. Embedding-Modelle erzeugen Vektor-Repräsentationen von Texten oder anderen Daten. Die semantische Bedeutung der Eingaben wird durch die Vektoren erfasst, wodurch sie im Bereich Clustering und Ähnlichkeitsanalysen verwendet werden können. Wenn externe Funktionen genutzt werden sollen, verwendet man Tool-Modelle. Um auf externe Datenquellen (Bsp. Wetterinformationen, Aktienkurse) zuzugreifen, werden Werkzeuge implementiert, die dann weiterverarbeitet werden können. Ein paar Modelle und deren unterschiedlich verfügbaren Varianten (Parameteranzahl) werden in nachfolgender Tabelle dargestellt (siehe Tabelle 1).

Tabelle 1: Übersicht von verschiedenen Modellen, die auf Ollama verfügbar sind.

Modellname	**Verfügbare Varianten**	**Kategorie**
DeepSeek-R1	1.5B, 7B, 14B, 32B, 70B, 671B	All
Gemma 2	2B, 9B, 27B	All
Phi 3 (4)	3.8B, 14B (14B)	All
Mistral	7B, 8×7B, 8×22B	Tool
Llama 3.1(2)	(1B, 3B), 8B, 70B, 405B	Tool
nomic-embed-text	137M	Embedding
LLaVA	7B, 13B, 34B	Vision

Je nach verfügbarem System gilt es zu beachten, welche Version man auswählt. Die Quantisierung ist eine Technik, mit der die Rechenanforderungen und der Speicherbedarf von LLMs reduziert wird [15]. Da der Anwendungsfall die lokale Ausführung und nicht ein eigener Trainingsprozess ist, reicht eine niedrige Quantisierungsstufe (Q4). Die Anzahl der Parameter eines Modells wird mit „B“ (Billion = Milliarden) angegeben. Je mehr Parameter ein Modell hat, also je größer es ist, desto besser ist sein Kontextverständnis und die Qualität der Ausgaben, weil komplexe Zusammenhänge besser verstanden werden. Wie bei der Quantisierung ist es auch bei der Anzahl der Parameter, je größer, desto mehr Speicher ist erforderlich. Für lokale Anwendungen eignen sich Modelle im Bereich 7B-8B. Als Faustregel kann man sagen, dass man für 7B Modelle 8 GB RAM benötigt. Bei 13B sind es 16 GB und bei 33B Modellen 32 GB RAM [14]. Hierbei muss noch der Unterschied zwischen CPU (Central Processing Unit) und GPU (Graphics Processing Unit) beachtet werden. Die CPU führt Aufgaben seriell aus und die GPU parallel. Bei Deep Learning Modellen, wie LLM, ist die parallele Verarbeitung der GPU ein wesentlicher Vorteil, da umfangreiche Matrizenoperationen (z.B. Tensor-Berechnungen) durchgeführt werden müssen. Die Geschwindigkeit hängt also auch von der Art des zur Verfügung stehenden Speichers ab. Es gibt Modelle, die ohne eine GPU gar nicht funktionieren.

Die Daten verdeutlichen das grundlegende Problem, wenn KI im Bildungsbereich verwendet wird. Wenn leistungsfähige Geräte zur Verfügung stehen, können bessere Modelle verwendet werden. Wer nicht über eine entsprechende Ausstattung verfügt, muss externe Anwendungen nutzen, bei denen die Modelle auf Servern gehostet werden, wodurch diese kostenpflichtig sind [15]. Je nach Anwendung wird ein Kredit- oder Tokensystem bzw. Kontingentmodell verwendet. Token werden über ein Abo erworben und für Anfragen benötigt. Je komplexer die Anfrage ist, desto mehr Tokens werden benötigt. Wenn keine mehr vorhanden sind, kann man auch keine weiteren Anfragen stellen.

In diesem Zusammenhang ist es wichtig zu wissen, dass bereits kleinere Modelle über ein ausreichendes Sprachverständnis verfügen, um effektive Tools einrichten zu können. Für die Lernassistenten kann entweder das Modell llama3.1 (8B) oder llama3.2 (3B) installiert werden. Je nach verfügbarem Platz auf der Festplatte kann auch ein anderes Modell oder eines mit mehr Parametern installiert werden. Je mehr Parameter ein Modell hat, desto mehr Platz benötigt es auf der Festplatte. Die Modelle von Meta AI basieren auf der Transformer-Architektur und wurden auf große und vielfältige Datensätzen trainiert. Durch die Offenheit und Zugänglichkeit eignen sich diese Modelle für praktische Anwendungen. Je nachdem für welches Modell man sich entschieden hat, führt man den Befehl zur Installation ollama run llama3.2 über das Terminal bzw. Eingabeaufforderung oder Shell des Systems aus. Wenn das Modell installiert ist, kann man über das Terminal mit diesem kommunizieren. Um den Chat zu starten, verwendet man einfach den gleichen Befehl. Die Interaktion mit dem LLM über das Terminal ist natürlich nicht optimal, weshalb im nächsten Kapitel erklärt wird, wie man eine grafische Benutzeroberfläche (GUI) einrichtet, um die Interaktion zu verbessern. Für besagte Oberfläche wird Open WebUI verwendet.

4 OPEN WEBUI

Open WebUI ist wie Ollama ein Open-Source-Projekt über das man ohne Programmierkenntnisse LLM wie llama3.1, lama3.2 etc. lokal über eine Benutzeroberfläche, die an ChatGPT von OpenAI erinnert, verwenden kann. Das Interface kann zudem um weitere Funktionen erweitert werden [16]. Mithilfe von Open WebUI wird der Austausch mit dem bereits installierten LLM vereinfacht. Die Einrichtung erfolgt über Docker, die es ermöglicht Anwendungen in Container zu verpacken, die dann geteilt und ausgeführt werden können (Docker, n.d.). Alle Abhängigkeiten (Bibliotheken, Konfigurationen etc.), die eine Anwendung benötigt sind in diesem Container enthalten. Der Vorteil hierbei ist, dass der Container plattformunabhängig ist und damit, nachdem er einmal entwickelt ist, überall läuft. Auf der offiziellen Seite von Docker (https://www.docker.com/) wird Docker Desktop in der Produktübersicht heruntergeladen. Wenn die Anwendung geladen und installiert ist, wird sie gestartet.

In der sich öffnenden Übersicht ist aktuell noch kein Container vorhanden. Um Open WebUI zu installieren, gehen wir auf das entsprechende GitHub Repository (https://github.com/open-webui/open-webui). Es gibt verschiedene Möglichkeiten, die App zu installieren. Die erforderlichen Schritte werden im Bereich „How to Install“ erklärt. Da hier Docker verwendet und Ollama bereits installiert ist, kann mithilfe des nachfolgenden Befehls über das Terminal Open WebUI installiert werden (siehe Abbildung 1).

```
docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main
```

Abbildung 1: Befehl zur Installation von Open WebUI als Docker Container.

Der Befehl kann über das Repository kopiert und in das Terminal implementiert werden. Nachdem der Befehl ausgeführt wurde, sollte der Open WebUI Container in Docker Desktop dargestellt werden. Man klickt auf „Port“, woraufhin die Anwendung im Browser gestartet wird. Für die Nutzung muss man einen Account erstellen. Nachdem das erledigt ist, startet die Anwendung. Als nächstes wählt man das heruntergeladene LLM aus und kann über das GUI mit diesem ohne Einschränkungen kommunizieren (siehe Abbildung 2).

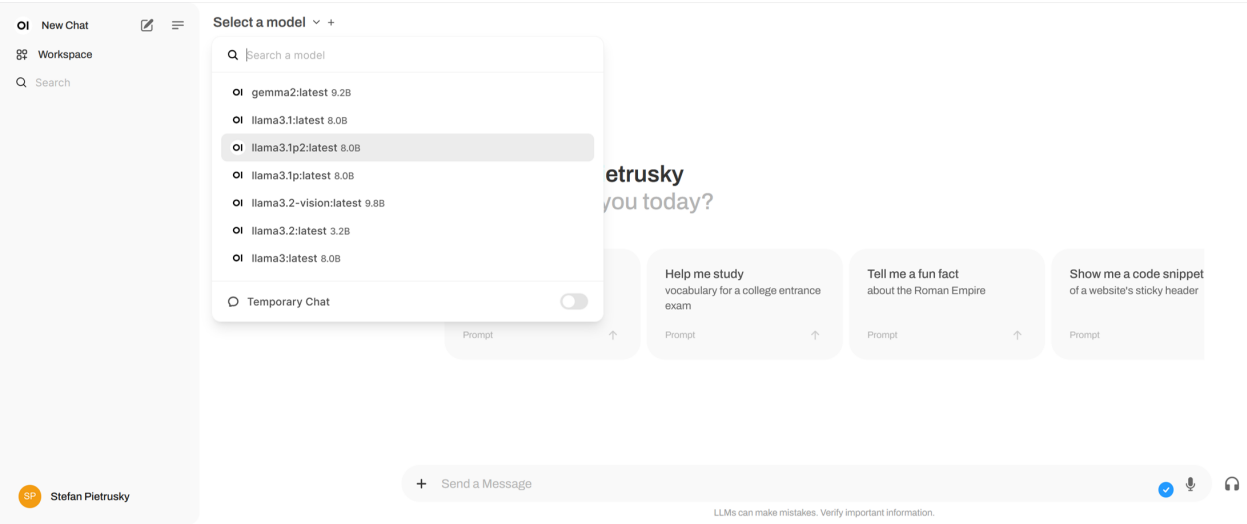

Abbildung 2: Befehl zur Installation von Open WebUI als Docker Container.

Die hier beschriebene Vorgehensweise verdeutlicht, wie mit wenigen Schritten ein individueller Lernassistent eingerichtet werden kann, ohne dass dabei irgendwelche Kosten anfallen. Da die bereits bei Ollama vorhandenen Modelle kontinuierlich weiterentwickelt werden, kann der Nutzer austesten, welche Modelle in bestimmten Bereichen am besten performen. Die Anzahl der Open-Source- bzw. Open-Weight-Modelle steigt kontinuierlich und die aktuellen Veröffentlichungen zeigen, dass frei zugängliche Modellen proprietäre in Benchmarks, standardisierte Test oder Metriken, übertreffen bzw. deren Leistungsdaten nicht weit von diesen entfernt sind [17]. Ein aktuelles Beispiel ist das Modell R1 des chinesischen Startups DeepSeek (siehe Tabelle 2).

Tabelle 2: Übersicht verschiedener Open-Source- und proprietärer Modelle und deren Leistungsdaten

Modell	**MMLU (%)**	**Arc-Challenge (%)**	**GSM8K (%)**	**HellaSwag (%)**	**OS**
Llama 3.1 (Meta)	79.8	70.5	87.3	84.2	Ja
Mistral 7B	76.5	68.3	85.1	81.7	Ja
DeepSeek R1	81.2	72.4	88.5	85.6	Ja
GPT-4o (OpenAI)	86.4	79.2	92.5	89.3	Nein
Claude 3.5 (Anthropic)	83.2	75.5	90.4	87.0	Nein
Gemini 1.5 (Google)	82.1	73.4	88.9	85.5	Nein

Auch wenn kommerzielle Systeme aktuell noch leistungsstärker sind, sind sie oft intransparent bzgl. ihrer Trainingsdaten. Hingegen bieten offene Modelle mehr Anpassungsmöglichkeiten, ermöglichen einen Datenschutz konformen lokalen Einsatz und fördern eine offene Wissenschaftskultur, welche die Grundlage für künftige Innovationen ist. Bevor auf die zweite Möglichkeit eingegangen wird, wie man sich mit einem LLM auszutauschen kann, wird im folgenden Kapitel erklärt, wie man durch die Konfiguration von ein paar Parametern entweder ein deterministisches (präzise) oder nicht-deterministisches (kreativ) Verhalten des Modells erzielen kann, wodurch sich weitere Einsatzmöglichkeiten ergeben.

5 MASCHINEN HALLUZINIEREN

Die Kombination von Ollama und Open WebUI bietet bereits viele Vorteile und Möglichkeiten, um mit verschiedenen LLM zu interagieren. Je nachdem welche Rechenleistung zur Verfügung stehen, wie bereits erklärt, können größere oder kleinere Modelle verwendet werden, ohne mit allzu langen Wartezeiten bei der Generierung des Outputs konfrontiert zu werden. Bei kleineren Modellen ist die Wissensdatenbank (Knowledge Base) begrenzt bzw. allgemein eingeschränkt und kann auf aktuelle Entwicklungen keine richtigen Antworten geben [18]. Wenn ein Modell auf eine Frage keine Antwort hat, kommt es zu kreativem Phantasieren, das durch verschiedene Parameter beeinflusst wird [19].

Die Zufälligkeit einer Ausgabe wird durch die Temperatur bestimmt. Ein hoher Wert führt zu mehr Kreativität aber auch zu mehr Halluzination [20]. Wenn weniger Kreativität gewünscht ist, werden niedrige Werte festgelegt. Für deterministische Ausgaben, also keinerlei Variation, kann der Parameter auch mit einem Wert von 0 bestimmt werden. Das Top-k Sampling begrenzt die Auswahl der nächsten Wörter auf die k wahrscheinlichsten Token (Wörter). Ein hoher Wert bei diesem Parameter führt zu mehr Variation und kreativem Raten, wohingegen ein niedriger Wert für sichere, faktenbasierte Antworten steht [21]. Die Kohärenz nimmt ab, wenn das Modell aus zu vielen Möglichkeiten wählt. Wenn die Auswahl der Tokens nicht nach einer fixen Anzahl (Top-k) erfolgt, sondern nach einer Wahrscheinlichkeitswelle, handelt es sich um Top-p Sampling. Auch hier sorgen hohe Werte für kreativere und manchmal weniger faktenbasierte Antworten [21]. Je höher Top-p eingestellt ist, desto eher werden unkonventionelle Begriffe in die Antwort eingebaut. Wie oft neue oder bereits genutzte Begriffe wiederholt werden, wird durch den Parameter Presence Penalty beeinflusst. Wenn das LLM innovativer und weniger repetitiv antworten soll, ist hier ein hoher Wert erforderlich.

Wenn bereits mehrfach verwendete Token bestraft werden sollen, spielt der Parameter Frequency Penalty eine Rolle. Um zu gewährleisten, dass sich das Modell neue Ausdrucksweisen ausdenkt, ist ein hoher Wert erforderlich. Ein weiterer wichtiger Faktor ist die Kontextlänge. Je nachdem für welches Modell man sich bei Ollama, oder einem anderen System entschieden hat, kann das LLM weniger Kontext behalten. Das LLM Llama 3.1 von Meta kann beispielsweise entweder 4096, 8192 oder 131072 Token behalten. Ein langes Kontextfenster ermöglicht es dem Modell sich besser an den Kontext zu halten und weniger zu erfinden [18]. Der Seed-Wert spielt ebenfalls eine wesentliche Rolle für die Konsistenz und Kreativität der Ausgaben eines Modells [21]. Wenn das LLM auf die gleiche Eingabe immer die gleiche Antwort geben soll, ist ein konstanter Parameterwert erforderlich. Für eine variierende Ausgabe ist hingegen ein zufälliger Wert notwendig. Je nach Einsatzgebiet können die genannten Parameter dazu beitragen, dass ein LLM halluziniert oder weniger Unsinn erfindet. Wenn LLM beispielsweise bei der Analyse großer Textmengen verwendet wird, kann es sein, dass eine nicht-deterministische Konfiguration des Modells durch entsprechende Werte bei den genannten Parametern, bei gleichem Inhalt, unterschiedlichen Output generiert. Für kreative Texte oder Geschichten müssen die Parameter anders eingestellt werden. In der nachfolgenden Tabelle werden zwei entsprechende Konfigurationen dargestellt (siehe Tabelle 3).

Tabelle 3: Parameterkonfiguration für deterministische und nicht-deterministische Modelle

Modell (deterministisch)		**Modell (nicht-deterministisch)**	
`temperature`	< 0.7	`temperature`	> 1.0
`top_k`	< 40	`top_k`	> 100
`top_p`	< 0.8	`top_p`	> 0.9
`presence_penalty`	0	`presence_penalty`	1.2
`frequency_penalty`	0–0.2	`frequency_penalty`	0.8
`context length`	max	`context length`	variabel
`seed`	42	`seed`	random

Ein weiterer Vorteil von Ollama ist, dass viele der Parameter über die Modeldatei, eine einfache .txt-Datei, angepasst werden können. Durch die Anpassung, die nicht von jedem Modell unterstützt wird, erstellt man neue Varianten von lokal installierten LLMs. Wie eine Modeldatei aufgebaut sein muss und wie der Befehl lautet, um eine speziell konfigurierte Version eines LLM zu erstellen, wird in nachfolgender Abbildung dargestellt (siehe Abbildung 3).

```
FROM llama3.1

SYSTEM Please strictly follow the instructions of the
transmitted context.

PARAMETER temperature 0
PARAMETER seed 42
PARAMETER top_p 1
PARAMETER stop "<|start_header_id|>"
PARAMETER stop "<|end_header_id|>"
PARAMETER stop "<|eot_id|>"
PARAMETER stop "<|reserved_special_token>„

ollama create llama3.1p -f C:|Users|XX|modelfile.txt
```

Abbildung 3: Aufbau einer Modeldatei zur Anpassung LLM spezifischer Parameter.

In der dargestellten Modeldatei wird als Grundlage für das angepasste Modell Llama3.1 verwendet. Die Systemanweisung (SYSTEM) wird präzisiert. Das Modell wird aufgefordert sich strikt an die übermittelten Kontexte zu halten. Es werden die Werte verschiedener Parameter angepasst, um die Ausgabe zu beeinflussen. Zum Schluss werden noch die Stop-Token genannt, welche die Ausgabe beenden, sobald sie im generierten Text erscheinen. Der Befehl, konkret der Pfad, zur Generierung des angepassten Modells muss angepasst werden je nachdem wo sich die Modeldatei (modelfile.txt) befindet. Das Wissen, dass die Ausgabe von LLM durch die Anpassung von Parametern beeinflusst werden kann, stellt eine wichtige Grundlage für Nutzer dar, wie Inhalte, die von einer KI generiert bewertet und kritisch hinterfragt werden sollten. Die Erkenntnis ist Bestandteil einer grundlegenden KI-Kompetenz. Die bereits jetzt schon verfügbaren Modelle bieten dadurch viele Möglichkeiten. Durch die Anpassung von ein paar Parametern steigen die Einsatzmöglichkeiten weiter, indem die Ausgabe eines LLM je nach Kontext angepasst werden kann. In folgendem Kapitel wird auf die zweite Möglichkeit eingegangen einen künstlichen Lernassistenten einzurichten.

6 INDIVIDUELL EDUCATIONAL CHATBOT

Die zweite Möglichkeit basiert auf einer weiterentwickelten Version des Individuell Educational Chatbots (IEC), der nach einer Erprobung mit Lehramtsstudenten weiter optimiert wurde und jetzt unabhängig von externen Frameworks eingesetzt werden kann [22]. Um die Anwendung lokal auszuführen, muss die Programmiersprache Python, mindestens Version 3.9, auf dem verwendeten System installiert sein. Python kann auf der offiziellen Seite (https://www.python.org/) heruntergeladen werden. Um zu prüfen, dass die Installation erfolgreich war, kann der Befehl python –version im Terminal verwendet werden. Wenn die installierte Version ausgegeben wird, hat alles funktioniert.

Die Anwendung kann über folgendes GitHub Repository heruntergeladen werden (Link). Um die App zu starten, müssen noch die Abhängigkeiten installiert werden, die in der Datei requirementx.txt aufgelistet sind. Mit dem Befehl pip install -r requirements.txt im Terminal, kann dies automatisch durchgeführt werden. Hierbei muss beachtet werden, dass der Befehl in dem Verzeichnis ausgeführt werden muss, indem sich die entsprechende Datei befindet. Da Ollama bereits installiert und ein LLM heruntergeladen wurde, kann die App mit dem Befehl python app.py gestartet werden. Vorher muss der Code noch angepasst werden, konkret die Datei app.py. Konkret geht es um das Modell, das man verwendet. Um die Datei bearbeiten zu können, empfiehlt es sich den Quellcode Editor Visual Studio Code zu installieren. Je nachdem wie man die Datei bearbeiten möchte, wird in der Zeile des Codes das Modell eingetragen [öllama", "run", "Modell"], dass die Eingaben des Nutzers verarbeiten soll. Wenn man beispielsweise llama3.1 von Ollama heruntergeladen hat, gibt man das entsprechend ein [öllama", "run", "llama3.1"]. Jetzt kann die Anwendung gestartet werden. Im Terminal sollte ein Link, standardmäßig http://127.0.0.1:5000/ angezeigt werden der angeklickt wird. Mithilfe des zweiten Links (http://192.168.1.23:5000) ist die Flask-App im gesamten Netzwerk erreichbar. Je nachdem welchen Link man anklickt, wird man zu folgender Oberfläche weitergeleitet (siehe Abbildung 4).

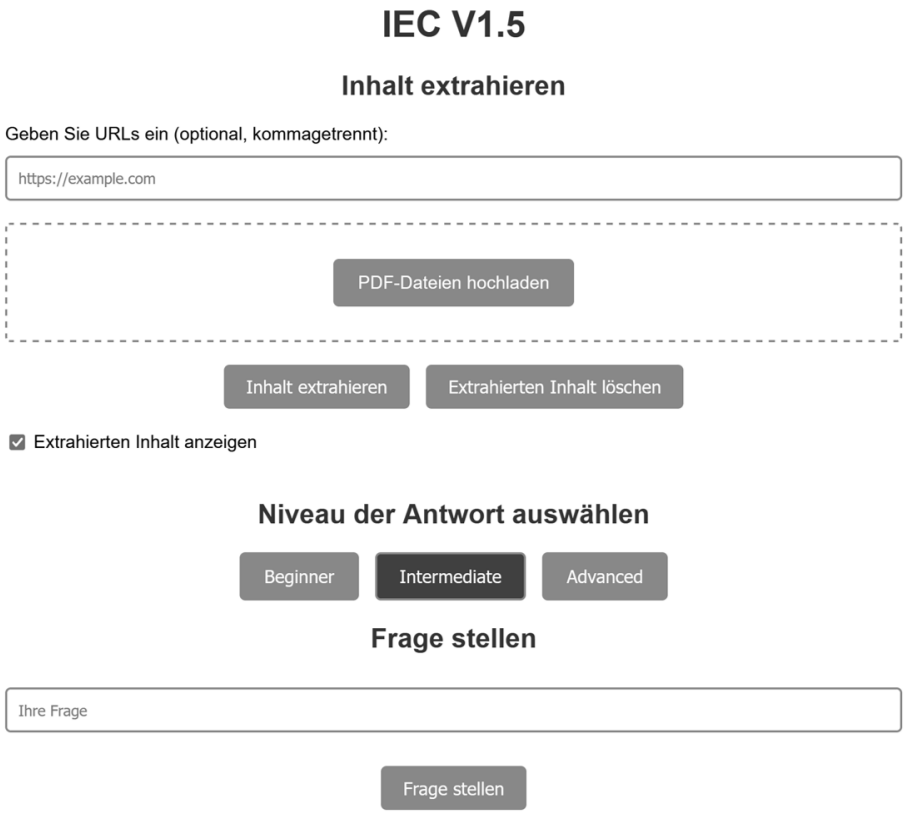

Abbildung 4: Aufbau des IEC-V1.5 Prototypen

Der Vorteil dieser Anwendung im Vergleich zur ersten Möglichkeit (Open WebUI) ist, dass hier das in Kapitel 5 beschriebene Problem nicht auftreten kann. Der Grund dafür ist, dass die Knowledge Base des LLM vom Nutzer selbst festgelegt wird [22]. Die Antwort des Modells basiert also nicht auf dem, worauf es trainiert wurde, sondern den Quellen, die vom Nutzer eingegeben werden. In die IEC App können die Nutzer in der aktuellen Version entweder Links von Internetseiten eintragen oder PDF-Dateien hochladen. Es ist auch möglich, Links und PDFs kombiniert als Quelle festzulegen. Der Inhalt der Eingaben wird gefiltert und bei Bedarf als Fließtext dargestellt. Der extrahierte Text dient als Grundlage für das Modell, um Fragen zu beantworten. Der Nutzer behält dadurch die Kontrolle über das, womit er sich befasst, indem er die Wissensbasis selbst definiert. Der IEC bietet zudem die Möglichkeit das Niveau der Antwort zu bestimmen. Aktuell gibt es drei Möglichkeiten, die aber auch vom Nutzer der App individuell über den Code angepasst werden können. Für die Anpassung sind keine Programmierkenntnisse erforderlich, sondern es müssen lediglich die Prompts in der Funktion generate_responses_from_blocks angepasst werden. Ein weiterer Vorteil ist, dass die Oberfläche des IEC im Vergleich zu Open WebUI nicht überladen ist mit verschiedenen Optionen. Dadurch ist der IEC besonders für Einsteiger gut geeignet.

Wenn die Quellen bestimmt und das Niveau ausgewählt wurde, kann man eine Frage stellen und erhält dann eine spezifische Antwort. Die Zeit, bis eine Antwort vom LLM ausgegeben wird, ist dabei davon abhängig, welches Modell man von Ollama verwendet und welche Rechenleistung das System hat, über das die App ausgeführt wird. Die Antwortgeschwindigkeit von proprietären Anwendungen ist schneller, aber auch nicht so groß, dass es einen wesentlichen Unterschied macht.

Im folgenden Kapitel werden didaktische Einsatzmöglichkeiten für die beiden beschriebenen Anwendungen vorgeschlagen.

7 DIDAKTISCHE EINSATZMÖGLICHKEITEN

Bildung ohne künstliche Barrieren bedeutet, dass alle Lernenden unabhängig von finanziellen Mitteln Zugang zu KI-gestützten Anwendungen haben. Hierfür sind zwei Aspekte entscheidend. Lernende und Lehrende müssen einfache Wege kennen, um Open-Source-KI-Modelle, wie die von Ollama, Hugging Face oder GPT4All, ohne große technische Einstiegshürden nutzen zu können. Neben dem technischen Zugang, der in diesem Beitrag beschrieben wurde, spielt die didaktische Integration eine entscheidende Rolle. Um die in Kapitel 2 genannten Kompetenzen zu fördern, müssen Lehrende lernen, wie sie KI gezielt in den Unterricht einbinden und didaktisch sinnvoll nutzen können [23].

Lernende können Open WebUI und IEC nutzen, um Texte zu analysieren, Feedback zu erhalten und Argumentationsstrukturen zu verbessern. Hierbei kann auch ausprobiert werden bei welchem LLM die besten Ergebnisse erzielt werden. Lehrende können Lernenden Texte bewerten lassen, die von einer KI generiert wurden. Beide Anwendungen können auch praxisnah eingesetzt werden, konkret in der Sprachanalyse oder in der Korrektur von Code im Informatikunterricht. Der IEC bietet aufgrund seiner Struktur zusätzlich die Möglichkeit einer personalisierten Nachhilfe, indem Materialien hochgeladen und Erklärungen passend zum ausgewählten Kompetenzniveau ausgegeben werden [22]. IEC und Open WebUI können zur Unterstützung von Rechercheaufgaben eingesetzt werden indem komplexe Texte zusammengefasst und Fachbegriffe erklärt werden [24]. Darüber hinaus können die Anwendungen auch als Reflexionsinstrument eingesetzt werden, um mit den Lernenden zu diskutieren, wie sich kommerzielle und offene KI-Systeme auf gesellschaftlich relevante Themen auswirken [25]. Auch das in Kapitel 5 beschriebene Problem der geschlossenen Wissensdatenbank vieler LLM und die damit verbundene Halluzination kann Thema von Diskussionsrunden sein. Das Ergebnis einer Analyse von 82 Artikeln aus dem Jahr 2024 mithilfe des FACTS Frameworks hinsichtlich der Fragestellung wie der Einsatz von KI die Bildung verändern wird, zeigte, dass in fast einem Drittel der Texte davon ausgegangen wird, dass KI das Lernen individueller gestalten wird [26]. Die hier genannten Beispiele sollen eine erste Anregung sein, wie ein didaktisch sinnvoll Einsatz aussehen könnte, um das zu erreichen.

8 ZUSAMMENFASSUNG

Die zunehmende Verbreitung von KI im Bildungsbereich stellt sowohl eine didaktische als auch gesellschaftliche Herausforderung dar. Während proprietäre Systeme den Zugang einschränken, können Open-Source Modelle eine echte Alternative für ein gerechtes Bildungssystem sein. Aus diesem Grund wurden in diesem Beitrag zwei Möglichkeiten beschrieben, wie man mit einfachen Mitteln künstliche Lernassistenten lokal einrichten, Modelle via Modelldateien modifizieren, aktuelle Modelle laden und dann verwenden kann. Hinsichtlich einer allgemeinen KI-Kompetenz sind diese Kenntnisse im Bereich Systemverständnis einzuordnen und bilden eine solide Grundlage für darauf aufbauende weitere Schritte. Die Tatsache, dass bereits viele LLMs über Ollama verfügbar sind und kontinuierlich neue sowie besser konfigurierte Modelle entwickelt und bereitgestellt werden, macht Open WebUI und IEC langfristig zu nützlichen Werkzeugen. Inzwischen gibt es auch erste Versuche spezialisierte LLM für den Bildungskontext zu entwickeln, da die meisten Modelle eher auf allgemeine Sprachverarbeitung trainiert sind.

Die rasante Entwicklung im Bereich KI sollte bestehende Ungleichheiten im Bildungssystem nicht weiter verstärken, sondern dazu beitragen, diese abzubauen. Bildungsinstitutionen müssen

mit diesen neuen Möglichkeiten sinnvoll und effektiv arbeiten. Dafür sind jedoch didaktische Konzepte und Fortbildungen erforderlich. Die in diesem Beitrag beschriebenen Anwendungen verdeutlichen, dass dafür keine kostspieligen Anschaffungen erforderlich sind. Unabhängig davon muss den Verantwortlichen in den verschiedenen Bildungsinstitutionen bewusst sein, dass für eine effektive Nutzung dieser Technologie eine kontinuierliche Auseinandersetzung erforderlich ist, um den Anschluss aufgrund der beschriebenen Entwicklungsdynamik nicht zu verlieren. Auch im Rahmen des Lehramtsstudiums ist es daher wichtig, Lehrveranstaltungen zu etablieren, die für alle verpflichtend sind und sich ausschließlich mit Künstlicher Intelligenz befassen. Anhand des Beitrags ergeben sich Fragen für konkrete weitere Forschung: Wie lassen sich Open-Source Modelle curricular in Schulen und Hochschulen verankern? Welche didaktischen Strategien sind am effektivsten, wenn es um den Erwerb von KI-Kompetenzen geht?

KI muss langfristig gesehen nicht nur als technisches Werkzeug betrachtet werden, sondern als integraler Bestandteil einer gerechten und offenen Bildungslandschaft.

9 ANMERKUNG

Der/die Autor(en) erhielt(n) keine finanzielle Unterstützung für die Forschung, Autorenschaft und/oder Veröffentlichung dieses Artikels.

LITERATURVERZEICHNIS

[1] Institut für Innovation und Technik. *Was bedeutet Open Source für Künstliche Intelligenz (KI)?* Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3iY.

[2] LCH. *Positionspapier: KI in der Schule – Chancen und Risiken für das Bildungssystem.* Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3j0.

[3] T. Knaus. „Künstliche Intelligenz und Bildung: Was sollen wir wissen? Was können wir tun? Was dürfen wir hoffen? Und was ist diese KI? Ein kollaborativer Aufklärungsversuch“. In: *Ludwigsburger Beiträge zur Medienpädagogik* 23 (2023). Abgerufen am 16. Mai 2025, S. 1–42. URL: https://urlz.fr/u3iZ.

[4] Digital David. *Die besten Open-Source-KI-Plattformen im Jahr 2024: Ein umfassender Leitfaden.* Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3iQ.

[5] Fraunhofer-Verbund IUK-Technologie. *Cloudbasierte KI-Plattformen.* Abgerufen am 16. Mai 2025. 2021. URL: https://urlz.fr/u3iX.

[6] M. Bucher. *Anwendungen von künstlicher Intelligenz in der Bildung – Chancen und Risiken.* Bachelorarbeit, Universität Zürich. Abgerufen am 16. Mai 2025. 2017. URL: https://urlz.fr/u3iN.

[7] Deloitte. *Künstliche Intelligenz im Bildungssektor: Herausforderungen und Chancen.* Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3iP.

[8] Europäisches Parlament. *EU AI Act: first regulation on artificial intelligence.* Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3iW.

[9] Plattform Lernende Systeme. *AI Act of the European Union.* Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3j1.

[10] Trail. *EU AI Act: Risk-Classifications of the AI Regulation.* Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3j2.

[11] White & Case LLP. *Long awaited EU AI Act becomes law after publication in the EU's Official Journal*. Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3j7.

[12] U.-D. Ehlers, M. Lindner und E. Rauch. *AIComp – Future Skills für eine durch KI geprägte Welt*. Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3iV.

[13] UNESCO. *AI competency framework for students*. Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3j5.

[14] A. Candel u. a. „H2O Open Ecosystem for State-of-the-art Large Language Models". In: *arXiv preprint* (2023). arXiv: 2310.13012.

[15] Yilong Zhao u. a. „Atom: Low-bit Quantization for Efficient and Accurate LLM Serving". In: *arXiv preprint* (2023). Abgerufen am 16. Mai 2025. arXiv: 2310.19102 [cs.LG]. URL: https://arxiv.org/abs/2310.19102.

[16] S. Pietrusky. *Optimize Open WebUI: Three practical extensions for a better user experience*. The Pythoneers. 2024.

[17] DeepSeek. *DeepSeek-R1 Release*. Abgerufen am 16. Mai 2025. 2025. URL: https://urlz.fr/u3iR.

[18] Vipula Rawte, Amit Sheth und Amitava Das. *A Survey of Hallucination in Large Foundation Models*. arXiv preprint arXiv:2309.05922. 2023. URL: https://arxiv.org/abs/2309.05922.

[19] Julien Siebert. *Halluzinationen von generativer KI und großen Sprachmodellen (LLMs)*. Fraunhofer IESE Blog. Blogpost veröffentlicht am 20. September 2024. Sep. 2024. URL: https://www.iese.fraunhofer.de/blog/halluzinationen-generative-ki-llm/.

[20] C.-C. Chang u. a. „KL-Divergence Guided Temperature Sampling". In: *arXiv preprint* (2023). arXiv: 2306.01286.

[21] A. Chornyi. *Understanding Temperature, Top-k, and Top-p Sampling in Generative Models*. Codefinity. Abgerufen am 16. Mai 2025. 2024. URL: https://urlz.fr/u3i0.

[22] S. Pietrusky. „Promoting AI Literacy in Higher Education: Evaluating the IEC-V1 Chatbot Prototype". In: *arXiv preprint* (2024). arXiv: 2412.16165.

[23] ZHAW. *Generative Künstliche Intelligenz – Integration in den Unterricht mit Hilfe von didaktischen Modellen*. Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3j9.

[24] BMBWF. *Lernen mit Künstlicher Intelligenz - Potential und Risiken von KI im Bildungssystem*. Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3iL.

[25] Wikimedia Deutschland. *Warum wir freie und offene KI in der Bildung brauchen*. Abgerufen am 16. Mai 2025. 2023. URL: https://urlz.fr/u3j8.

[26] S. Pietrusky. „Automatic answering of scientific questions using the FACTS-V1 framework: New methods in research to increase efficiency through the use of generative AI". In: *arXiv preprint* (2024). arXiv: 2412.16165.