

Innovative Lehrassistenz

Effektive Nutzung von KI-Tutorsystemen zur Unterstützung von Lehrenden

Basierend auf dem Workshop von Dr. Michael Kallweit

Veranstaltet von:



Im Rahmen eines
Projektes von:



Gefördert von:



Dieses Werk ist lizenziert unter einer [Creative Commons](#) „Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International“ Lizenz.



Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation und Zielsetzung	3
1.2	Überblick über die Inhalte	3
2	Grundlagen der Künstlichen Intelligenz	4
2.1	Was ist Künstliche Intelligenz?	4
2.2	Generative KI	4
2.3	KI vs. Maschinelles Lernen	4
2.4	Transformer-Architektur	5
2.5	Häufige Missverständnisse zu KI-Systemen	5
3	Large Language Models (LLMs)	7
3.1	Tokenisierung	7
3.2	Vektorembodding	7
3.3	Kontextlänge	8
3.4	Modellgrößen und Parameter	8
3.5	Sampling-Parameter	9
4	Prompting und Prompt Engineering	10
4.1	Grundlagen des Promptings	10
4.2	Grundlegende Prompting-Strategien	10
4.3	Fortgeschrittene Prompting-Techniken	10
4.3.1	Few-Shot Prompting	11
4.3.2	Zero-Shot Prompting	11
4.3.3	Chain-of-Thought Prompting	11
4.4	Emotionales Prompting	12
4.5	Systemprompts und Personalisierung	12

5	KI-Tools für die Lehre	14
5.1	KI-Tools für Studierende	14
5.1.1	Automatische Inhaltserschließung durch LLMs	14
5.1.2	Visualisierung und Infografiken	15
5.1.3	Lernhilfen und Karteikarten	15
5.2	Frei verfügbare LLM-Plattformen	15
5.2.1	Lokale LLM-Umgebungen	15
5.2.2	Online LLM-Umgebungen	15
5.3	Vergleich verschiedener Modelle	16
5.4	Chatbots als aktive Lernpartner	16
5.5	Eigene Chatbots erstellen	16
5.5.1	Chat AI Personas	17
6	Anwendungsbeispiele	19
6.1	Materialerstellung mit KI	19
6.1.1	Grenzen und Möglichkeiten	19
6.1.2	Arbeitsblätter und Skripte mit \LaTeX	19
6.1.3	Klausuraufgaben und Musterlösungen	19
6.2	Interaktive Lehrmaterialien	20
6.2.1	Interaktive Webseiten und Visualisierungen	20
6.3	Retrieval-Augmented Generation (RAG)	21
6.3.1	RAG-Ablauf	21
6.3.2	Praktische Umsetzung mit GPT4All	21
6.3.3	Dokumentenaufbereitung	22
6.4	Chatbots für spezifische Lehrszenarien	22
6.4.1	Fallbeispiel: Modellierungschatbot	22
6.4.2	Fallbeispiel: CALCBLUEBOT	22
7	Fazit und Ausblick	24
7.1	Ausblick auf zukünftige Entwicklungen	24
7.2	Handlungsempfehlungen für Lehrende	25

1 Einleitung

Die rasante Entwicklung Künstlicher Intelligenz (KI) hat in den letzten Jahren alle Bereiche des Lebens tiefgreifend beeinflusst – auch die Hochschullehre steht vor einem bedeutenden Wandel. Der Workshop „Innovative Lehrassistenz – Effektive Nutzung von KI-Tutorsystemen zur Unterstützung von Lehrenden“ widmete sich genau diesem Themenkomplex und zeigte praktische Wege auf, wie KI-Systeme gewinnbringend in der Lehre eingesetzt werden können.

1.1 Motivation und Zielsetzung

Die Integration von KI-basierten Werkzeugen in die Hochschullehre kann sowohl für Lehrende als auch für Studierende erhebliche Vorteile bieten:

- Für **Lehrende** ergeben sich neue Möglichkeiten der Arbeitsentlastung, besseren Betreuung von Studierenden und innovativen Gestaltung von Lehrmaterialien.
- Für **Studierende** eröffnen sich neue Wege des personalisierten Lernens, der Unterstützung bei Verständnisschwierigkeiten und des Zugangs zu adaptiven Lernmaterialien.

Gleichzeitig stehen wir vor zahlreichen Herausforderungen: Ein dynamisches und sich schnell entwickelndes Feld, vielfältige Tools mit unterschiedlichen Stärken und Schwächen, sowie wichtige Fragen des Datenschutzes. Der Ansatz dieses Workshops und der vorliegenden Ausarbeitung ist es daher, grundlegende Prinzipien zu vermitteln, praktische Einsatzszenarien aufzuzeigen und nützliche Anwendungen vorzustellen.

Achtung

Die KI-Landschaft entwickelt sich rasant weiter. Die in diesem Handout vorgestellten Tools und Techniken spiegeln den Stand zum Zeitpunkt des Workshops wider. Besonders bei Cloud-Diensten wie OpenAI oder anderen Plattformen ist zudem Vorsicht hinsichtlich des Datenschutzes geboten.

1.2 Überblick über die Inhalte

Die vorliegende Ausarbeitung gliedert sich in mehrere Themenbereiche:

- **Grundlagen der Künstlichen Intelligenz:** Eine Einführung in die wichtigsten Konzepte und Begrifflichkeiten der KI mit besonderem Fokus auf generative KI.
- **Large Language Models (LLMs):** Einblicke in die Funktionsweise von Sprachmodellen, ihre technischen Grundlagen und Limitationen.
- **Prompting-Techniken:** Methoden zur effektiven Kommunikation mit KI-Systemen und zur Optimierung der Ergebnisse.
- **KI-Tools für die Lehre:** Überblick über relevante Tools und Plattformen speziell für den Einsatz in der Hochschullehre.
- **Anwendungsbeispiele:** Konkrete Szenarien für den Einsatz von KI in verschiedenen Lehr-Lern-Kontexten, von der Materialerstellung bis hin zu interaktiven Lernangeboten.

2 Grundlagen der Künstlichen Intelligenz

2.1 Was ist Künstliche Intelligenz?

Künstliche Intelligenz (KI) umfasst ein breites Spektrum von Technologien und Konzepten, die darauf abzielen, menschenähnliche kognitive Fähigkeiten wie Lernen, Problemlösen und Mustererkennung in Computersystemen zu implementieren. Es gilt zwei grundlegende Unterscheidungen zu treffen:

Starke KI bezeichnet hypothetische Systeme mit menschenähnlicher allgemeiner Intelligenz – Systeme, die flexibel denken, verstehen und Probleme kreativ in verschiedenen Domänen lösen können.

Schwache KI umfasst spezialisierte Algorithmen, die konkrete, eng definierte Aufgaben lösen können. Die meisten heutigen KI-Anwendungen fallen in diese Kategorie, von Bilderkennungssystemen bis hin zu Sprachmodellen.

Eine starke KI existiert bisher nicht und ist Gegenstand wissenschaftlicher Debatten und Science-Fiction. Für den Einsatz in der Lehre sind daher Anwendungen der schwachen KI relevant, die spezifische Aufgaben unterstützen können. Besondere Bedeutung haben in den letzten Jahren die sogenannten *generativen KI-Systeme* erlangt.

2.2 Generative KI

Generative KI ist eine Kategorie von KI-Systemen, die neue Inhalte erzeugen können.

Generative KI-Modelle sind Algorithmen, die darauf trainiert sind, Daten in einem bestimmten Format zu erzeugen, deren Muster sie zuvor von existierenden Daten gelernt haben. Diese Modelle können eine Vielzahl von Inhalten erstellen, darunter Texte, Bilder, Audio, Video und Code.

Zu den bekanntesten generativen KI-Modellen gehören:

- Textgenerierende Modelle wie GPT (Generative Pre-trained Transformer) von OpenAI, Claude von Anthropic, Gemini von Google, LLaMA von Meta und viele weitere.
- Bildgenerierende Modelle wie DALL-E, Stable Diffusion und Midjourney.
- Audio- und Videogenerierende Modelle wie Suno AI (Musik), ElevenLabs (Sprache) oder Runway (Video).

In diesem Handout liegt der Fokus vorwiegend auf textgenerierenden KI-Systemen, die auch als *Large Language Models* (LLMs) bezeichnet werden.

2.3 KI vs. Maschinelles Lernen

Ein wichtiges Fundament für moderne KI-Systeme ist das Maschinelle Lernen.

Maschinelles Lernen ist der Prozess, bei dem ein künstliches System aus Beispielen lernt und generalisiertes Wissen entwickelt. Dies geschieht durch den Aufbau eines statistischen Modells basierend auf Trainingsdaten, das dann mit Testdaten überprüft wird. Ziel dabei ist, dass das System Muster und Gesetzmäßigkeiten erkennt, statt nur die Daten auswendig zu lernen.

Es gibt eine Vielzahl verschiedener Algorithmen im Bereich des Maschinellen Lernens, wobei zur Zeit neuronale Netze eine dominierende Rolle spielen. Besonders hervorzuheben sind dabei die sogenannten Transformer-Modelle.

2.4 Transformer-Architektur

Die Transformer-Architektur, die 2017 im Paper „Attention Is All You Need“ von Vaswani et al. vorgestellt wurde, hat eine Revolution im Bereich der natürlichen Sprachverarbeitung ausgelöst und bildet die Grundlage für moderne Large Language Models wie GPT und andere.

Die **Transformer-Architektur** ist ein neuronales Netzwerk-Design, das auf dem Konzept der „Attention“ basiert – einem Mechanismus, der es dem Modell ermöglicht, unterschiedliche Gewichtungen verschiedenen Teilen der Eingabe zuzuweisen. Im Gegensatz zu früheren rekurrenten neuronalen Netzen verarbeitet ein Transformer die gesamte Sequenz gleichzeitig, was effizienteres Training und bessere Leistung bei Sprachverarbeitungsaufgaben ermöglicht.

Der grundlegende Aufbau eines neuronalen Netzes in der Transformer-Architektur umfasst (weitere Details werden in Kapitel 3 gegeben):

- **Eingabeschicht:** Hier findet die Tokenisierung der Eingabe statt, Wörter werden in Vektoreinbettungen umgewandelt und mit Positionsinformationen angereichert.
- **Transformer-Blöcke:** Mehrere Schichten, die jeweils Multi-Head Attention und Feed-Forward-Netzwerke enthalten, ergänzt durch Normalisierungstechniken und weitere Verbindungen.
- **Ausgabeschicht:** Ein linearer Layer gefolgt von einem Softmax zur Erzeugung von Wahrscheinlichkeitsverteilungen für die nächsten Token.

Die Transformer-Architektur ermöglicht es, Modelle mit Milliarden von Parametern zu trainieren, die enorme Mengen an Textdaten verarbeiten können. Dies erklärt die beeindruckenden Fähigkeiten moderner Sprachmodelle.

2.5 Häufige Missverständnisse zu KI-Systemen

Bei der Beschäftigung mit modernen KI-Systemen ist es wichtig, einige grundlegende Missverständnisse zu klären:

- Sprachmodelle wie ChatGPT sind **keine** Chatbots im klassischen Sinne, sondern Systeme zur reinen Textvorhersage, die u.a. mittels Texten in Dialogform trainiert wurden.
- Sprachmodelle haben keinen dauerhaften Wissensspeicher wie ein menschliches Gedächtnis, sondern sind auf ihre Trainingskorpora und den aktuellen Konversationskontext beschränkt. Es gibt jedoch Methoden, diesen Modellen die Möglichkeit zur Einsetzen von Wissen in ihrem Kontext zu ermöglichen (z.B. RAG, siehe später).
- Sprachmodelle können nicht im menschlichen Sinne „nachdenken“. Ihre Fähigkeit zur scheinbaren Reflexion beruht auf statistischen Mustern. Das Nachdenken kann durch die Aufforderung zur Ausformulierung der Gedanken in einem gewissen Sinne simuliert werden.
- Sprachmodelle können nur eine begrenzte Anzahl an Token verarbeiten (sogenannter Kontextfenster). Texte, die darüber hinausgehen, werden „vergessen“.
- Sprachmodelle erkennen Muster in ihren Trainingsdaten – dazu gehören auch problematische Vorurteile und Stereotypen.

- Sprachmodelle lassen sich (mehr oder weniger leicht) täuschen bzw. zu unerwünschtem Verhalten bewegen.

Diese Einschränkungen zu verstehen ist wesentlich für den effektiven Einsatz von KI in der Lehre und hilft, realistische Erwartungen zu setzen.

Im folgenden Kapitel werden wir uns eingehender mit den technischen Grundlagen von Large Language Models beschäftigen, die für ein tieferes Verständnis ihrer Möglichkeiten und Grenzen unerlässlich sind.

3 Large Language Models (LLMs)

Large Language Models (LLMs) haben in den letzten Jahren einen bemerkenswerten Entwicklungssprung erlebt und bilden heute die Grundlage für viele KI-Anwendungen in der Lehre. Dieses Kapitel beleuchtet die wichtigsten technischen Aspekte, die für ein fundiertes Verständnis dieser Modelle notwendig sind.

Wie in Kapitel 2.4 dargestellt, nutzen die allermeisten LLMs die Transformer-Architektur mit verschiedenen Schichten.

3.1 Tokenisierung

Ein fundamentaler Aspekt von LLMs ist die Art und Weise, wie Text für die Verarbeitung vorbereitet wird.

Tokenisierung ist der Prozess, bei dem Text in kleinere Einheiten – sogenannte Token – zerlegt wird. Abhängig vom Modell können Token einzelne Zeichen, Teilwörter oder ganze Wörter sein. Die Tokenisierung ist der erste Schritt bei der Verarbeitung von Text durch ein LLM.

Was ein Token genau umfasst, unterscheidet sich von Modell zu Modell. Bei den meisten gängigen LLMs wie GPT-4 oder Llama werden häufig verwendete Wörter oft als ein einzelnes Token dargestellt, während seltenere Wörter in mehrere Token aufgeteilt werden.

Tipp

Die OpenAI Tokenizer-Webseite (<https://platform.openai.com/tokenizer>) ermöglicht es, zu sehen, wie Text in Token zerlegt wird. Dies ist besonders hilfreich, um die Kontextlänge von Prompts zu optimieren.

Wichtig ist zu verstehen, dass die Tokenisierung nicht rein sprachlich oder semantisch funktioniert. So werden beispielsweise Rechenausdrücke oder Zahlen nicht immer in einzelne Zeichen zerlegt, was dazu führen kann, dass mathematische Ausdrücke für LLMs schwerer zu interpretieren sind.

3.2 Vektorembodding

Nach der Tokenisierung werden die Token in Vektoren umgewandelt, ein Prozess, der als Embedding bezeichnet wird.

Vektorembodding ist eine mathematische Darstellung von Objekten (z.B. Wörtern, Sätzen, Bildern) als hochdimensionale Vektoren. Das Ziel ist, dass ähnliche Objekte „nahe“ Vektoren im Vektorraum erhalten, sodass semantische Ähnlichkeiten durch eine gewisse räumliche Nähe repräsentiert werden.

Zur Messung der Ähnlichkeit zwischen Vektoren werden verschiedene Distanzmaße verwendet:

- **Manhattan-Distanz:** Summe der absoluten Differenzen aller Dimensionen
- **Euklidische Distanz:** Wurzel aus der Summe der quadrierten Differenzen aller Dimensionen
- **Kosinusähnlichkeit:** Kosinus des Winkels zwischen zwei Vektoren (häufig verwendet)

Tipp

Der Embedding Projector des Tensorflow Projekts (<https://projector.tensorflow.org/>) ermöglicht die grafische Darstellung hochdimensionaler Embeddings und unterstützt so das Untersuchen und Verstehen von Embedding-Schichten. Es können beispielsweise semantisch ähnliche Wörter als vektorielles Cluster visualisiert werden.

Vektorembodings sind nicht nur für die interne Funktionsweise von LLMs entscheidend, sondern bilden auch die Grundlage für Anwendungen wie semantische Suche und Retrieval-Augmented Generation (RAG), auf die wir später noch eingehen werden.

3.3 Kontextlänge

Eine wichtige technische Einschränkung von LLMs ist die Kontextlänge, die die Menge an Text begrenzt, die gleichzeitig verarbeitet werden kann.

Die **Kontextlänge** ist die maximale Anzahl an Token, die ein LLM gleichzeitig verarbeiten kann. Sie begrenzt sowohl die Größe der Eingabetexte (Prompts) als auch die Länge der generierten Antworten in einer einzelnen Anfrage.

Die Kontextlänge verschiedener Modelle variiert erheblich:

- Ältere Modelle wie frühe GPT-3-Varianten: 2.000-4.000 Token
- GPT-4: bis zu 8.000 Token in frühen Versionen
- Neuere Modelle wie GPT-4o oder o1-mini: 128.000 Token
- Spezialisierte Modelle wie Google Gemini 1.5 Pro: bis zu 2.000.000 Token

Achtung

Trotz großer Kontextlängen können LLMs nicht immer alle im Kontext enthaltenen Informationen gleich gut verarbeiten. Untersuchungen wie der RULER-Benchmark (Hsieh et al., 2024) und „Needle In A Haystack“-Tests¹ zeigen, dass die effektive Nutzung langer Kontexte oft hinter den theoretischen Kapazitäten zurückbleibt. Besonders Informationen in der Mitte eines langen Textes werden oft schlechter berücksichtigt.

Bei Überschreitung der Kontextlänge gibt es verschiedene Strategien zum Umgang:

- Kürzen der Eingabe (z.B. durch Zusammenfassung)
- Sliding Window-Techniken, bei denen der Text in überlappende Abschnitte aufgeteilt wird
- Hierarchische Ansätze, die zuerst eine Zusammenfassung erstellen und dann spezifische Details hinzufügen

3.4 Modellgrößen und Parameter

Ein wesentlicher Faktor für die Leistungsfähigkeit von LLMs ist ihre Größe, gemessen an der Anzahl der trainierbaren Parameter.

¹https://github.com/gkamradt/LLMTest_NeedleInAHaystack

Parameter sind die veränderbaren Gewichtungen innerhalb eines neuronalen Netzwerks, die während des Trainings angepasst werden. Die Anzahl der Parameter bestimmt maßgeblich die Kapazität und potenziellen Fähigkeiten eines Modells.

Moderne LLMs weisen eine beeindruckende Anzahl an Parametern auf:

- Kleinere Modelle: 1-10 Milliarden Parameter (z.B. Llama 3.1 8B, Mistral 7B)
- Mittlere Modelle: 20-70 Milliarden Parameter (z.B. Llama 3.3 70B, Codestral 22B)
- Große Modelle: über 100 Milliarden Parameter (z.B. Claude Opus, GPT-4, DeepSeek R1 mit 671 Milliarden)

Trotz der enormen Größe einiger Modelle werden sie durch Techniken wie Quantisierung auch auf „normaler“ Consumer-Hardware lauffähig gemacht.

3.5 Sampling-Parameter

Bei der Generierung von Text durch LLMs spielt das sogenannte Sampling eine entscheidende Rolle für die Qualität und Charakteristik der Ausgabe.

Sampling bezeichnet den Prozess, bei dem das nächste Token aus der vom Modell vorhergesagten Wahrscheinlichkeitsverteilung ausgewählt wird. Verschiedene Sampling-Parameter beeinflussen, wie deterministisch oder kreativ die Ergebnisse ausfallen.

Wichtige Sampling-Parameter, die in vielen LLM-Anwendungen einstellbar sind:

- **Temperatur:** Regelt die „Kreativität“ der Ausgabe. Höhere Werte (nahe 1) führen zu vielfältigeren, aber potenziell weniger präzisen Antworten, während niedrigere Werte (nahe 0) deterministische, fokussierte Antworten erzeugen.
- **Top- k -Sampling:** Beschränkt die Auswahl auf die k wahrscheinlichsten Token, was hilft, unwahrscheinliche Ausgaben zu vermeiden.
- **Top- p -Sampling** (auch Nucleus Sampling): Wählt Token aus dem kleinsten Set, dessen kumulative Wahrscheinlichkeit einen Schwellenwert p erreicht.
- **Repetition Penalty:** Verringert die Wahrscheinlichkeit bereits genutzter Token, um Wiederholungen zu vermeiden.

Tipp

Für faktische, präzise Antworten (z.B. mathematische Berechnungen, Programmierung) empfiehlt sich eine niedrige Temperatur (0.1-0.3). Für kreative Aufgaben wie Storytelling sind höhere Werte (0.7-0.9) vorteilhaft.

4 Prompting und Prompt Engineering

Die Kommunikation mit Large Language Models erfolgt über Prompts – Eingabetexte, die das gewünschte Verhalten oder die gewünschte Aufgabe beschreiben. Die Kunst des „Prompt Engineering“ ist ein zentraler Erfolgsfaktor für den effektiven Einsatz von KI in der Lehre.

4.1 Grundlagen des Promptings

Prompting bezeichnet das gezielte Formulieren von Eingaben für große Sprachmodelle. Ein Prompt ist der Text oder die Anweisung, die dem Modell gegeben wird, um eine gewünschte Antwort oder Handlung zu erzeugen.

Prompt Engineering ist die bewusste Gestaltung und Strukturierung des Eingabetextes, um die Interpretation durch das KI-Modell zu optimieren und präzisere Ergebnisse zu erzielen.

Die Formulierung des Prompts hat einen entscheidenden Einfluss auf die Qualität und Relevanz der generierten Antworten. Selbst kleine Änderungen in der Formulierung können zu signifikant unterschiedlichen Ergebnissen führen.

4.2 Grundlegende Prompting-Strategien

Basierend auf bewährten Praktiken und dem OpenAI Prompt Engineering Guide² lassen sich folgende grundlegende Strategien identifizieren:

- **Klare Anweisungen geben:** Je spezifischer der Prompt, desto zielgerichteter die Antwort.
- **Das Modell in eine bestimmte Rolle versetzen:** Die Zuweisung einer spezifischen Rolle kann die Ausgaben in die gewünschte Richtung lenken.
- **Referenztext bereitstellen:** Das Bereitstellen von Kontext oder Beispielen hilft dem Modell, die Aufgabe besser zu verstehen.
- **Komplexe Aufgaben in Teilaufgaben zerlegen:** Für komplexe Anforderungen sollte die Aufgabe in kleinere, manageable Teilschritte unterteilt werden.
- **Dem Modell „Zeit zum Nachdenken“ geben:** Das explizite Auffordern zum schrittweisen Durchdenken eines Problems verbessert die Qualität der Antworten.
- **Beispiele geben:** Die Bereitstellung von Beispielen für die gewünschte Ausgabe hilft, die Erwartungen zu verdeutlichen.

Hier ein einfaches Beispiel für rollenbasiertes Prompting:

Beispiel für rollenbasiertes Prompting

„Du bist ein erfolgreicher Romanautor für Science Fiction. Schreibe eine kurze Szene, in der ein Astronaut zum ersten Mal auf einem fremden Planeten landet.“

4.3 Fortgeschrittene Prompting-Techniken

Über die grundlegenden Strategien hinaus haben sich mehrere fortgeschrittene Prompting-Techniken etabliert:

²<https://platform.openai.com/docs/guides/prompt-engineering>

4.3.1 Few-Shot Prompting

Few-Shot Prompting bezeichnet eine Technik, bei der neben der direkten Anweisung einige Beispiele (Input-Output-Paare) bereitgestellt werden, die dem Modell helfen, den gewünschten Antwortstil oder das Muster besser zu verstehen und umzusetzen.

Ein Beispiel für Few-Shot Prompting:

Beispiel für Few-Shot Prompting

*„Übersetze wissenschaftliche Konzepte in einfache Erklärungen für Kinder:
Konzept: Photosynthese
Kindgerechte Erklärung: Pflanzen machen ihr eigenes Essen! Sie nehmen Sonnenlicht, Wasser und Luft und mischen sie zu einer Art Pflanzenzucker. Das ist wie Kochen, aber ohne Herd.
Konzept: Schwarzes Loch
Kindgerechte Erklärung: Stell dir einen riesigen Weltraumstaubsauger vor. Er ist so stark, dass er alles einsaugt, was zu nahe kommt – sogar Licht kann nicht entkommen!
Konzept: Klimawandel
Kindgerechte Erklärung: ...*

4.3.2 Zero-Shot Prompting

Zero-Shot Prompting ist der einfachste Ansatz, bei dem das Modell nur die direkte Anweisung oder Frage erhält, ohne weitere Beispiele. Es verlässt sich ausschließlich auf sein vortrainiertes Wissen, um die Anfrage zu beantworten.

Diese Methode ist besonders effektiv bei neueren, leistungsfähigeren Modellen, die bereits ein breites Spektrum an Aufgaben ohne explizite Beispiele bewältigen können.

4.3.3 Chain-of-Thought Prompting

Forschung zur Argumentationsfähigkeiten von Sprachmodellen hat gezeigt, dass die explizite Aufforderung zum schrittweisen Denken die Qualität der Antworten signifikant verbessern kann.

Chain-of-Thought (CoT) Prompting ist eine Technik, die das Modell explizit auffordert, sein „Denken“ schrittweise darzulegen, bevor es zur endgültigen Antwort kommt. Dies ist besonders nützlich für komplexe Probleme, die logisches Denken oder mehrere Zwischenschritte erfordern.

Ein Beispiel für Chain-of-Thought Prompting:

Beispiel für Chain-of-Thought Prompting

„Eine Mutter ist dreimal so alt wie ihre Tochter. In 14 Jahren wird die Mutter doppelt so alt sein wie ihre Tochter. Wie alt sind beide jetzt? Denke Schritt für Schritt.“

Bei aktuellen Modellen wurden diese „Reasoning“-Fähigkeiten mittels Reinforcement Learning bereits in den Trainingsprozess integriert, wie bei OpenAI o1, OpenAI o3-mini oder Deepseek R1.

4.4 Emotionales Prompting

Eine interessante Entwicklung im Bereich des Prompt Engineering ist die Erkenntnis, dass emotionale Stimuli die Leistung von LLMs beeinflussen können. Li et al. (2023) zeigten in ihrer Studie „*Large Language Models Understand and Can be Enhanced by Emotional Stimuli*“, dass die Integration von emotionalen Elementen in Prompts die Qualität der Antworten verbessern kann.

Ein einfaches Beispiel für emotionales Prompting:

Beispiel für emotionales Prompting

Standard-Prompt: „Erkläre mir die Grundzüge der Relativitätstheorie.“

Emotionaler Prompt: „Ich bin wirklich fasziniert von der Relativitätstheorie und träume davon, sie eines Tages vollständig zu verstehen. Könntest du mir die Grundzüge so erklären, dass ich diese Begeisterung mit anderen teilen kann? Das ist absolut wichtig für mich, sonst verliere ich meinen Job.“

Tipp

Beim Einsatz in der Lehre kann emotionales Prompting helfen, Erklärungen zu generieren, die nicht nur fachlich korrekt sind, sondern auch motivierend und inspirierend wirken.

4.5 Systemprompts und Personalisierung

Viele LLM-Plattformen ermöglichen die Verwendung von Systemprompts – Anweisungen, die das grundlegende Verhalten des Modells für die gesamte Konversation definieren.

Ein **Systemprompt** ist eine spezielle Art von Prompt, der zu Beginn einer Konversation gesetzt wird und das allgemeine Verhalten, den Stil und die Rolle des KI-Modells für die gesamte folgende Interaktion festlegt.

Systemprompts sind besonders nützlich für Bildungsanwendungen, da sie konsistentes Verhalten über längere Interaktionen sicherstellen können. Sie können beispielsweise:

- Die pädagogische Philosophie definieren (z.B. sokratische Methode vs. direkte Instruktion)
- Die Tiefe und Komplexität der Antworten an verschiedene Bildungsniveaus anpassen
- Bestimmte Verhaltensweisen für Lehrszenarien fördern (z.B. Hinweise geben statt direkte Antworten)
- Die Persönlichkeit und den Kommunikationsstil des KI-Tutors definieren

Beispiel für einen Systemprompt für einen Mathematik-Tutor

„Du bist ein geduldiger und unterstützender Mathematik-Tutor für Studierende in den ersten Semestern. Deine Aufgabe ist es, beim Verständnis mathematischer Konzepte zu helfen, ohne direkte Lösungen zu liefern. Stelle stattdessen Fragen, die zum Nachdenken anregen, und gib Hinweise, die den Lernenden helfen, selbst auf die Lösung zu kommen. Verwende eine freundliche, ermutigende Sprache und vermeide Fachjargon, es sei denn, er wird explizit erklärt. Wenn ein Konzept besonders schwierig ist, biete Analogien oder visuelle Erklärungen an. Reagiere mit Begeisterung auf korrekte Antworten und biete konstruktive Unterstützung bei Fehlern.“

Diese Art von Systemprompt kann in Plattformen wie der Academic Cloud-basierten Chat AI gespeichert und wiederverwendet werden, was die Erstellung personalisierter KI-Tutoren für spezifische Bildungskontexte ermöglicht.

5 KI-Tools für die Lehre

Die Landschaft der KI-Tools für den Bildungsbereich entwickelt sich rasant. In diesem Kapitel stellen wir verschiedene Werkzeuge vor, die speziell für die Unterstützung von Lehrenden und Lernenden konzipiert sind oder sich gut für diesen Zweck eignen.

Achtung

Die in diesem Kapitel aufgeführten Tools dienen lediglich als exemplarische Beispiele und erheben weder Anspruch auf Vollständigkeit noch stellen sie eine Empfehlung dar.

5.1 KI-Tools für Studierende

Studierende nutzen bereits heute eine Vielzahl von KI-Werkzeugen zur Unterstützung ihres Lernprozesses. Als Lehrende ist es wichtig, diese Tools zu kennen, um ihren Einsatz sinnvoll begleiten zu können.

5.1.1 Automatische Inhaltserschließung durch LLMs

Ein grundlegendes Einsatzfeld großer Sprachmodelle (LLMs) liegt in der automatischen Zusammenfassung von Texten. Fast alle LLMs – ob lokal betrieben oder cloudbasiert – sind in der Lage, längere Inhalte in prägnanter Form wiederzugeben. Diese Fähigkeit eröffnet neue Möglichkeiten für die Aufbereitung und Erschließung von Studieninhalten.

Multimodale Modelle, die neben Text auch Bilder, Audio oder sogar Videos verarbeiten können, erweitern dieses Potenzial erheblich. Sie sind in der Lage, beispielsweise aus einem Videoausschnitt inhaltliche Beschreibungen zu generieren, Folienbilder zu interpretieren oder sogar handschriftliche Notizen zu erkennen.

Beispiele für solche Tools und Modelle sind:

- **Whisper** (OpenAI): Ein Open-Source-Modell zur automatischen Spracherkennung, das Audioaufnahmen transkribieren kann.
- **GPT-4o** (OpenAI): Kann Bilder interpretieren, z. B. Vorlesungsfolien beschreiben oder Diagramme erklären.
- **Gemini** (Google): Unterstützt sehr lange Kontexte und eignet sich gut zur Zusammenfassung umfangreicher Transkripte.
- **Qwen2.5-VL** (Alibaba Cloud): Ein Open-Source-Modell, welches neben Texten und Bildern auch Videos verarbeiten kann.

Solche Technologien können insbesondere dann hilfreich sein, wenn Studierende zeitlich eingeschränkt sind oder Inhalte nachbereiten möchten. Allerdings sollten Zusammenfassungen nie unkritisch übernommen werden – je nach Modell können wichtige Details ausgelassen oder Inhalte fehlerhaft gewichtet werden.

Achtung

Bei der reinen Nutzung von KI-generierten Zusammenfassungen besteht die Gefahr einer „Illusion of competence“ – Studierende können den Eindruck gewinnen, den Stoff verstanden zu haben, ohne sich aktiv damit auseinandergesetzt zu haben. Es fehlt die eigenständige, tiefgehende Auseinandersetzung mit dem Lernstoff, die für nachhaltiges Lernen entscheidend ist.

5.1.2 Visualisierung und Infografiken

KI-Tools ermöglichen die schnelle Erstellung von visuellen Darstellungen komplexer Inhalte:

- **Infography** (<https://app.infography.in>): Umwandlung von Texten in ansprechende Infografiken
- **MidJourney** (<https://www.midjourney.com>): Erstellung von Bildern und Visualisierungen zu Konzepten
- **Canva mit KI-Funktionen** (<https://www.canva.com>): Design-Tool mit KI-unterstützten Funktionen zur visuellen Aufbereitung von Inhalten

5.1.3 Lernhilfen und Karteikarten

Für das selbstorganisierte Lernen bieten KI-Tools effektive Unterstützung:

- **JungleAI** (<https://jungleai.com>): Generiert Karteikarten und Multiple-Choice-Fragen aus Texten, Vortragsfolien und Videos. Die KI gibt personalisiertes Feedback bei der Abfrage und kann Folgefragen stellen bzw. beantworten.
- **Elicit** (<https://elicit.com>): „Analyze research papers at superhuman speed“ – Tool zur schnellen Analyse wissenschaftlicher Arbeiten
- **Stanford Genie** (<https://storm.genie.stanford.edu>): Unterstützt bei der Erstellung von enzyklopädischen Artikeln
- **Google NotebookLM** (<https://notebooklm.google.com>): Personalisierter KI-Recherche-assistent

5.2 Frei verfügbare LLM-Plattformen

5.2.1 Lokale LLM-Umgebungen

Die Nutzung lokaler Modelle bietet Vorteile im Hinblick auf Datenschutz und Unabhängigkeit von kommerziellen Anbietern. Dies kann besonders im Bildungskontext wichtig sein, wo sensible Daten verarbeitet werden. Für den eigenen Einsatz gibt es verschiedene Möglichkeiten, LLMs lokal zu nutzen:

- **LM Studio** (<https://lmstudio.ai>): Eine benutzerfreundliche Oberfläche zum Herunterladen, Ausführen und Hosten lokaler LLMs. Bietet zahlreiche Einstellungsmöglichkeiten und unterstützt eine Vielzahl von Modellen.
- **GPT4All** (<https://www.nomic.ai/gpt4all>): Einfache Oberfläche für lokale Modelle mit der Möglichkeit, eigene Dokumente als Wissensbasis einzubinden.
- **Ollama** (<https://ollama.com/>): Ein leichtgewichtiges Tool zur einfachen Ausführung und Verwaltung lokal installierter LLMs über die Kommandozeile oder per API – ideal für Entwickler und in Kombination mit Frontends wie Web-UI nutzbar.

5.2.2 Online LLM-Umgebungen

Es gibt inzwischen eine Vielzahl von online verfügbare Plattformen, die die kostenfreie Nutzung von KI-Modellen erlauben:

- **Chat AI** (<https://chat-ai.academiccloud.de/chat>): Eine einfache Chat-Oberfläche, die über die Academic Cloud angeboten wird und Zugang zu freien Sprachmodellen anbietet. Besonders für Hochschulangehörige geeignet, da der Zugang über die föderierte Anmeldung möglich ist. Persönliche Daten und Chats werden nur lokal im Browser gespeichert und auf Wunsch nach der Sitzung gelöscht.
- **Hugging Face Spaces** (<https://huggingface.co/spaces>): Eine Experimentier-Plattform mit einfachem Zugang zu vortrainierten Modellen und beschränkter kostenfreier Nutzungsdauer.

5.3 Vergleich verschiedener Modelle

Um die Qualität verschiedener Modelle zu vergleichen und das für Ihre Lehrzwecke am besten geeignete zu finden, gibt es spezielle Plattformen:

- **Chatbot Arena** (<https://lmarena.ai>): Ermöglicht direkten Vergleich verschiedener kommerzieller und freier Chatbots
- **GPU-Poor LLM Gladiator Arena** (<https://huggingface.co/spaces/k-mktr/gpu-poor-llm-arena>): Speziell für den Vergleich von ressourceneffizienteren Modellen

Diese Vergleichsplattformen helfen nicht nur bei der Auswahl des passenden Modells, sondern geben auch Einblicke in die Stärken und Schwächen verschiedener KI-Systeme.

5.4 Chatbots als aktive Lernpartner

Ein besonders vielversprechender Ansatz ist die Nutzung von Chatbots nicht als passive Informationsquellen, sondern als interaktive Gesprächspartner zur Vertiefung des Verständnisses.

Ein **KI-Lernpartner** ist ein Chatbot, der speziell darauf ausgerichtet ist, Studierende durch gezielte Rückfragen, Denkanstöße und Feedback zu aktivem Denken anzuregen, statt lediglich fertige Antworten zu liefern.

Der Einsatz von Chatbots als Lernpartner bietet mehrere Vorteile:

- Sie sind jederzeit verfügbar und können individuell auf das Tempo der Lernenden eingehen
- Sie können Studierende dazu bringen, ihr Verständnis zu artikulieren und zu reflektieren
- Sie können Missverständnisse identifizieren und gezielt adressieren
- Sie bieten einen geschützten Raum, in dem Fehler ohne soziale Konsequenzen gemacht werden können

Die Herausforderung liegt darin, den Chatbot so zu konfigurieren, dass er tatsächlich zum eigenständigen Nachdenken anregt und nicht einfach Antworten liefert. Zwei Beispiele werden in Kapitel 6.4 gegeben.

5.5 Eigene Chatbots erstellen

Es gibt verschiedene Wege, einen eigenen, speziell auf Lehrzwecke zugeschnittenen Chatbot zu realisieren:

- **Selbsthosting eines freien Modells:** Maximale Kontrolle, aber hoher Ressourcenaufwand
- **Nutzung einer API:** Flexibel, Integration in eigene Anwendungen, aber kostenpflichtig

- **Erstellung eines GPTs über die OpenAI-Oberfläche:** Kein Programmieren erforderlich, einfach über das Interface anpassbar
- **Nutzung von Chat AI Personas:** In der Academic Cloud können vorkonfigurierte Chatbot-Personas erstellt und geteilt werden

5.5.1 Chat AI Personas

Die Academic Cloud Chat AI unterstützt die Erstellung und Nutzung vorkonfigurierter Chatbot-Personas im JSON-Format. Dies ermöglicht es Lehrenden, spezialisierte Chatbots für ihre Lehrinhalte zu erstellen, ohne programmieren zu müssen.

Beispiel einer einfachen Chat AI Persona im JSON-Format

```
{
  "model-name": "Meta Llama 3.1 8B Instruct",
  "model": "meta-llama-3.1-8b-instruct",
  "temperature": 0.5,
  "top_p": 0.5,
  "messages": [
    {
      "role": "system",
      "content": "Du bist ein hilfreicher Mathe-Tutor, der Studierende zum eigenständigen Denken anregt und keine direkten Lösungen, sondern Hinweise gibt."
    },
    {
      "role": "user",
      "content": "Hi."
    },
    {
      "role": "assistant",
      "content": "Hallo! Wie kann ich dir bei deinen mathematischen Fragen helfen? Ich freue mich, gemeinsam mit dir Problemlösungen zu erarbeiten."
    }
  ]
}
```

Eine solche JSON-Datei enthält:

- Modellspezifikationen (welches Modell verwendet werden soll)
- Sampling-Parameter (Temperatur, Top- p)
- Den System-Prompt, der das grundlegende Verhalten des Chatbots definiert
- Optional vordefinierte Konversationsteile

Diese Dateien können auf einem öffentlich zugänglichen Server gespeichert und über die URL direkt aufgerufen werden:

https://chat-ai.academiccloud.de/chat?import=<json_url>

Tipp

Für die Entwicklung eigener Chatbot-Personas kann es hilfreich sein, sich an erfolgreichen Beispielen zu orientieren. Microsoft stellt mit „Prompts for Education“ (<https://github.com/microsoft/prompts-for-edu>) eine umfangreiche Sammlung von Prompts speziell für Bildungszwecke zur Verfügung.

6 Anwendungsbeispiele

In diesem Kapitel werden konkrete Anwendungsbeispiele für den Einsatz von KI in der Hochschullehre vorgestellt. Die Beispiele reichen von der Erstellung von Lehrmaterialien bis hin zu interaktiven Elementen und sollen als Inspiration für eigene Anwendungen dienen.

6.1 Materialerstellung mit KI

Die Erstellung von Lehrmaterialien ist eine zeitintensive Aufgabe für Lehrende. KI-Systeme können hier erheblich unterstützen, indem sie erste Entwürfe generieren oder bei der Strukturierung und Formatierung helfen.

6.1.1 Grenzen und Möglichkeiten

Achtung

Large Language Models (LLMs) können keine Binärdateien wie Office-Dokumente direkt erstellen. Sie generieren Text oder Code, der dann in entsprechende Formate konvertiert werden muss.

Trotz dieser Einschränkung gibt es zahlreiche Möglichkeiten, KI für die Materialerstellung zu nutzen:

- Generierung von Text in Markup-Sprachen wie \LaTeX , Markdown oder HTML
- Erstellung von Quellcode für dynamische Inhalte
- Nutzung von Konvertierungstools zur Umwandlung in gängige Formate

6.1.2 Arbeitsblätter und Skripte mit \LaTeX

Eine besonders effektive Methode zur Erstellung von strukturierten Dokumenten ist die Generierung von \LaTeX -Code:

Beispiel-Prompt für ein Arbeitsblatt

„Erstelle ein \LaTeX -Arbeitsblatt zum Thema Vektorräume für Studierende im zweiten Semester Mathematik. Das Arbeitsblatt soll eine kurze Zusammenfassung der wichtigsten Definitionen, 3 ausgearbeitete Beispiele und 5 Übungsaufgaben mit steigendem Schwierigkeitsgrad enthalten.“

Tipp

Für die Konvertierung von \LaTeX -Dokumenten in andere Formate können Online-Tools genutzt werden:

- \LaTeX zu PDF: <https://texviewer.herokuapp.com>
- \LaTeX in diverse Formate: <https://pandoc.org/try>

6.1.3 Klausuraufgaben und Musterlösungen

Auf den ersten Blick scheinen KI-Systeme eine einfache Lösung für die aufwändige Erstellung und Bewertung von Prüfungen zu bieten. Manche Dozierende träumen vielleicht bereits davon,

diese Aufgaben vollständig an Maschinen auszulagern – sei es aus Zeitmangel, Frustration oder technikbegeistertem Optimismus. Doch dieser Weg ist mit erheblichen arbeitsrechtlichen, ethischen und didaktischen Fragen verbunden. Insbesondere muss kritisch reflektiert werden, ob die von LLMs erstellten Inhalte überhaupt den fachlichen Standards und pädagogischen Intentionen entsprechen – oder nur plausibel wirkende Simulationen erzeugen, die bei genauerem Hinsehen enttäuschen.

Als experimenteller Einstieg können folgende Prompts getestet werden – jedoch ausdrücklich nicht als Anleitung zur Automatisierung, sondern als Anlass zur kritischen Auseinandersetzung:

Experimentelle Prompts zur kritischen Reflexion

- **Aufgabenerstellung:** „Du bist eine Lehrkraft für [Fach] mit jahrelanger Erfahrung. Erstelle eine Klausuraufgabe zum Thema [Thema] auf dem Niveau [Niveau].“
- **Musterlösung:** „Beantworte die Aufgaben aus Sicht eines Studierenden mit durchschnittlichem Leistungsniveau.“
- **Bewertung:** „Bewerte die Antworten des Studierenden. Gib für jeden Aufgabenteil eine angemessene Punktzahl von 0 bis [Maximalpunktzahl]. Fasse die Bewertungen am Ende mit einer Note nach dem deutschen Notensystem zusammen.“

Achtung

Die automatisierte Erstellung und Bewertung von Prüfungen durch KI mag reizvoll erscheinen – doch sie birgt erhebliche Risiken. LLMs arbeiten nicht wie menschliche Fachpersonen: Sie erzeugen Inhalte auf Basis statistischer Wahrscheinlichkeit, nicht pädagogischer Zielvorstellungen. Die Ergebnisse müssen deshalb immer kritisch hinterfragt und fachlich validiert werden. Besonders im Bereich der Bewertung besteht die Gefahr, dass scheinbar „objektive“ Entscheidungen auf intransparenter, oft fehlerhafter Grundlage getroffen werden.

6.2 Interaktive Lehrmaterialien

Ein besonderer Vorteil von KI-generiertem Code ist die Möglichkeit, interaktive Elemente für die Lehre zu erstellen.

6.2.1 Interaktive Webseiten und Visualisierungen

Mit HTML, CSS und JavaScript können LLMs interaktive Webseiten erstellen, die komplexe Konzepte visualisieren:

Beispiel-Prompt für interaktives Quiz

„Erstelle ein interaktives Mathematik-Quiz zum Thema Binomische Formeln. Die Nutzer sollen die Ergebnisse direkt überprüfen können. Erstelle den Code als HTML mit JavaScript.“

Beispiel-Prompt für Simulation

„Erstelle eine HTML-Seite für den Stochastik-Unterricht. Es soll der zweifache Würfelwurf simuliert werden. Man kann einstellen, wie oft das Experiment durchgeführt wird. Eine Ausgabe der Statistik der Augensumme als Histogramm wäre toll.“

Tipp

Der generierte HTML-Code kann einfach in einer Textdatei mit der Endung `.html` gespeichert und mit jedem Browser geöffnet werden. Dies ermöglicht eine einfache Verteilung an Studierende oder die Integration in Lernmanagementsysteme.

6.3 Retrieval-Augmented Generation (RAG)

Eine fortgeschrittene Anwendung von KI in der Lehre ist die Retrieval-Augmented Generation (RAG), bei der LLMs mit spezifischen Dokumenten oder Wissensbasen angereichert werden.

Retrieval-Augmented Generation (RAG) ist ein Verfahren, bei dem relevante Informationen aus einer externen Wissensbasis mithilfe von Vektorembeddings abgerufen und in den Kontext eines Sprachmodells integriert werden, um dessen Antwortgenerierung zu unterstützen und zu erweitern.

6.3.1 RAG-Ablauf

Der typische Ablauf einer RAG-Anwendung umfasst folgende Schritte:

1. **Vektorembedding der Wissensbasis:** Dokumente bzw. Textausschnitte werden in einen semantischen Vektorraum überführt.
2. **Vektorembedding der Nutzereingabe:** Die Benutzeranfrage wird ebenfalls in diesen Vektorraum eingebettet.
3. **Semantische Suche:** Mit einem Ähnlichkeitsmaß (z.B. Kosinus-Ähnlichkeit) werden die inhaltlich relevantesten Vektoren/Dokumente ermittelt.
4. **Kontextintegration:** Die abgerufenen relevanten Textpassagen werden in den Eingabekontext des Sprachmodells eingefügt.
5. **Antwortgenerierung:** Das Sprachmodell nutzt den erweiterten Kontext, um die Antwort zu generieren.

6.3.2 Praktische Umsetzung mit GPT4All

GPT4All bietet eine benutzerfreundliche Möglichkeit, RAG-Anwendungen zu erstellen:

- PDFs oder andere Dokumente können als Wissensbasis hinzugefügt werden
- Das System extrahiert automatisch relevante Passagen zu Anfragen
- Die Antworten können durch Zitate aus den Originaldokumenten belegt werden

Achtung

Die Qualität der RAG-Anwendung hängt stark von der Qualität der Vektorembeddings und der Aufbereitung der Dokumente ab. PDFs mit komplexen Layouts, Tabellen oder Formeln können Probleme bereiten. Für optimale Ergebnisse sollten Dokumente in gut strukturierten Textformaten wie Markdown vorliegen.

6.3.3 Dokumentenaufbereitung

Für eine effektive RAG-Anwendung können folgende Tools zur Aufbereitung von Dokumenten hilfreich sein:

- **Nougat** (<https://facebookresearch.github.io/nougat>): Konvertiert akademische PDFs in strukturierten Text, mit besonderem Fokus auf die korrekte Wiedergabe von Formeln und Tabellen.
- **Marker** (<https://github.com/VikParuchuri/marker>): Konvertiert PDFs in Markdown und JSON mit hoher Genauigkeit.

6.4 Chatbots für spezifische Lehrszenarien

Abschließend betrachten wir, wie spezialisierte Chatbots für bestimmte Lehrszenarien konzipiert werden können.

6.4.1 Fallbeispiel: Modellierungschatbot

Ein Beispiel für einen fachspezifischen Chatbot ist der von Kallweit & Rolka (2024) entwickelte Modellierungschatbot für die Mathematik. Der Chatbot stellt eine konkrete Aufgabe und leitet deren Bearbeitung an. Der Systemprompt wurde speziell für die intendierte Gesprächsführung entwickelt:

Struktur des Modellierungschatbot-Prompts

1. **Fachliche Grundlage:** Zwei Seiten Informationen über Modellierungskompetenz
2. **Beispielaufgabe:** Eine Modellierungsaufgabe mit möglicher Lösung
3. **Verhaltensanweisungen:** Detaillierte Anweisungen für den Gesprächsablauf, u.a.:
 - Keine direkten Lösungen verraten
 - Schritte nicht explizit benennen
 - Tipps nur auf Nachfrage geben
 - Auf Fehler hinweisen
 - Begründungen für Annahmen einfordern
 - Unrealistische Annahmen hinterfragen lassen
 - Fehler nicht sofort auflösen
 - Immer nur eine Nachfrage auf einmal stellen

Dieses Beispiel zeigt, wie detaillierte Anweisungen einen Chatbot zu einem effektiven pädagogischen Werkzeug machen können, das nicht einfach nur Antworten liefert, sondern den Lernprozess unterstützt.

6.4.2 Fallbeispiel: CALCBLUEBOT

Ein weiteres Beispiel ist CALCBLUEBOT von Prof. Robert Ghrist (University of Pennsylvania), ein speziell für die Unterstützung im Bereich der multivariablen Differentialrechnung konzipierter Chatbot:

- Der Prompt umfasst 427 Zeilen (etwa 28.000 Zeichen / 7.000 Token)
- Er enthält detaillierte fachliche Informationen und pädagogische Anweisungen

- Der Chatbot ist öffentlich auf GitHub³ verfügbar und kann als Vorlage für ähnliche Projekte dienen

Tipp

Bei der Entwicklung fachspezifischer Chatbots ist es oft hilfreich, mit einem umfangreichen Basis-Prompt zu beginnen und diesen durch Nutzungstests iterativ zu verfeinern. Die Rückmeldungen von Studierenden können wertvolle Hinweise für die Optimierung geben.

³<https://github.com/robertghrist/calculus>

7 Fazit und Ausblick

Die Integration von KI-Technologien in die Hochschullehre bietet vielfältige Möglichkeiten, steht aber auch vor zahlreichen Herausforderungen.

Eine zentrale Botschaft dieses Handouts ist, dass KI als Werkzeug zur Unterstützung von Lehrenden und Lernenden verstanden werden sollte, nicht als Ersatz für menschliche Interaktion und pädagogische Expertise.

KI-Systeme in der Bildung sind am effektivsten, wenn sie:

- Routineaufgaben automatisieren und so Zeit für wertvolle pädagogische Interaktionen schaffen
- Individualisierte Unterstützung ermöglichen, ohne den menschlichen Kontakt zu ersetzen
- Kreativität und kritisches Denken fördern, anstatt passive Konsumption zu begünstigen
- Als Reflexionswerkzeug dienen, das zum tieferen Verständnis anregt

Die größte Stärke von KI-Systemen in der Lehre liegt nicht in der Bereitstellung von Faktenwissen, sondern in der Unterstützung von Lernprozessen, dem Generieren von Ideen und der Förderung von Reflexion.

7.1 Ausblick auf zukünftige Entwicklungen

Die KI-Landschaft entwickelt sich mit atemberaubender Geschwindigkeit weiter. Einige Trends, die für die Hochschullehre besonders relevant sein dürften, sind:

- **Multimodale Systeme:** Zukünftige KI-Systeme werden zunehmend verschiedene Modalitäten (Text, Bild, Audio, Video) nahtlos integrieren, was neue Möglichkeiten für multimediales Lernen eröffnet.
- **Adaptive Lernsysteme:** KI-gestützte Systeme werden immer besser darin, sich an die individuellen Bedürfnisse, Lernstile und Fortschritte von Studierenden anzupassen.
- **Verbessertes Reasoning:** Die Fähigkeit von KI-Systemen zu logischem Denken und komplexer Problemlösung wird sich weiter verbessern, was besonders für MINT-Fächer relevant ist.
- **Zugänglichere Tools:** Die Entwicklung benutzerfreundlicher Schnittstellen wird die Technologie einem breiteren Kreis von Lehrenden zugänglich machen.
- **Institutionelle Integration:** Hochschulen werden zunehmend eigene KI-Infrastrukturen entwickeln und in ihre bestehenden Lernmanagementsysteme integrieren.
- **Ethik und Governance:** Die Entwicklung von Standards, Richtlinien und Best Practices für den verantwortungsvollen Einsatz von KI in der Bildung wird an Bedeutung gewinnen.

Tipp

Um mit den schnellen Entwicklungen im KI-Bereich Schritt zu halten, empfiehlt es sich, regelmäßig einschlägige Fachportale, Hochschuldidaktik-Plattformen oder spezialisierte Communitys zu verfolgen. Auch der Austausch mit Kolleginnen und Kollegen über Erfahrungen und Best Practices kann wertvolle Erkenntnisse liefern.

7.2 Handlungsempfehlungen für Lehrende

Abschließend möchten wir einige konkrete Handlungsempfehlungen für Lehrende geben, die KI-Technologien in ihrer Lehre einsetzen oder dies planen:

1. **Experimentieren Sie selbst:** Sammeln Sie eigene Erfahrungen mit verschiedenen KI-Tools, um ihre Möglichkeiten und Grenzen kennenzulernen.
2. **Beginnen Sie mit kleinen Schritten:** Implementieren Sie KI-Unterstützung zunächst in gut abgegrenzten Bereichen, bevor Sie umfassendere Anwendungen entwickeln.
3. **Fördern Sie KI-Kompetenz:** Integrieren Sie die Vermittlung von KI-Grundkenntnissen und kritischem Umgang mit KI-generierten Inhalten in Ihre Lehre.
4. **Evaluieren Sie kontinuierlich:** Sammeln Sie systematisch Feedback von Studierenden und reflektieren Sie kritisch über die Wirksamkeit Ihrer KI-unterstützten Lehrkonzepte.
5. **Bleiben Sie informiert:** Verfolgen Sie aktuelle Entwicklungen im Bereich KI für die Bildung und passen Sie Ihre Strategien entsprechend an.
6. **Teilen Sie Ihre Erfahrungen:** Tragen Sie zum kollegialen Austausch und zur kollektiven Wissensentwicklung bei, indem Sie Ihre Erkenntnisse mit Kolleginnen und Kollegen teilen.
7. **Achten Sie auf ethische Aspekte:** Reflektieren Sie kritisch über mögliche ethische Implikationen Ihrer KI-Nutzung und stellen Sie sicher, dass Sie Datenschutz und Fairness berücksichtigen.

Die Integration von KI in die Hochschullehre steht noch am Anfang einer spannenden Entwicklung. Mit einem reflektierten, kreativen und ethisch verantwortungsvollen Ansatz können Lehrende diese Technologien nutzen, um Bildungserfahrungen zu bereichern, ohne dabei die Kernwerte guter Lehre aus den Augen zu verlieren. Die Zukunft der KI in der Bildung wird nicht durch die Technologie allein, sondern durch die pädagogischen Visionen und Entscheidungen der Lehrenden gestaltet, die sie einsetzen.