

Tipps zur Nutzung des KI-Chatbots LobeChat

Wie funktioniert LobeChat?

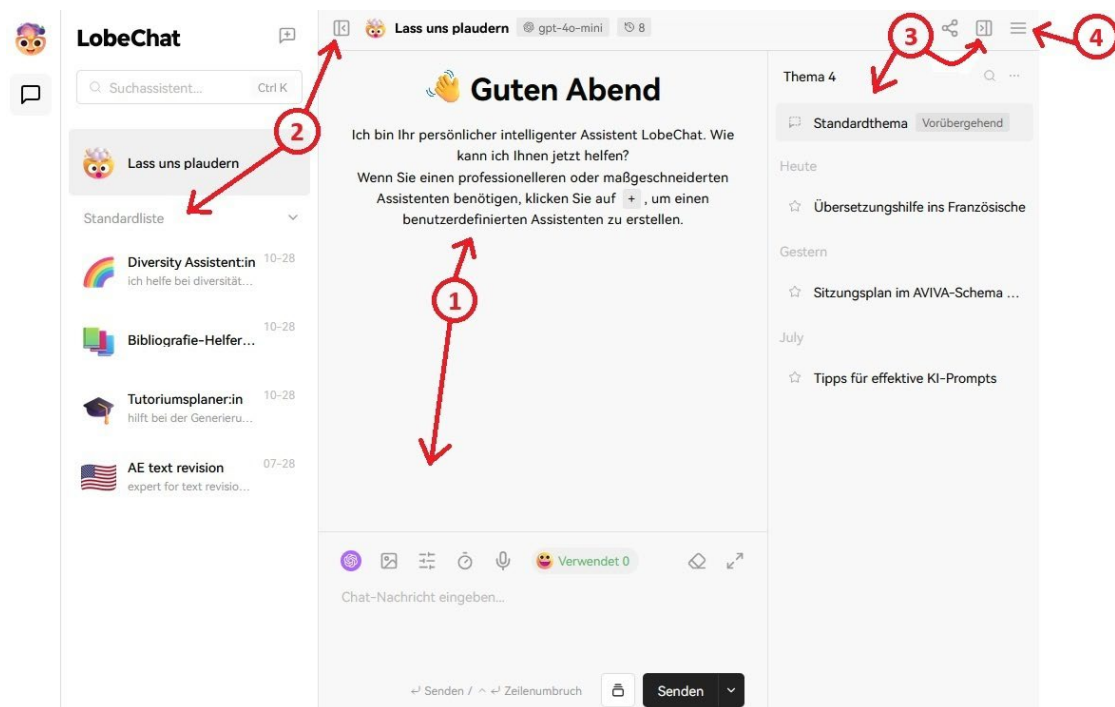
Gespräche in LobeChat finden über einen **Assistenten** statt.

Ein neuer Chat verwendet zunächst den **Standardassistenten**, siehe ①, der für grundlegende Anfragen ausreichend ist. Für komplexere Anfragen empfiehlt es sich, den Assistenten entsprechend zu konfigurieren.

Die **Liste der bereits verwendeten Assistenten** befindet sich auf der linken Seite, siehe ②.

Wenn man einen dieser Assistenten anwählt, kann man rechts das **Archiv der mit diesem Assistenten geführten Chats** einsehen, siehe ③.

Oben rechts öffnet man die **Einstellungen**, siehe ④. **ACHTUNG:** Solange man noch keinen Chat begonnen hat, öffnet dieses Symbol die **allgemeinen Einstellungen**. Sobald man sich in einem aktiven Chat befindet, öffnet das Symbol die **Sitzungseinstellungen** dieses Chats.



In den **Sitzungseinstellungen** gibt es u. A. folgende Optionen:

- In den **Assistenteninformationen** können Name, Beschreibung und Erscheinung des Assistenten geändert werden.
- In der **Rollenkonfiguration** kann ein Rollen-Prompt eingegeben werden, um das Verhalten des Assistenten zu definieren.
- In der **Chat-Präferenz** lässt sich u. A. definieren, wieviel der Chat-Historie bei den Antworten in Betracht gezogen wird. Je höher dieser Wert ist, desto mehr Token werden verbraucht und die Antworten werden langsamer generiert, aber sie werden auch besser auf den bisherigen

Materialien zum Workshop "Nachhaltiger KI-Einsatz in der Tutorienarbeit" (N. Veith/S. Berns)

Chatverlauf abgestimmt. Im Zweifelsfall sollten hier zunächst die Standardwerte beibehalten werden.

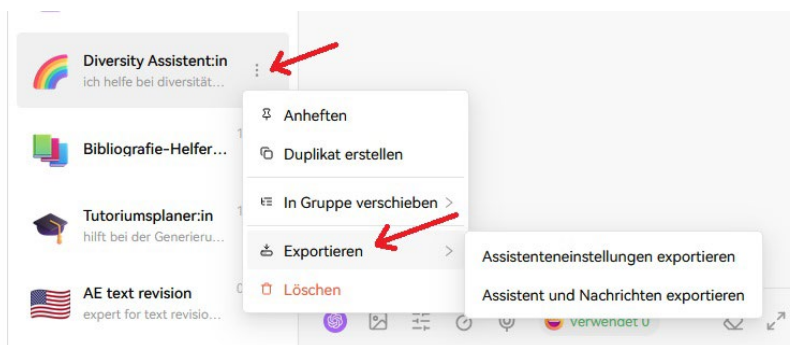
- In den **Modell-Einstellungen** kann das Modell gewählt und konfiguriert werden, indem man Änderungen an der Komplexität der Sprachverwendung und Argumentation vornimmt.
ACHTUNG: Insbesondere bei zu hoher Einstellung des Kreativitätsgrads sind einige Modelle überfordert und liefern keinen sinnvollen Output mehr. Bei den meisten Modellen empfiehlt sich ein Kreativitätsgrad von maximal 1,2–1,5. Im Zweifelsfall sollten auch hier zunächst die Standardwerte beibehalten werden.

Export und Import von Assistenten

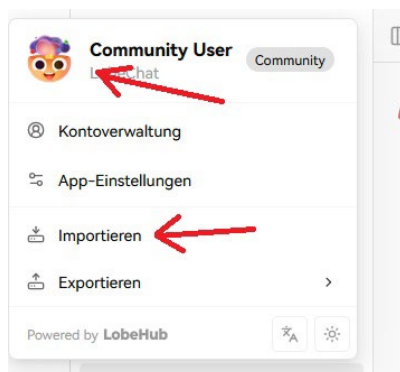
Einen gut funktionierenden Assistenten kann man bei LobeChat einfach exportieren, um ihn etwa mit Kolleg*innen und Komilliton*innen zu teilen.

Zum **Export** eines Assistenten gibt es zwei Möglichkeiten:

- Entweder öffnet man im Chat die **Sitzungseinstellungen** (④) und klickt dort im Menü links unten auf die Schaltfläche „Exportieren“.
- Oder man klickt in der Assistentenliste (②) auf das entsprechende Icon mit den drei Punkten neben dem Assistenten und wählt im sich öffnenden Menü „Exportieren“.



Daraufhin wird eine **json-Datei** generiert, die sich speichern und teilen lässt.

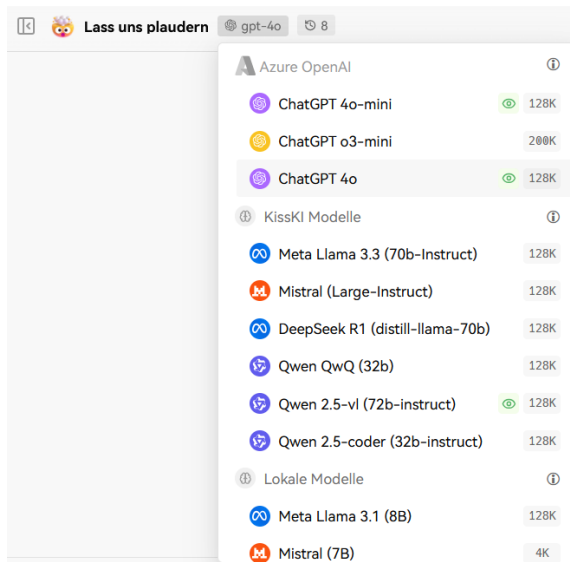


Zum **Import** eines Assistenten klickt man das Smiley-Symbol ganz oben links und wählt im sich öffnenden Menü die Option „Importieren“ und lädt die entsprechende json-Datei hoch.

Welches Modell passt zu welcher Anfrage?

Oberhalb des Chatfensters kann man das **Modell** auswählen (z.B. ChatGPT 4o, Meta Llama 3.3, Mistral, etc.). Die Modelle eignen sich unterschiedlich gut für unterschiedliche Anfragen – wenn man mit einem Modell nicht das gewünschte Resultat erhält, empfiehlt sich ggf. ein anderes Modell.

Grundsätzlich sind für komplexere Aufgaben **große Modelle wie ChatGPT 4o** zu empfehlen. „Mini“-Versionen eignen sich für leichte und schnelle Aufgaben. Entwicklerorientierte Modelle wie „Qwen 2.5-coder“ und „Meta Code Llama“ können speziell das Programmieren unterstützen und multimodale Modelle (z.B. Qwen 2.5-vl) verlangen Inputs mit klaren Bild-Text-Instruktionen.



Azure OpenAI bietet einen Zugang zu den kommerziellen Modellen von OpenAI wie ChatGPT. Sie gelten als leistungsstarke Allrounder mit Hosting in der EU Data Zone. Die Kosten der Anwendung werden vom Budget der Goethe-Universität gedeckt. Die Anfragen an die LLMs werden jedoch anonymisiert an externe Anbieter zur Verarbeitung weitergegeben.

Die **KissKI-Modelle** sind aus dem gleichnamigen Projekt der GWDG entstanden und eignen sich als kostenlose, in Deutschland gehostete Alternativen zu kommerziellen Anbietern, die zudem Fair-Use-Bedingungen garantieren.

Auf einem Server der Goethe-Universität sind **lokale Modelle** gehostet. Auch diese sind kostenlos und richten sich nach den deutschen Datenschutzbestimmungen und Sicherheitsrichtlinien.

Azure OpenAI:

| Modell | Anwendungszweck |
|-----------------|---|
| ChatGPT 4o-mini | <input checked="" type="checkbox"/> Leichte Aufgaben, schnelle Antworten, einfache Texte ⚡ Limitiert in komplexer Logik |
| ChatGPT o3-mini | <input checked="" type="checkbox"/> Einfache Recherche, weniger komplexe Aufgaben ⚡ Weniger geeignet für tiefergehende Textanalysen und -generierungen |
| ChatGPT 4o | <input checked="" type="checkbox"/> Komplexe Analysen, kreative Texte, präzise Antworten ⚡ Langsamer als Mini-Modelle |

KissKI-Modelle:

| Modell | Anwendungszweck |
|----------------|--|
| Meta Llama 3.3 | <input checked="" type="checkbox"/> Allgemeine KI-Anwendungen, textbasierte Aufgaben ⚡ Möglicherweise weniger präzise als ChatGPT-Modelle |
| Mistral | <input checked="" type="checkbox"/> Technische Inhalte ⚡ Erfordert präzise Eingaben |
| DeepSeek R1 | <input checked="" type="checkbox"/> Datenanalyse, explorative Informationen |
| Qwen QwQ | <input checked="" type="checkbox"/> Generelle Anwendungen, informelle Gespräche ⚡ Nicht für komplexe Aufgaben geeignet |
| Qwen 2.5-vl | <input checked="" type="checkbox"/> Aufgaben mit visuellen Inhalten ⚡ Benötigt klare Input-Instruktionen |
| Qwen 2.5-coder | <input checked="" type="checkbox"/> Programmierung, Codeanalyse, technische Projekte ⚡ Präzise technische Eingaben notwendig |

Lokale Modelle:

| Modell | Anwendungszweck |
|-----------------|---|
| Meta Llama 3.1 | <input checked="" type="checkbox"/> Basic-KI-Aufgaben, generelle Textnutzung |
| Mistral | <input checked="" type="checkbox"/> Technische Inhalte |
| Meta Code Llama | <input checked="" type="checkbox"/> Programmierung, Code-Generierung, technische Dokumentation ⚡ Fokus auf präzise Codeanforderungen |

Wie schreibt man einen Prompt?

Durch einen Prompt erteilt man dem KI-Chatbot eine Anweisung, wie er antworten soll. Je spezifischer der Prompt ist, desto höher ist die Qualität der Antworten. Folgende Elemente und Angaben sind dabei hilfreich:

- **Rolle & Kontext** – Weise dem Chatbot eine Rolle zu, z. B.: „Du bist ein Experte für Methodeneinsatz in der Hochschullehre,“ oder „Du berätst zum Studium mit körperlicher Behinderung.“
- **Aufgabe & Zweck** – Sag dem Chatbot genau, was zu tun ist, z. B. „Du sollst mir bei der Strukturierung meiner Notizen für die Klausurvorbereitung helfen.“
- **Zielgruppe** – Erkläre, mit wem der Chatbot kommuniziert und ggf. welche Bedürfnisse diese Person hat, z. B.: „Die Zielgruppe sind Tutor*innen an der Uni, die Erstsemester in wissenschaftliche Arbeitstechniken einführen,“ oder „Der Assistent ist zur Unterstützung für Studierende mit Lernschwäche, die Hilfe bei der Organisation benötigen.“

Materialien zum Workshop "Nachhaltiger KI-Einsatz in der Tutorienarbeit" (N. Veith/S. Berns)

- **Regeln** – Lege fest, was beim Antworten alles beachtet werden soll, z. B. „Frag zu Beginn immer nach dem Vorwissen der Teilnehmenden,“ „Schlag keine Methoden vor, für die man ein Flip Chart braucht,“ oder „verwende nur Informationen aus untenstehender Liste.“
- **Format & Länge** – Definiere, wie die Antwort ausgegeben werden soll, z. B.: „Schreibe einen Prosatext von ca. 400 Wörtern,“ „Stell die Ergebnisse in einer chronologisch sortierten Tabelle dar“, oder „Gib mir eine schrittweise Anleitung, wie ich mein Ziel erreichen kann.“
- **Sprache & Stil** – Gib vor, wie die Antworten formuliert werden sollen, z. B. „Antworte so kurz und einfach wie möglich,“ oder „Erklär es in freundlichem, professionellem Ton.“
- **Kontextinformationen** – Stelle relevante Informationen zur Verfügung, z.B. bestimmte Modelle, Theorien, Definitionen etc.
- **Beispiel** – Füge einige Beispiele ein („Few-Shot-Beispiele“), wie eine gute Antwort aussehen könnte.

Zusätzlicher Hinweis:

Komplexe Prompts funktionieren oft besser, wenn man sie in der Auszeichnungssprache **Markdown** strukturiert (es ist aber nicht verbindlich). Hier eine kurze Übersicht einiger gängiger Markdown-Syntax-Elemente:

Durch **Hashtags** kann man eine Überschriftenhierarchie einfügen:

Überschrift 1. Ordnung (H1)
 ## Überschrift 2. Ordnung (H2)
 ### Überschrift 3. Ordnung (H3)
 Etc.

Durch einen **Bindestrich** oder ein **Plus** kann man eine ungeordnete Liste mit Bullet Points einfügen:

- Erster Punkt
 - Zweiter Punkt
 Etc.

Durch Zahlen gefolgt von Punkten und Leerzeichen kann man eine nummerierte Liste einfügen:

1. Erster Punkt
 2. Zweiter Punkt
 Etc.

Wie verringert man Halluzinationen?

Halluzinationen treten bei allen Sprachmodellen auf, da es sich um statistische Modelle handelt, die die Sequenz von Wörtern basierend auf Wahrscheinlichkeiten in Abhängigkeit von der Eingabe (*prompt*) berechnen. So ergeben Studien, dass **fast die Hälfte der von Chatbots generierten Antworten unzuverlässig** sind. Dem gegenüber steht allerdings ein übergroßes Vertrauen der Nutzer*innen (*overreliance*) in die generative KI, welches die Aneignung und Weitergabe von Fehlinformationen begünstigt. Zudem sind Sprachmodelle meist **pre-trained**. Sie lernen demnach nicht durch User-Prompts hinzu, da ihr Training zum Zeitpunkt der Nutzung bereits abgeschlossen ist und einem veralteten Wissensstand entspricht. So steht auch GPT für *generative pre-trained transformer* (Oertner 2024, S. 261-266).

Materialien zum Workshop "Nachhaltiger KI-Einsatz in der Tutorienarbeit" (N. Veith/S. Berns)

Um Halluzinationen also zu verringern, empfehlen sich folgende Maßnahmen:

1. Nutze **fortschrittlichere Modelle**, damit die Trainingsdaten möglichst aktuell sind, z.B. ChatGPT-4 statt ChatGPT-3.
2. Formuliere **präzise Anweisungen** und nutze gegebenenfalls Teilfragen, um genauere Ergebnisse zu erhalten.
3. Füge möglichst viele vorab bekannte Informationen ein, z.B. „Ich weiß, dass [Fakt]. Kannst du dem weitere Argumente hinzufügen?“
4. Fordere **Quellen** an und überprüfe diese anschließend, z.B. „Nenne mindestens zwei seriöse Quellen“. **ACHTUNG**: Häufig sind Quellenangaben erfunden oder inhaltliche Verweise inkorrekt.
5. Definiere die Regel, dass die KI unmittelbar eine **Überprüfung der Ausgabe** durchführen solle, z.B. „Wenn Informationen fehlen oder unklar sind, nachfragen und auf keinen Fall etwas erfinden,“ oder „Wenn Fehler auftauchen, um Klärung und Korrektur bitten.“

AI-Bias erkennen und vorbeugen

KI ist erfahrungsgemäß nicht gut in der Berücksichtigung von Diversität. Dies liegt daran, dass die Trainingsdaten der Modelle häufig **unausgewogen** sind und **Fehler** oder **Vorurteile** enthalten. So gehörten zum Trainingskorpus der GPT-Modelle von OpenAI u. a. frei zugängliche Internet-Plattformen wie Reddit, auf denen viele nicht verifizierte Informationen sowie gelegentlich auch gezielte Desinformation oder Diskriminierung zirkulieren. Die an diesen Daten trainierten Modelle zeigen eine entsprechende Tendenz, Vorurteile zu reproduzieren, Informationen einseitig darzustellen, die Perspektiven von marginalisierten Personengruppen zu ignorieren oder den Eindruck eines falschen Konsenses zu erzeugen.

Als Endnutzer*in kann man zwar keinen Einfluss auf die Trainingsdaten der Modelle nehmen, jedoch ist es möglich, AI-Bias durch gezielte Prompt-Anweisungen vorzubeugen. Dies funktioniert am effektivsten in zwei Schritten:

1. Fordere den Assistenten von Anfang an auf, **Vorurteile gezielt zu vermeiden**, z. B. „Vermeide stereotype Darstellungen in Bezug auf Diskriminierungskategorien wie Geschlecht, Hautfarbe, Alter, Behinderung, etc.“, „argumentiere sachlich neutral ohne eine Perspektive zu bevorzugen,“ oder „Verwende in deiner Antwort diversitätssensible Sprache“.
2. Fordere den Assistenten nach dem Antworten auf, **seine eigene Aussage nochmal kritisch zu reflektieren**, auf eventuelle Vorurteile zu prüfen und ggf. Verbesserungen vorzuschlagen.

Literaturhinweise

M. Oertner, ChatGPT als Recherchetool? Fehlertypologie, technische Ursachenanalyse und hochschuldidaktische Implikationen, *Bibliotheksdienst* 58 (5), 259-297, 2024, entnommen aus: <https://doi.org/10.1515/bd-2024-0042> (zuletzt eingesehen am 16.06.2025)