Prompting
kurz & gut

LLMs verstehen, ChatGPT & Co.
professionell nutzen

DIE LESEPROBE

Attention please!

Die große Leistungsfähigkeit moderner LLMs geht vor allem auf die bereits erwähnten Attention-Mechanismen zurück – eine Neuerung, die um 2015 erstmals in der maschinellen Übersetzung eingesetzt wurde und 2017 mit dem sogenannten Transformer-Modell ins Zentrum der KI-Entwicklung rückte. Damals beschrieb eine bei Google angesiedelte Gruppe von KI-Forscherinnen und -Forschern in einem wegweisenden Paper mit dem Titel »Attention Is All You Need« (*arxiv.org/abs/1706.03762*) eine Architektur, die vollständig auf Attention basiert.

Mithilfe von Attention-Mechanismen können die Beziehungen zwischen Wörtern bzw. Tokens in einem Text erkannt und gewichtet werden. Dabei wird jedes Token mit allen anderen Tokens, einschließlich sich selbst, in Beziehung gesetzt – ein Vorgang, der als *Self-Attention* bezeichnet wird. So erkennt das Modell, welche Teile des Texts für das aktuell betrachtete Token besonders relevant sind. Die Fokussierung auf die Beziehungen zwischen Textelementen ähnelt in gewisser Weise der Art, wie Menschen Texte verarbeiten und sich Bedeutungszusammenhänge erschließen.

In LLMs, die auf der Transformer-Architektur basieren, werden meist mehrere parallel arbeitende Attention-Mechanismen (*Multi-Head-Attention*) eingesetzt, die die Zusammenhänge innerhalb eines Texts gleichzeitig unter unterschiedlichen Aspekten betrachten, beispielsweise aus grammatikalischer sowie aus semantischer Perspektive.

Die Ergebnisse der parallel berechneten Attention-Mechanismen werden miteinander kombiniert und verwendet, um eine neue, kontextabhängige Repräsentation des Tokens zu berechnen – basierend auf seinem ursprünglichen Embedding und den Beziehungen zu den anderen Tokens im Satz.

In Transformer-Modellen folgen mehrere solcher *Transformationen* der Token-Embeddings (daher auch der Name dieser Modelle) aufeinander, wobei in jedem Schritt alle Tokens des Inputs parallel verarbeitet und ihre jeweiligen kontextuellen Beziehungen untereinander untersucht werden.

Wie sehr sich die Bedeutung eines Worts durch seinen Kontext verändern kann, verdeutlicht folgendes Beispiel:

Satz	Bedeutung
Der Schlüssel passt nicht ins Schloss.	Schlüssel als Werkzeug zum Öffnen
Der Schlüssel zum Erfolg ist Ausdauer.	Schlüssel als Metapher

Die Bedeutung von »Schlüssel« hängt hier von den umgebenden Wörtern ab. Nach dem Durchlaufen der Attention-Schichten erhält das Wort in jedem Satz eine leicht unterschiedliche, kontextabhängige Repräsentation. In einigen Dimensionen werden die Gewichte unterschiedlich ausfallen: In einem Fall wird der physische Charakter als Objekt betont, im anderen die übertragene Bedeutung als Metapher.

Diese Fähigkeit zur kontextsensitiven Bedeutungsanpassung ist eine der zentralen Stärken moderner Sprachmodelle – sie erlaubt ihnen, Wörter nicht nur lexikalisch, sondern auch inhaltlich zu analysieren.

Generativ, pre-trained, Transformer: GPT

Die breite Öffentlichkeit kam erstmals durch ChatGPT mit Sprachmodellen auf Basis der Transformer-Architektur in Berührung – vor allem durch das dort zunächst verwendete Modell GPT-3.5. Die einzelnen Bestandteile der Abkürzung *GPT*, die bekanntlich für *Generative Pre-trained Transformer* steht, sind uns mittlerweile vertraut:

- *Generativ* heißt ein solches Modell, weil Sprache (oder anderer Output wie Bilder oder Videos) neu erzeugt wird.
- Es ist *pre-trained*, weil es auf umfangreichen Datensätzen »breit« trainiert wurde – als sogenanntes *Foundation Model* (Basismodell), das anschließend für spezifische Anwendungen weiter angepasst werden kann.
- Und es wird als *Transformer* bezeichnet, weil die initialen Token-Embeddings – wie zuvor beschrieben – mithilfe von Attention-Mechanismen in vielen Schritten transformiert werden, um kontextuell passende Repräsentationen des Texts zu erzeugen.

Auch moderne LLMs, die diese Abkürzung nicht in der Bezeichnung tragen (wie Claude, Gemini oder die neueren Modelle von OpenAI wie o1, bei denen dieses Akronym ebenfalls nicht mehr verwendet wird), sind heutzutage generativ, pre-trained und Transformer-Modelle.

Und wie kommt Sprache heraus?

Eine letzte wichtige Frage bleibt noch: Wie erzeugt ein LLM den Text, den wir als Antwort erhalten?

Jedes Mal, wenn Sie einen Prompt abschicken, wird der *gesamte Text und der Kontext eines Chats* (sofern alles ins Kontextfenster passt) mithilfe der Attention-Mechanismen und weiterer Verarbeitungsschritte analysiert. Die dabei berechneten kontextuellen Embeddings fließen in den nächsten Berechnungsschritt ein: Am Ende steht ein Vektor, der den »aktuellen Zustand« des Texts repräsentiert. Dieser Vektor wird mit allen Tokens im Wörterbuch abgeglichen – je ähnlicher das Token-Embedding diesem Gesamtvektor ist, desto wahrscheinlicher eignet es sich als Fortsetzung des Texts.

Aus den wahrscheinlichsten Kandidaten wird beim Sampling – einer gewichteten Zufallsauswahl – das nächste Token der zu generierenden Antwort ausgewählt (siehe hierzu auch »Playgrounds, Sampling und Hyperparameter« in Kapitel 3). Dass dabei nicht grundsätzlich das wahrscheinlichste Token ausgewählt wird, erlaubt dem LLM, »kreativ« zu sein und auf denselben Prompt leicht unterschiedliche Antworten zu geben.

Zur Fortsetzung der Antwort wird dieser Auswahlprozess nun für jedes weitere zu generierende Token wiederholt, wobei jedes Mal erneut der nun um ein Token verlängerte Gesamttext analysiert wird. Dieser Schritt-für-Schritt-Ablauf, bei dem immer ein Token nach dem anderen erzeugt wird, wird auch als *autoregressiv* bezeichnet.

Die Erzeugung einer Antwort könnte also wie in Tabelle 1.1 ablaufen. Bitte beachten Sie, dass bei Folgetokens gegebenenfalls das einleitende Leerzeichen mit zum Token gehört.

Tabelle 1.1: Die Antwort eines Sprachmodells wird Token für Token generiert.

Eingabe	Erzeugtes Token
Macht Prompting Spaß?	Das
Macht Prompting Spaß? Das	kommt
Macht Prompting Spaß? Das kommt	ganz
Macht Prompting Spaß? Das kommt ganz	darauf
Macht Prompting Spaß? Das kommt ganz darauf	an
Macht Prompting Spaß? Das kommt ganz darauf an	!
Macht Prompting Spaß? Das kommt ganz darauf an!	Viele
Macht Prompting Spaß? Das kommt ganz darauf an! Viele	Menschen
…	…

Dieser Ablauf wiederholt sich, bis entweder eine vorgegebene maximale Länge der Antwort erreicht ist oder das LLM ein spezielles Endtoken generiert. (Dass diese immer neue Betrachtung des gesamten Chatverlaufs bzw. Kontextfensters zum großen Energiehunger von KI-Anwendungen beiträgt, liegt auf der Hand.)

Bei der Generierung der Antwort und der Auswahl der Tokens greifen weitere trainierte Verhaltensweisen – etwa solche, die verhindern sollen, dass beleidigende oder gefährliche Inhalte erzeugt werden. Solche als *Alignment* bezeichneten »Einhegungen« eines LLM beschränken aufgrund ethischer und rechtlicher Überlegungen bewusst die per Prompt erzielbaren Ergebnisse.

LLM: eine zweite Definition

Unsere erste Definition eines Sprachmodells lautete:

> **Definition: Large Language Model – Version 1**
>
> Ein LLM ist ein künstliches neuronales Netzwerk, das mit großen Mengen an Textdaten trainiert wurde, um mithilfe statistischer Methoden Sprache zu verstehen und zu generieren. Es ist in der Lage, komplexe Zusammenhänge zu erfassen und Antworten in menschenähnlicher Sprache zu formulieren.

Auf der Basis der bisherigen Erörterung könnten wir diese Definition nun erweitern und verfeinern:

Definition: Large Language Model – Version 2

Ein LLM ist ein künstliches neuronales Netzwerk, basierend auf der Transformer-Architektur, das mit großen Mengen an Text vortrainiert ist.

Durch Attention-Mechanismen kann es kontextuelle Beziehungen zwischen allen Elementen eingegebener Textsequenzen erfassen und hat gelernt, sprachliche Muster sowohl auf grammatikalischer als auch auf der Bedeutungsebene zu erkennen.

Das Zusammenspiel dieser Fähigkeiten erlaubt dem Modell, menschenähnliche Antworten zu generieren – allein basierend auf den im Training erlernten Mustern und Kontextbeziehungen, ohne auf feste Textbausteine oder gespeicherte Inhalte zurückzugreifen.

Chatbots – die Anwendungsschicht

Als User nutzt man Sprachmodelle normalerweise innerhalb einer Anwendung – sei es im Browser, in einer Mobil- oder einer Desktop-App. Diese Anwendungsschicht ist das, was wir üblicherweise als Chatbot bezeichnen. In vielen dieser Apps lässt sich festlegen, welches Sprachmodell für eine Anfrage verwendet werden soll. Modell und Chatbot sind also nicht dasselbe.

Die Anwendungsschicht bestimmt auch, welche zusätzlichen Werkzeuge und Features beim Arbeiten mit Sprachmodellen zur Verfügung stehen: ob eine Websuche integriert ist, welche Dateien man als Kontext hochladen kann, ob on-the-fly Programme ausgeführt werden können, wie Spracheingaben verarbeitet oder Quellenangaben präsentiert werden. All das (und vieles mehr) sind Eigenschaften der App – nicht des eingesetzten Modells.

Das Zusammenspiel ist entscheidend

Im Vorwort hatte ich flott formuliert, dass der Dreiklang von Prompt, Kontext und Modell über die Qualität der Ergebnisse entscheidet, die uns ein LLM liefert. Ganz rund wird diese Aussage erst, wenn man auch den Chatbot selbst in die Betrachtung mit einbezieht.

Diese vier Komponenten kann man gezielt steuern, indem man sich fragt: Wie formuliere ich den Prompt? Was gebe ich an Kontext hinzu? Welches Modell wähle ich für eine bestimmte Aufgabe? Und welche Anwendung – also welcher Chatbot – soll es sein?

Redensartlich »hinken« die meisten Vergleiche – inklusive des Bilds des »hinkenden Vergleichs«. Trotzdem: Stellen Sie sich vor, Sie wohnten in einem großen Anwesen und verfügten über den Luxus eines Privatkochs samt Küchenpersonal. Sie kommen aus einem exotischen Urlaub nach Hause und haben ein brandneues Rezept mitgebracht.

Der Auftrag an Ihren Koch, Ihnen abends dieses Rezept zuzubereiten, wäre der Prompt – inklusive Angaben dazu, wann Sie essen möchten und wie viele Personen am Dinner teilnehmen werden.

Der explizite Kontext wäre in diesem Fall das Rezept selbst. Das Vorwissen von Koch und Küche, ob Sie es lieber spicy mögen oder Ihnen bereits die ersten Schweißtropfen auf der Stirn stehen, wenn jemand nur »rotes Curry« sagt, könnte man als impliziten Kontext betrachten – solchen Kontext, der heutzutage von Chatbots als interne »Erinnerungen« gespeichert werden.

Der Ablauf in der Küche – die hoffentlich gut »vortrainiert« ist – entspräche dem Modell. Versteht der Koch das Rezept? Weiß er, wie er es in Anweisungen an sein Team umsetzt? Kennen alle Mitarbeitenden die Zutaten und ihre Besonderheiten? Setzen alle die Anweisungen korrekt um? Haben sie gute Ideen, um das fertige Gericht ansprechend zu präsentieren?

Den Chatbot könnte man vielleicht mit den baulichen und technischen Voraussetzungen vergleichen: Ist die Klemmleiste praktisch,

an denen die Bestellungen hängen? Wie viele Kochstellen hat die Küche? Funktionieren die Küchengeräte – und welche gibt es überhaupt? Sind die Vorräte gut und schnell erreichbar?

Bei der Wahl des Modells und des Chatbots hilft nur, sich zu informieren und zu experimentieren. Neben der reinen Leistungsfähigkeit eines Modells und seiner Eignung für die jeweilige Aufgabenstellung spielen bei der Auswahl auch noch weitere Aspekte eine wichtige Rolle, darunter Kosten, Nutzungsbedingungen, Benutzeroberfläche, eventuelle Vorgaben des Arbeitgebers und nicht zuletzt persönliche Vorlieben und Vorkenntnisse.

Exkurs: KIs halluzinieren immer, nicht nur manchmal!

Jetzt haben Sie einen Überblick darüber gewonnen, wie ein LLM Sprache lernt und wie es selbst Sprache generiert. Wir müssen aber unbedingt noch einen weiteren wichtigen Begriff genauer unter die Lupe nehmen: den der allseits gefürchteten *KI-Halluzinationen*.

Landläufig wird darunter eine Fehlinformation verstanden, die eine KI erfunden hat. Vielleicht dichtet sie Ihnen in einem Gespräch den Besitz eines Haustiers an, das es gar nicht gibt. So berichtete der amerikanische KI-Forscher Gary Marcus, dass ChatGPT ihm fälschlich den Besitz eines Haushuhns namens Henrietta »unterjubeln« wollte.

Andere Halluzinationen sind ungleich schwerer zu erkennen, wenn etwa Quellenangaben oder Websites fantasiert werden, die es gar nicht gibt. ChatGPT trieb es dabei teils so bunt, dass es behauptete, bestimmte Webseiten besucht und die dortigen Inhalte abgerufen und in seiner Antwort berücksichtigt zu haben, obwohl dies ganz offensichtlich nicht der Fall war.

In gewisser Weise ist *jede* von einem Sprachmodell generierte Antwort – selbst wenn sie faktisch korrekt ist – eine Form der Halluzination, denn sie entsteht durch einen rein probabilistischen Prozess: In vielen Iterationen wählt das Modell jeweils das nächste Token basierend auf Wahrscheinlichkeiten aus.

Definition: Halluzination

Eine schöne, allgemeine Definition zu KI-Halluzinationen findet sich im Merriam-Webster, einem bekannten englischen (Online-) Wörterbuch:

A plausible but false or misleading response generated by an artificial intelligence algorithm.

Der Duden umschreibt es so:

[Eine] durch KI erzeugte, nicht auf Fakten oder realen Daten basierende falsche, jedoch glaubhaft erscheinende Information.

Dass man jede Antwort als Halluzination betrachten kann, ist eine bewusst provokante Behauptung und bedeutet nicht, dass Antworten von Sprachmodellen immer falsch sind. Diese Formulierung soll nur eindrücklich unterstreichen, dass Antworten – unabhängig davon, ob sie am Ende faktisch richtig oder falsch sind – immer auf die gleiche Art und Weise produziert werden: auf der Basis des gesamten Inputs einer Anfrage inklusive des gesamten vorhandenen Kontexts.

Im Kernmodell werden keine Fakten geprüft. Eine Faktenprüfung kann – sofern vorhanden – über externe Tools erfolgen, deren Ergebnisse lediglich als zusätzlicher Kontext in die Inferenz einfließen (teils vorab, teils während der Generierung der Antwort). Wenn Antworten schlussendlich korrekt sind, liegt das daran, dass die Kombination aus im Training gelernten Mustern und allen Kontextinformationen ausreicht, damit die erzeugte Antwort keine Fehler enthält.

SPIEGEL-Autor Ole Reißmann hat es einmal so auf den Punkt gebracht:[1]

Es sind Sprachmodelle, keine Faktenmodelle.

1 *https://www.spiegel.de/karriere/kuenstliche-intelligenz-wie-sie-mit-ki-anwendungen-ihre-kollegen-verblueffen-a-a992b64d-2b50-49d1-8151-d7b4e1204249*

KAPITEL 3

Fortgeschrittene Techniken und Tools

Nachdem Sie mit den Grundlagen vertraut sind, werfen wir nun einen Blick auf fortgeschrittene Techniken und Themen sowie unterstützende Tools rund ums Prompting, damit Sie Sprachmodelle noch gezielter und effektiver einsetzen können.

Zuerst beschäftigen wir uns mit typischen Features der führenden Chatbots, etwa mit Projekten, Erinnerungsfunktionen oder Deep Research, bevor wir uns dem wichtigen Thema der Zuweisung von Rollen und Personas widmen.

Ein weiterer Schwerpunkt liegt auf den Denkprozessen, dem *Reasoning* von LLMs. Daneben erfahren Sie, was Hyperparameter sind und wie Sie damit die Antworten von Sprachmodellen gezielt beeinflussen können. Im weiteren Verlauf dieses Kapitels schauen wir uns unterschiedliche Techniken und Tools an, die den Umgang mit Prompts erleichtern. Abschließend weiten wir den Blick – wir betrachten einige Grenzen von Sprachmodellen und experimentieren mit kreativen Strategien.

Typische Chatbot-Features

Wie gut und effektiv man mit Chatbots arbeiten kann, hängt nicht nur von den Fähigkeiten der verwendeten Modelle, sondern auch stark von der jeweiligen Benutzeroberfläche ab. Seit Einführung von ChatGPT hat sich die Benutzererfahrung nicht nur im Chatbot von OpenAI, sondern in allen »großen« Chatbots deutlich verbessert.

Um in langen, iterativen Chats bei der Arbeit an Content-Elementen – etwa einem Text oder einem Programm – nicht den Überblick zu verlieren, gibt es heutzutage UX-Elemente, die als *Canvas* (ChatGPT, Gemini) oder *Artifact* (Claude) bezeichnet werden.

Um in einer Vielzahl von gespeicherten Chats besser die Übersicht zu behalten, wurden *Projekte* eingeführt (ChatGPT, Claude). Hinzu kamen zusätzliche Werkzeuge wie die Websuche, Deep Research und die Analyse von visuellen Inhalten wie Bildern und Screenshots. Längst sind außerdem alle Chatbots als Mobil-Apps verfügbar, und es wurden zunehmend Optionen zur Sprachbedienung ergänzt.

Beim Prompten und der Bereitstellung von Kontext sollte man die Möglichkeiten der Chatbots mit berücksichtigen: Gerade bei der Sprachbedienung ist es hilfreich, die im jeweiligen Chatbot verwendeten Bezeichnungen der Features zu kennen, um gegebenenfalls direkt im Prompt eindeutig darauf verweisen zu können.

Individuelle Anpassung

Beginnen wir mit der Individualisierung von Chatbots am Beispiel von ChatGPT (hier in der Bezahlversion). Andere Chatbots weisen teils ähnliche Anpassungsmöglichkeiten auf.

Die *Einstellungen* – wir betrachten hier ChatGPT im Browser – erreichen Sie über das Menü, das sich hinter Ihrem Avatarbild verbirgt.

Neben vielen weiteren Einstellungen, die unterschiedliche Aspekte des Verhaltens von ChatGPT steuern, finden Sie in der Rubrik *Personalisierung* unter der Überschrift *Anpassung* den Eintrag *Individuelle Hinweise* (siehe Abbildung 3.1).

Wenn Sie diese Option anklicken, können Sie in einem weiteren Fenster *ChatGPT individuell konfigurieren* (siehe Abbildung 3.2). Zu diesem Fenster kommen Sie auch direkt über das Kontextmenü des Avatarbilds mit derselben Bezeichnung: *ChatGPT individuell konfigurieren*.

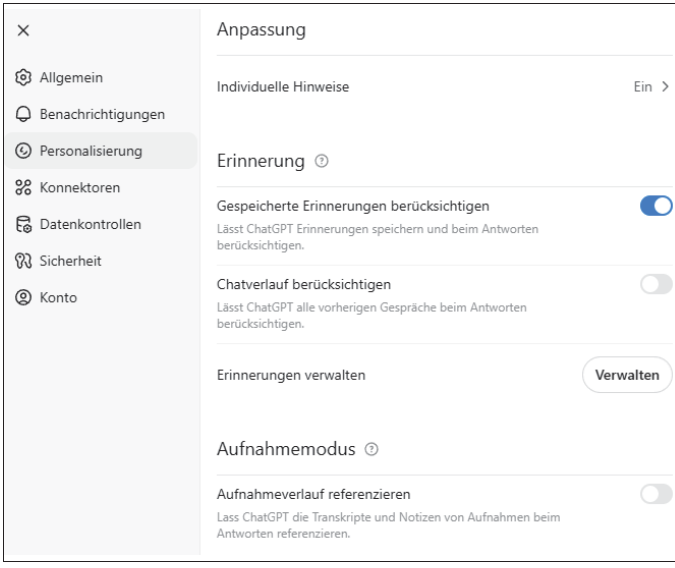

Abbildung 3.1: Personalisierung von ChatGPT

Schalten Sie jetzt zuerst – falls nicht automatisch aktiviert – ganz unten die Option *Für neue Chats aktivieren* ein.

Alle Angaben sind optional. Ob Sie gern Ihren Namen oder einen Spitznamen hinterlegen wollen, damit Sie persönlich angesprochen werden können, oder Ihren Beruf, ist sicher Geschmackssache.

Zeitgleich mit der Einführung von GPT-5 wurde erstmals eine Option angeboten, eine *Persönlichkeit* auszuwählen. Über das Info-Icon gelangen Sie auf eine Seite, auf der diese Persönlichkeiten etwas genauer beschrieben werden. Aus welchem Grund man allerdings mit einer zynischen ChatGPT-Version arbeiten sollte, erschließt sich mir nicht. Ich würde Ihnen empfehlen, bei der Standardpersönlichkeit zu bleiben und alle nötigen Anpassungen im nächsten Eingabefeld vorzunehmen.

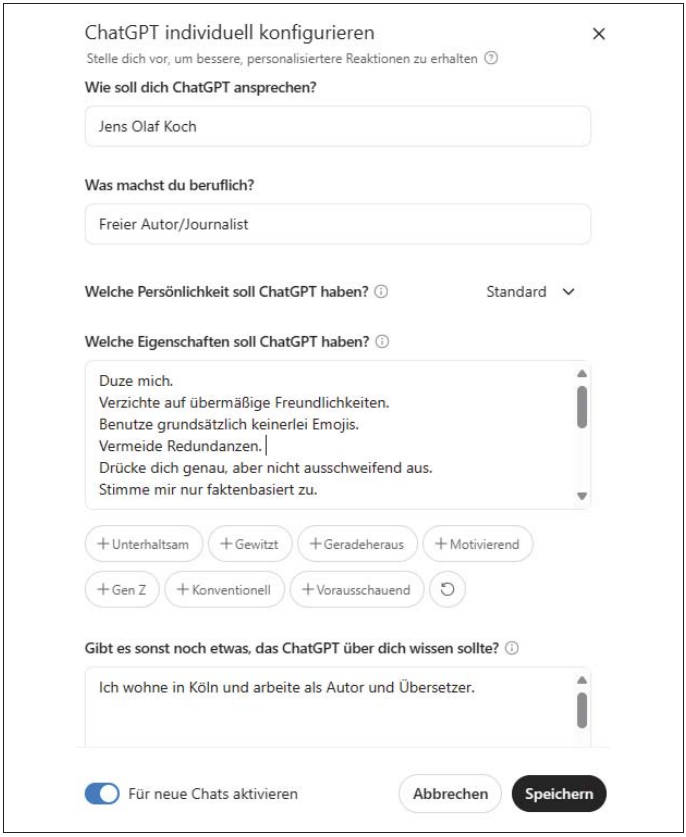

Abbildung 3.2: Individuelle Konfiguration

Spannender wird es im Eingabefeld mit der Überschrift *Welche Eigenschaften soll ChatGPT haben?*. Hier können Sie Angaben zum gewünschten Verhalten machen, etwa:

```
Duze mich.
Verzichte auf übermäßige Freundlichkeiten.
Benutze grundsätzlich keinerlei Emojis.
Verzichte auf Genderzeichen. ODER: Benutze Genderzeichen.
```

Vermeide Redundanzen.
Drücke dich genau, aber nicht ausschweifend aus.
Stimme mir nur faktenbasiert zu.
Stelle keine Fragen, die nur dazu dienen, den Chat zu verlängern.
Wenn du an von mir vorgegebenen Texten Änderungen vorschlägst, markiere diese fett, damit ich sie sofort erkennen kann.

Das ist nichts anderes als ein Prompt mit Vorgaben zum Verhalten des Chatbots, die für alle Konversationen gelten sollen. Bei Gemini finden Sie die analoge Funktion in den *Einstellungen* unter *Gespeicherte Informationen*, bei Claude direkt unter *Einstellungen*.

Vorgaben, die nur für einzelne oder Gruppen von Chats gelten sollen, gehören entweder in den ersten Prompt eines Chats oder in die Prompt-Vorgaben für Projekte (siehe hierzu auch »Projekte« oder »Eigene Chatbots: GPTs und Gems« in Kapitel 3).

Beachten Sie aber, dass die verschiedenen Sprachmodelle, die ein Chatbot nutzt und die in unregelmäßigen Abständen durch verbesserte Versionen ergänzt oder ersetzt werden, diese Instruktionen nicht immer vollkommen identisch auslegen werden. Selbst ein und dasselbe Modell wird sich mal enger, mal lockerer an die Vorgaben halten. Das liegt an den »statistischen Freiheiten«, die die Sprachmodelle bei der von Wahrscheinlichkeitsfunktionen gesteuerten Auswahl der generierten Tokens haben (Siehe hierzu vor allem »Playgrounds, Sampling und Hyperparameter« in Kapitel 3).

Unterschiedliche Sprachmodelle verhalten sich unterschiedlich. Man könnte fast von einem jeweils eigenen Charakter oder einer eigenen Persönlichkeit sprechen, auch wenn man damit in Gefahr gerät, LLMs zu sehr zu vermenschlichen. Das kann mitunter dazu führen, dass man nach Einführung eines Modells mit dessen Charakteristika unzufrieden ist und sich ein Vorgängermodell zurückwünscht. Falls dieses nicht mehr zur Verfügung steht, kann man versuchen, ein früheres Verhalten über geschicktes Prompten in den Personalisierungseinstellungen nachzubilden. Möglicherweise möchte man mehr Fließtext sehen und weniger Listen – oder »wärmere«, persönlichere Antworten und weniger »kalte« Erklärungen. Solch eine Anpassung ist weder einfach noch notwendigerweise von Erfolg gekrönt, kann das Leben aber deutlich erleichtern. Wie immer gilt: Experimentieren lohnt sich.

Im letzten Abschnitt der Personalisierungseinstellungen unter *Fortgeschritten/ChatGPT-Funktionen* können Sie unter anderem festlegen, ob die *Internetsuche*, die spontane Generierung und Ausführbarkeit von Python-*Code* und die Verwendung von Canvas-Bereichen erlaubt sein sollen (siehe Abbildung 3.3). (Siehe hierzu auch »KI-Arbeitsflächen: Canvas, Artifacts & Co.« weiter unten in Kapitel 3.)

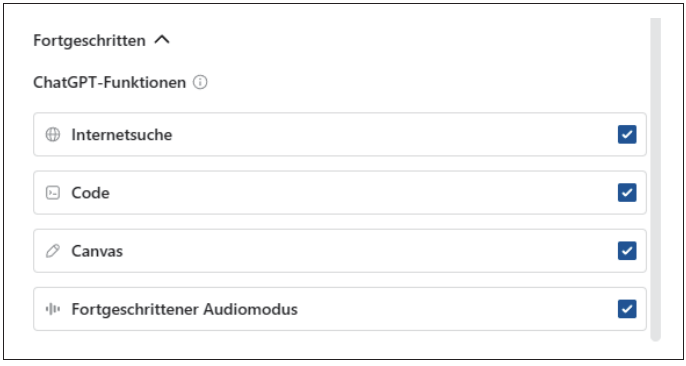

Abbildung 3.3: Welche Features soll ChatGPT einsetzen können?

Der *Fortgeschrittene Audiomodus*, der aktuell überwiegend zahlenden Abonnenten vorbehalten ist, basiert auf einem multimodalen Modell, das Sprache direkt »hört« und generiert – also ohne den Umweg über Text. Dadurch wirkt die Unterhaltung deutlich natürlicher, kann auf Ihre Sprechgeschwindigkeit reagieren und sogar emotionale Nuancen in der Stimme widerspiegeln. Im normalen Audiomodus dagegen wird Ihre Sprache zunächst in Text umgewandelt und erst dann von einem Sprachmodell beantwortet. Das wirkt weniger dynamisch und verzögert mitunter die Reaktion – ist für viele Anwendungsfälle aber völlig ausreichend.

Ich würde empfehlen, grundsätzlich alle Features zu aktivieren, solange es keine speziellen Gründe gibt, die dagegensprechen. Die Tendenz von ChatGPT, sich in einem Chat zu oft für die Darstellung eines Ergebnisses in Form eines Canvas zu entscheiden, kann mitunter nerven. Falls es Ihnen genauso geht, deaktivieren Sie einfach die entsprechende Option.

Erinnerungsfunktionen

Wäre es nicht schön, wenn sich die KIs, die wir nutzen, merken würden, welche Vorlieben und Kenntnisse wir haben? Welche Bücher und Filme wir mögen, welche Serien wir schauen, welche Programmiersprachen wir beherrschen, welche Anwendungen wir benutzen, welchen Beruf wir haben, welche Art der Kommunikation wir bevorzugen, wo wir zuletzt in Urlaub waren und wohin wir beim nächsten Mal reisen möchten, wie die Tante in Amerika heißt, ob wir Haustiere mögen (und wenn ja, welche) – und so weiter und so fort.

Um diesem Ziel näher zu kommen, bauen KI-Anbieter zunehmend Erinnerungsfunktionen in ihre Anwendungen ein. In ChatGPT finden Sie die entsprechende Funktion ebenfalls in den *Einstellungen* unter *Personalisierung* (siehe Abbildung 3.1). Eine ähnliche Funktion gibt es unter anderem auch in Gemini Advanced.

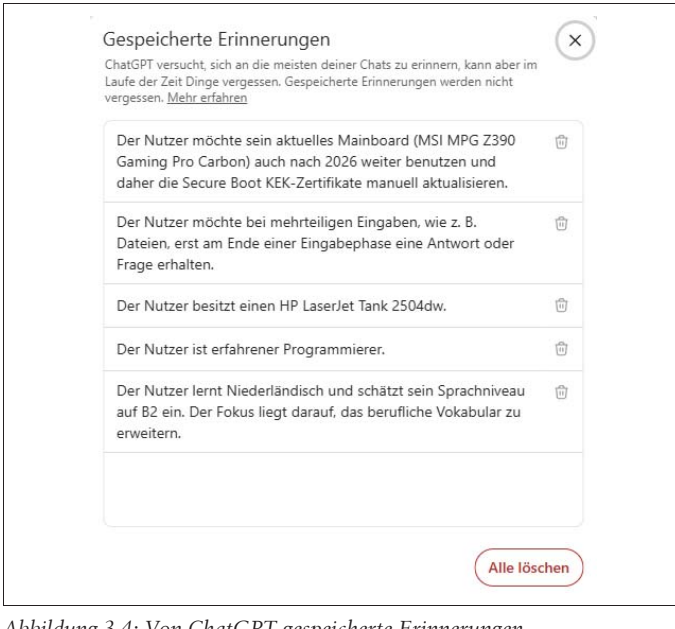

Abbildung 3.4: Von ChatGPT gespeicherte Erinnerungen

Zum Teil merken sich die Chatbots eigenständig solche Details aus Chats, die sie für wichtig halten, damit sie in späteren Unterhaltungen auf diesen zusätzlichen Kontext zurückgreifen können. Man kann Chatbots mit Erinnerungsfunktion aber auch selbst innerhalb einer Konversation dazu anhalten, sich etwas dauerhaft zu merken, indem man bestimmte Phrasen wie »Vergiss bitte nicht« oder »Merke dir, dass ...« benutzt. Welche Formulierungen effektiv dazu führen, dass Erinnerungen gespeichert werden, unterscheidet sich von Chatbot zu Chatbot – hier ist also etwas Experimentieren gefragt.

Die einzelnen Einträge, die als Erinnerungen angelegt werden, etwa:

```
Der Nutzer kocht selbst Marmelade.
Der Nutzer besitzt einen HP LaserJet Tank 2504dw.
```

stehen dem Modell als zusätzliche Angabe zu Verfügung, die meist zu Beginn eines Chats im Hintergrund abgerufen bzw. dem ersten internen Prompt hinzugefügt werden.

An der – von mir umgehend wieder gelöschten – Erinnerung zur selbst gekochten Marmelade lässt sich gut erkennen, dass die Verwendung der Erinnerungsfunktion auch Nachteile hat. Sprachmodelle können hier im Prinzip zwei Fehler machen: Sie können Informationen speichern, die eigentlich nicht relevant sind (nicht ich koche die Marmelade, meine 87-jährige Mutter kocht ... und nutzt kein ChatGPT).

Und sie können bereits gespeicherte Erinnerungen überbewerten: Nur weil man einmal nach einem indischen Restaurant gesucht hat, möchte man in einem anderen Chat, in dem man nach leckeren Reisgerichten fragt, sicherlich nicht ausschließlich indische Rezepte vorgeschlagen bekommen. Ein überspitztes Beispiel, das aber das Grundproblem verdeutlicht.

Neben den gespeicherten Erinnerungen können Sie bei ChatGPT festlegen, dass auch frühere Chats als Kontext herangezogen werden sollen (siehe Abbildung 3.1). Die Bezeichnung *Chatverlauf* bezieht sich dabei – anders als üblicherweise in diesem Buch – nicht auf den Verlauf eines einzelnen Chats, sondern auf die gesamte Historie aller gespeicherten Chats. Auch das kann zu einem *Memory Bias* (einer Art Erinnerungsdominanz) führen, bei der sich ein Modell zu sehr auf frühere Informationen fokussiert.

Rollen, Personas und benutzerdefinierte Chatbots

Zu Beginn eines Chats sendet ein Chatbot nicht nur den ersten Benutzerprompt an das ausgewählte Sprachmodell, sondern auch einen internen Prompt, oft als *Systemprompt* bezeichnet. Das geschieht für Benutzer unsichtbar im Hintergrund. Dieser vom Anbieter formulierte Systemprompt wird dabei jedem neuen Chat intern automatisch vorangestellt. (Wie ein solcher Systemprompt aussieht, schauen wir uns in »Interna zur Arbeitsweise der KIs erfragen« in Kapitel 3 beispielhaft an.)

Systemprompts spielen bei LLMs eine zentrale Rolle: Sie beeinflussen das grundlegende Verhalten des Modells und dessen Reaktionen auf Nutzereingaben.

Aber auch als Benutzer kann man ähnliche »Grundanweisungen« hinterlegen, wobei sich die entsprechenden Möglichkeiten von Anbieter zu Anbieter etwas unterscheiden. Interne, vom Anbieter definierte Systemprompts werden zwar vorrangig beachtet, dennoch bieten benutzerdefinierte Einstellungen viele Möglichkeiten. Einige davon haben Sie bereits im Abschnitt »Individuelle Anpassung« kennengelernt.

Besonders nützlich und beliebt ist es, einem Sprachmodell per Prompt bestimmte *Rollen* zuzuweisen. Das beginnt, normalerweise im ersten Prompt eines Chats, mit einfachen Zuschreibungen, beispielsweise:

```
Du bist ein hilfreicher Assistent für ...
Erkläre mir als Kfz-Monteur, warum ich ...
Unterstütze mich als Lehrer (Mentor, Coach, Code Reviewer ...)
bei ...
Agiere und sprich mit mir in der Art des fiktionalen Charakters
Forrest Gump aus dem gleichnamigen Film.
```

Das lässt sich weiter zu nahezu beliebiger Komplexität steigern. Diese Rollenzuweisungen werden auch oft als *Personas* bezeichnet.

Eigene Chatbots: GPTs und Gems

Damit man solche Zuweisungen nicht immer wiederholen muss, bieten die verschiedenen Chatbots – teilweise jedoch nur in den Bezahlabos – die Möglichkeit, diese dauerhaft in einer benutzerdefinierten Version des Chatbots zu hinterlegen. Bei ChatGPT wird schlicht von *GPTs* gesprochen, bei Google Gemini heißen die angepassten Versionen *Gems* (»Juwelen« oder »Schmuckstücke«).

Benutzerdefinierte GPTs, die andere User öffentlich bereitgestellt haben, können hingegen auch mit einem kostenlosen Konto genutzt werden. Die Verwaltung erfolgt über den Eintrag *GPTs* in der Sidebar von ChatGPT oder direkt über *chat.openai.com/gpts*. Dort lassen sich alle verfügbaren GPTs durchsuchen, starten und gegebenenfalls auch selbst erstellen und verwalten. Die Gems von Gemini ermöglichen es ebenfalls, personalisierte KI-Chatbots für spezifische Aufgaben oder Rollen zu erstellen. Eigene Gems kann man nur mit Gemini Advanced, dem kostenpflichtigen Gemini-Abonnement, anlegen – die Nutzung öffentlicher Gems ist dagegen kostenfrei möglich. Die Verwaltung erfolgt über den sogenannten *Gem-Manager*.

In Anthropics Claude gab es bei Redaktionsschluss dieses Buchs keine direkt vergleichbare Möglichkeit – Sie können dort aber stattdessen Projekte nutzen, um Instruktionen und Kontext wiederzuverwenden (siehe auch den Abschnitt »Projekte« in Kapitel 3).

Wenn Sie Personas wiederverwenden möchten (etwa um unterschiedliche Chatbots zu testen), bietet es sich an, den Prompt lokal als Snippet zu speichern, etwa mit Espanso (siehe auch »Prompts modular zusammensetzen: Espanso« in Kapitel 3). Auch manche Browsererweiterungen, beispielsweise Superpower ChatGPT, können Personas verwalten (siehe »Browsererweiterung: Superpower ChatGPT« in Kapitel 3).

> **Tipp: Persona-Prompts sind sinnvoll bei sich wiederholenden Umständen**
>
> Persona-Prompts bieten sich vor allem für zwei Anwendungsfälle an: für *längerfristige Projekte* und für *immer wiederkehrende Arbeitsumstände*.

Cheatsheet: Fortgeschrittene Techniken und Tools

Die folgende Übersicht zeigt die wichtigsten fortgeschrittenen Techniken und Ansätze, die in Kapitel 3 dieses Buchs besprochen wurden.

Technik/Ansatz	Beschreibung
Ebenen der Problembearbeitung reflektieren	Gliedern Sie Aufgaben in hierarchische Ebenen wie Strategie, Konzeption, Strukturierung, Produktion und Optimierung. Dieser Ansatz hilft, komplexe Probleme in überschaubare Schritte zu unterteilen und zielgerichtet zu lösen.
Rollen und Personas zuweisen	Definieren Sie spezifische Rollen für die KI, um Antworten besser an die Aufgabenstellung oder die Zielgruppe anzupassen. Das kann von fiktiven Charakteren bis hin zu spezialisierten Expertenrollen reichen. Nutzen Sie benutzerdefinierte OpenAI-GPTs oder vergleichbare Anpassungen bei anderen Anbietern, um für wiederkehrende Aufgaben durch geschicktes benutzerdefiniertes Prompting spezialisierte »Varianten« eines Chatbots anzulegen.
Chain-of-Thought-Methoden einsetzen	Fordern Sie bei Non-Reasoning-Modellen das LLM auf, Denkschritte explizit darzulegen, entweder vor der Ausführung einer Aufgabe oder parallel dazu. Diese Methode verbessert die Nachvollziehbarkeit und minimiert logische Fehler. (Bei Reasoning-Modellen überflüssig.)
Reasoning bei Hybridmodellen anfordern	Aktivieren Sie – sofern verfügbar – den Thinking-Modus über die Benutzeroberfläche (oder gegebenenfalls API). Im Prompt selbst setzen Sie am besten Qualitätsziele: »Denke gründlich und prüfe Annahmen und führe einen Gegencheck durch.«
Meta-Prompting anwenden	Nutzen Sie LLMs, um Prompts für spezifische Aufgaben zu generieren oder zu optimieren. Das ist besonders nützlich, wenn Sie unsicher sind, wie ein effektiver Prompt formuliert werden könnte.
Reverse Prompt Engineering einsetzen	Lassen Sie ein LLM bestehende Lösungen oder einen vorhandenen iterativen Chatverlauf analysieren, um einen Prompt entwickeln zu lassen, der das gewünschte Ergebnis möglichst direkt und effizient liefert.

Technik/Ansatz	Beschreibung
Playgrounds und Hyperparameter nutzen	Experimentieren Sie mit Plattformen wie dem OpenAI Playground, um Hyperparameter wie Temperatur und Top-p anzupassen. Dadurch können Sie Antworten hinsichtlich Kreativität, Konsistenz und Länge feinjustieren.
Model Hubs erkunden	Nutzen Sie zentrale Schnittstellen wie Model Hubs, um Zugriff auf unterschiedliche Modelle zu erhalten und die richtige KI für Ihre Aufgabe auszuwählen.
Prompten per Spracheingabe	Setzen Sie (auch am PC) Spracheingabe ein, um Modelle direkter zu steuern, entweder über die integrierten Funktionen der Chatbots oder eigenständige Software zur Spracherkennung. Das eröffnet neue Möglichkeiten für Anwendungen, bei denen schriftliche Eingaben unpraktisch sind, und bietet, je nach persönlicher Tippgeschwindigkeit und Qualität der Transkription, erhebliche Geschwindigkeitsvorteile.
Prompt-Bibliotheken nutzen	Sichten Sie gegebenenfalls kuratierte Prompt-Sammlungen und adaptieren Sie dort gefundene Vorlagen für Ihre Zwecke.
Hilfreiche Funktionen in Chatbots nutzen	Setzen Sie bewusst zusätzliche Features wie Canvas/Artifacts, Projekte oder Deep Research ein, um komplexe Aufgaben strukturierter zu bearbeiten und zwischen verschiedenen Domänen und Vorhaben zu wechseln.
Tools gezielt einsetzen	Verwenden Sie spezialisierte Werkzeuge wie Agenten, Wolfram Alpha oder Prompt-Generatoren in den Anbieter-Playgrounds, um erweiterte Aufgaben zu lösen, Prompts zu optimieren oder datengetriebene Analysen durchzuführen.
Erinnerungsfunktionen aktivieren	Nutzen Sie die Erinnerungsfunktionen und Benutzereinstellungen von Chatbots, um persönliche Präferenzen oder wiederkehrende Informationen zu speichern und Kontexte zu verbessern.
Prompting-Templates erstellen und Prompts archivieren	Entwickeln Sie gegebenenfalls wiederverwendbare Prompt-Vorlagen, die sich leicht an verschiedene Aufgaben anpassen lassen, und archivieren Sie sie systematisch für den späteren Einsatz. Das spart Zeit bei der Bearbeitung ähnlicher Aufgaben und sichert die in die Prompts geflossene Arbeitsleistung.
Cross-Bot-Validierung durchführen	Prüfen Sie gegebenenfalls Ergebnisse eines Chatbots durch einen zweiten Bot, um alternative Perspektiven oder Fehler in den Antworten zu identifizieren und zu korrigieren.
Prompten für Bild-KIs	Nutzen Sie modellspezifische Parameter und gegebenenfalls künstlerische und fotografische Fachbegriffe (Stil, Komposition, Perspektive, Licht, Schärfe). Greifen Sie gegebenenfalls auf Meta-Prompting in Chatbots zurück, um Prompts zu optimieren.