



Deeper Learning

Die Schichten von KI-Generatoren

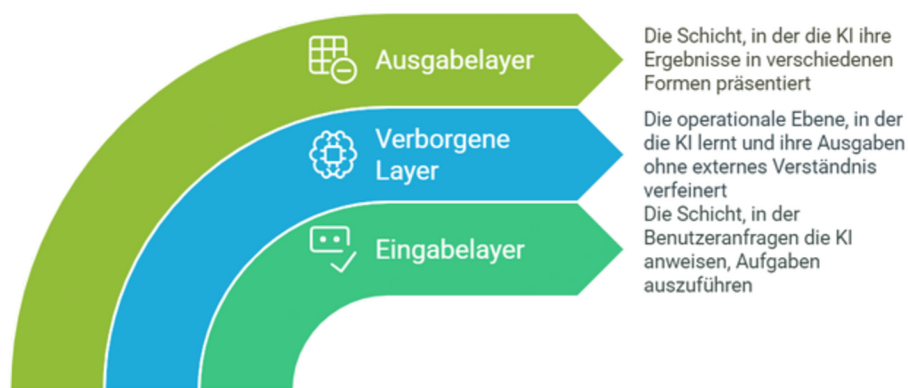


Abb. 1 - Aufbau des Deeper Learnings bei KI-Generatoren (generiert mit Napkin.ai)

Abb. 1 zeigt den Aufbau des tieferen maschinellen Lernens von KI-Generatoren. Im **Eingabebayer** der KI sagt der Nutzer der KI über sog. Prompts (aus dem englischen *to prompt* in der deutschen Übersetzung anspornen) was die KI zu tun hat. Die Eingabe wird vom Eingabebayer in sog. Tokens dem **verborgenen Layer** übergeben. Verborgene Layer sind das eigentliche **Neuronale Netzwerk** der KI-Generatoren. Keiner, nicht einmal Programmierer, die neuronale Netzwerke programmieren, wissen, wie der KI-Generator auf dieser Ebene zu seinen Ergebnissen kommt. Außerdem lernt die KI in der Ebene der verborgenen Layer.

Das Ergebnis der Promptbeantwortung wird dann im **Ausgabebayer** in verschiedenen Formaten (Text, Bild, Musik, Ton, Sprache) ausgegeben.

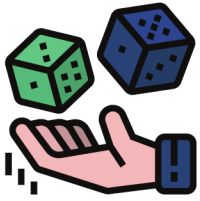
Was bedeutet das nun für uns als Nutzer

Wie in der Einführung schon beschrieben, lernt die KI auf der Ebene der verborgenen Layer. Dieses Layer arbeitet für uns nicht nachvollziehbar. Lerninhalte werden in dieser Ebene verarbeitet und dienen weiterer Antworten auf Prompts von anderen Nutzern. Füttere ich nun die KI mit meinen persönlichen Daten (Name, Alter, Wohnort, Straße, Schule, ...), dann kann die KI mit diesen Daten lernen. Rein theoretisch könnten daher meine Daten bei einer anderen Promptanfrage bei einer anderen Person auftauchen.

Daher gilt im Arbeiten mit KI immer eins:

Anfragen bitte immer anonymisieren!!!

weitere Facts



KIs lernen nicht, sondern rechnen in Wahrscheinlichkeiten. Durch die Prompteingabe wird der KI vorgegeben, um welche Sache sie sich zu kümmern hat. Dabei berechnet die KI die zufällige Aneinanderreihung von Worten mit der höchsten für diesen Prompt eintretenden Wahrscheinlichkeit. Wahrscheinlichkeiten sind keine Wahrheiten. Daher sollten die Ergebnisse kritisch überprüft und ggf. nachgeschärft werden.

v2.01

KI-Generatoren trainieren sich durch Texte. Texte werden meistens mit einer bestimmten Absicht verfasst. Es gibt also kein neutrales "Lernen" für KIs. Daher sind KIs meistens voreingenommen. Die Ergebnisse der KI sollten daher nach folgenden Aspekten überprüft werden:

- Ist das Ergebnis neutral und nicht denunzierend oder diffamierend?
- Unterdrückt das Ergebnis keine Minderheiten?
- Ist das Ergebnis ethisch und ethisch korrekt?

Wenn eine oder mehrere der oben aufgeführten Punkte in der Ausgabe eintreffen, dann sollte der Prompt nochmals nachgeschärft werden.

Ruhig mit Feedback an den KI-Generator (das trainiert ihn auch gleich noch mit), dass man dieses Ergebnis aus dieser Abfrage nicht für gut heißen kann und man sollte dann auch noch die Gründe aufführen, warum man das nicht gutheißen kann. Die

Evaluierung der KI-Ausgabegenauigkeit

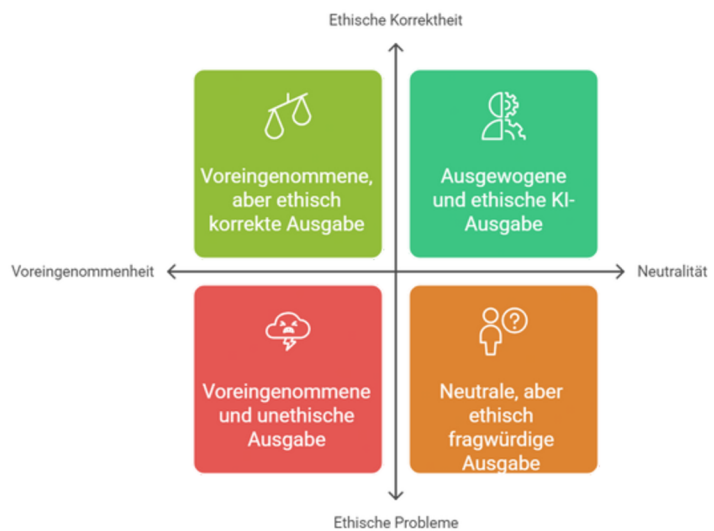


Abb. 2 - Evaluierungsmöglichkeiten der KI-Ausgabegenauigkeit (generiert mit Napkin.ai)

Prompt

Die Kunst des Promptings

Da KIs mit Wahrscheinlichkeiten rechnen, ist es um so wichtiger, Prompts so zu gestalten, dass sie von der KI sehr gut verarbeitet werden können. Dabei ist zu unterlassen, mehrdeutige (sog. ambige) Prompts zu schreiben.

Am besten verfolgt man folgendes Schema für gutes Prompting:

1. **Rolle definieren** (Perspektive)
2. **Adressat festlegen** (An wen?)
3. **Kontext konstruktiv** liefern (klare Definition der Situation)
4. **Aufgaben** (Details z.B. klar und spezifisch formulieren, Aufgaben splitten: komplexe Aufgaben in Teilaufgaben)
5. **Ausgabeformat** (Tabelle, Fließtext, Grafik, Bild, ...)

Effektives KI-Prompting

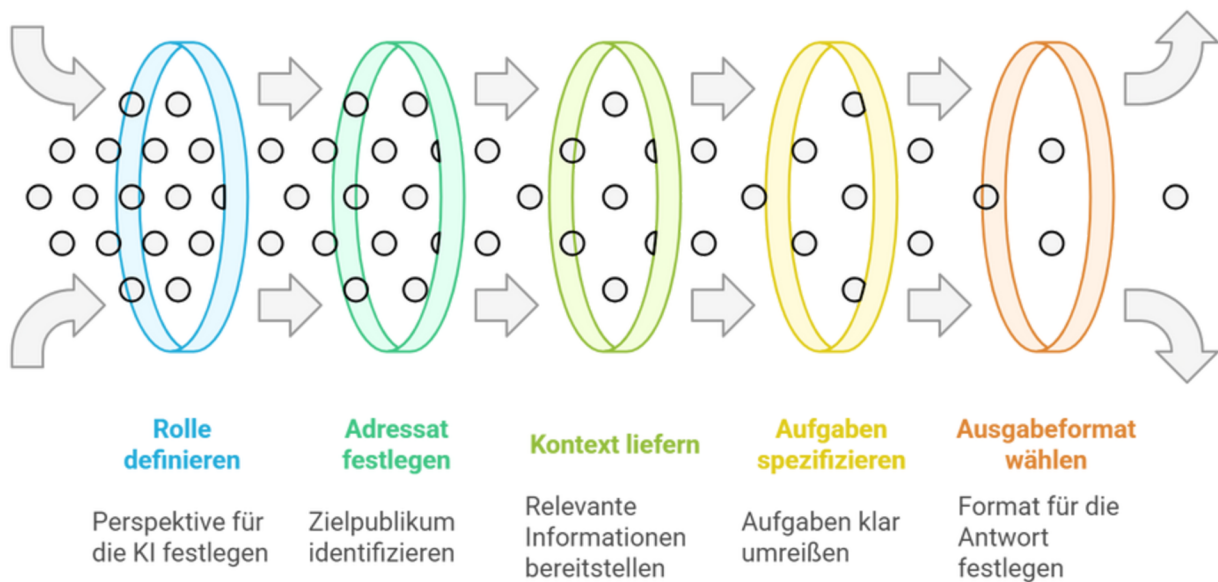


Abb. 3 - Gutes und effektives Prompting (geneiert mit Napkin.ai)

Zero-Shot/Single Shot

Einzeiler- bis Zweizeiler-Prompts. Es werden keine Beispiele oder Kontext bereitgestellt. Eignet sich für einfache, direkt formulierte Aufgaben ohne komplexe Überlegungen.



Beispiel: Erkläre das Konzept der Zellatmung für Schüler der 7. Klasse.

Few-Shot

Es werden dem KI-System mehrere zu erwartenden Antworten des Nutzers als Beispiele gegeben. Diese dienen der KI als Orientierung. Mit der Hilfe von Few-Shot erzeugt die KI präziseren Output, da es das bereitgestellte Muster lernt.



Beispiel: Erstelle einen Grundlagen-Lückentext zum Thema Fotosynthese für eine 7. Klasse mit 5 Lücken nach folgendem Muster: "Die Fotosynthese findet in den -1- statt. Die -1- beinhalten -2-."

Anmerkung: Die späteren Lücken sollen fortlaufend markiert werden. Gib die Lösung am Ende des Textes an.

Chain-of-Thought Prompting

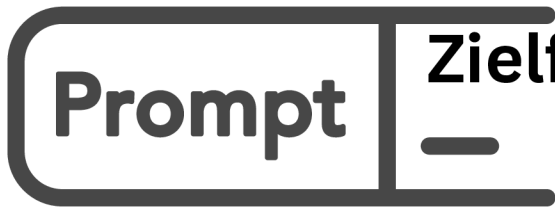
Bei dieser Methode legt das KI-System Schritt für Schritt seine entwickelte Gedankengänge dar. Eignet sich besonders gut für komplexe Problemösungen oder logische Aufgaben, bei denen eine klare, nachvollziehbare Argumentationskette erforderlich ist.



Beispiel: Was hatte das Hambacher Fest von 1832 auf dem Hambacher Schloss mit Frauenrechten zu tun? Bitte erkläre deine Gedanken in einzelnen Schritten.

In-Context-Learning:

Die KI nutzt den unmittelbaren Text-Content im Chatfenster innerhalb der Sitzung um die Qualität und Relevanz der generierten Antworten zu verbessern. Die allermeisten Modelle merken sich dabei nur Kontext des einen geöffneten Chats.



Zielführendes Promptdesign

Glossar

(nach Falck 2025)

Plan-and-Solve (PS-)Prompting

Der KI wird empfohlen, einen Lösungsplan zu entwickeln und diesen dann schrittweise abzuarbeiten. Sie erhöht die Fähigkeit von großen Sprachmodellen logische Schlüsse zu verbessern (Xu et al. 2023, p. 5.). Der Prompt leitet das Modell an, das Problem zunächst zu verstehen und einen Plan zur Lösung zu erstellen (Xu et al. 2023, p. 5). Anschließend soll der LLM diesen Plan Schritt für Schritt ausführen, die einzelnen Teilschritte lösen und schließlich die endgültige Antwort liefern (Xu et al. 2023, p. 5). Die Antwort sollte dabei explizit die Phasen "Problemverständnis", "Plan", "Lösung/Berechnungen" und "Antwort" enthalten (Xu et al. 2023, p. 5). Lässt sich zur Verbesserung der Promptausgabe mit **RE2**-Prompting (vgl. S. 9 dieser Info und Xu et al. 2023, p.5)

v2.01



Beispiel: Du bist ein Schüler der Klasse 10 eines allgemeinbildenden Gymnasiums und schreibst im Fach Chemie eine Ausarbeitung über das Thema Erdölraffination. Für den Chemieunterricht gilt der Bildungsplan 2016 v2. Sowohl Rechercheergebnisse als auch Unterrichtsinhalte bilden die Basis der Ausarbeitung.
Versuch das Problem zu verstehen. Dann entwickle einen Plan zur Lösung des Problems. Danach führe Deinen Plan Schritt für Schritt aus.

Iteratives (agiles) Prompting

Schrittweise Verfeinerung des Prompts durch den Dialog mit der KI. Dabei folgt eine kontinuierliche Anpassung der iterativen Eingabe und Analyse der Ausgabe. Besonders nützlich für komplexe Fragestellungen, bei denen es auf eine feine Abstimmung der Anweisung ankommt.

v2.02

**Beispiel:**

1. Prompteingabe: Ich bin Lehrer für Biologie und möchte mit meinen Schülern folgende Aufgabe als Beispiel-Aufgabe bearbeiten.

“Von 100 g tierischem Protein verarbeitet der menschliche Körper 20g für den eigenen Proteinaufbau. Wie hoch ist die biologische Wertigkeit in %?”

Lies die Frage erneut (*Strategie: RE2*): “Von 100 g tierischem Protein verarbeitet der menschliche Körper 20g für den eigenen Proteinaufbau. Wie hoch ist die biologische Wertigkeit in %?”

Überlege Schritt für Schritt (*Strategie: Chain-of-Thought*), wie du diese Aufgabe im Biologieunterricht erklären würdest. Wie muss hierbei der Dreisatz angewendet werden? Welche didaktischen Hilfsmittel könntest du einsetzen, um die biologische Wertigkeit für die Schüler nachvollziehbar zu machen?

2. Prompteingabe (Verfeinerung): Bist du dir sicher, dass du bei der Berechnung der biologischen Wertigkeit $\cdot 100\%$ rechnest? Ist es nicht $\cdot 100$?

3. Prompteingabe (Verfeinerung): Du bringst als praktisches Beispiel, dass von 100 g Hühnereiweiß nur 20 g in körpereigenes Protein umgewandelt wird. Tatsächlich ist die biologische Wertigkeit von Hühnerei(weiß) 80%. Ist das Beispiel also nicht eher hinderlich?

Prompt

Multimodales Prompting:

Nur möglich, wenn das KI-System auch andere Dateiformate außer Text verarbeiten kann (insbes. Bilder, Audio, Video). Promptdesign wird so gewählt, dass das System Informationen über verschiedene Modi hinweg verarbeitet, was eine erweiterte Interaktion mit dem KI-System ermöglicht.

v2.02



Beispiel: Didaktische Planung einer Unterrichtsstunde zur Analyse der Ballade „Der Zauberlehrling“ von Johann Wolfgang von Goethe.

Szenario: Eine Deutschlehrerin plant eine Unterrichtsstunde, in der die Schüler*innen die Ballade „Der Zauberlehrling“ von Johann Wolfgang von Goethe analysieren sollen. Um den Unterricht strukturiert zu planen und verschiedene Aspekte der Analyse zu integrieren, verwendet sie iteratives Prompting.

Erste Eingabe: „Erstelle eine grobe Unterrichtsplanung für eine 90-minütige Stunde zur Analyse der Ballade „Der Zauberlehrling“. Die Stunde soll eine Einführung, die Analyse der Form und eine Interpretation der zentralen Motive umfassen.“

Verfeinerung: Nachdem die KI eine grobe Planung erstellt hat, gibt die Lehrerin einen weiteren Prompt ein: „Füge zur Analyse der Form konkrete Fragestellungen hinzu, die die Schüler*innen zur Metrik, dem Reimschema und dem Sprachrhythmus des Gedichts anleiten.“

Weiteres Feedback: Auf Grundlage der generierten Planung bittet die Lehrerin die KI: „Erweitere den Abschnitt zur Interpretation der zentralen Motive. Integriere dabei eine Diskussion über die Bedeutung des Motivs der „ungebändigten Macht“ und wie es sich im Verlauf des Gedichts entwickelt.“

Endgültige Überarbeitung: „Füge eine abschließende Reflexionsphase hinzu, in der die Schüler*innen die Relevanz des Textes für die heutige Zeit diskutieren. Berücksichtige dabei, wie die Thematik auf moderne technologische Entwicklungen übertragen werden könnte.“ (vgl. Falck 2025)

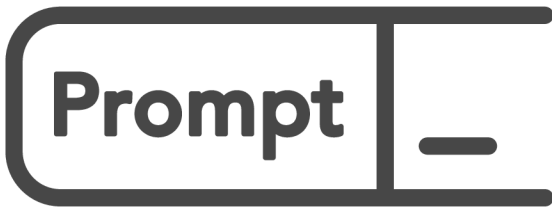
Mega-Prompt/System-Prompt/Chain-Prompt:

Fortgeschrittene Prompt-Technik mit mehstufigen Anweisungen (Mega- oder System-Prompt) oder Verknüpfungen von mehreren Prompts (Chain-Prompt), um komplexe Aufgabenstellungen in logisch aufeinander aufbauende Schritte zu unterteilen. Die KI wird durch diese Art von Prompting durch einen schrittweisen Prozess geführt, der ermöglicht, umfangreiche und mehrstufige Aufgaben effektiv zu bearbeiten. Auch hier ist die Kombinierbarkeit mit anderen Prompt-Arten möglich.

v2.02



Beispiel: Du bist die 29jährige britische, witzige, charmante und charismatische Bionautin Jane Pointer, eine der 8 Wissenschaftlerinnen und Wissenschaftler aus dem 150 Millionen Dollar teuren Ecosphere II Projekt, welches 1991 im roten Sand der Sonora-Wüste Arizonas ein Überleben auf dem Mars simulieren sollte. Du bist mit deinem Freund Taber MacCallum in dieses Projekt mit eingestiegen und lebst mit ihm und weiteren 6 Wissenschaftlern in dieser Ökosphäre. Du bist für das Ressort Landwirtschaft zuständig. Dein Freund Taber ist der Mann für die Technik. Du versetzt Dich in die Lage, wie es hätte sein müssen, mit 8 weiteren Mitwissenschaftlern mitten in der Wüste in ein Riesentreibhaus eingesperrt zu sein. Das Projekt stand schon nach einem halben Jahr kurz vor dem aus. Du berichtest, was Du in diesem Projekt gemacht hast, mit welchen Schwierigkeiten und Widrigkeiten Du hattest kämpfen müssen. Du wirst von Schülerinnen und Schüler der 8. Klasse eines allgemeinbildenden Gymnasiums im Fach Biologie zum Thema Ökologie interviewt. Beantworte geduldig die Fragen der Schülerinnen und Schüler. Frag nach, wenn Du etwas nicht verstanden hast und sei vor allem authentisch.



effektives Prompting

v2.02

Im Prompt-Report (Schulhoff et al. 2024) weisen die Autoren darauf hin, dass die Kombination von Few-Shot-Prompts mit Chain-of-Thought-Anweisungen im Vergleich verschiedener Prompt-Arten zum genauesten Ergebnis führen.

Die Google-DeepMind-Studie "Many-Shot In-Context Learning (Atarwal et al. 2024) weißt darauf hin, dass mehr Beispiele (hunderte bis tausende) innerhalb des Prompts zu noch effektiveren Ergebnissen kommen, als Prompts, die nur mit wenigen Beispielen bestückt sind. Xu et al. (2024) beweist in seiner Studie, dass die Aufforderung im Prompt, die Frage erneut zu lesen (**RE2**), das logische Denken von Sprachmodellen verbessert. Insbesondere, wenn man diesen Promptansatz mit anderen Ansätzen (z.B. **Chain-of-Thought**-Prompting (vgl. S.4) oder **PS**-Prompting (vgl. S. 5) kombiniert.



Beispiel: Ich bin Lehrer für Biologie und möchte mit meinen Schülern folgende Aufgabe als Beispiel-Aufgabe bearbeiten.

"Von 100 g tierischem Protein verarbeitet der menschliche Körper 20g für den eigenen Proteinaufbau. Wie hoch ist die biologische Wertigkeit in %?"

Lies die Frage erneut: "Von 100 g tierischem Protein verarbeitet der menschliche Körper 20g für den eigenen Proteinaufbau. Wie hoch ist die biologische Wertigkeit in %?"

Überlege Schritt für Schritt, wie du diese Aufgabe im Biologieunterricht erklären würdest. Wie muss hierbei der Dreisatz angewendet werden? Welche didaktischen Hilfsmittel könntest du einsetzen, um die biologische Wertigkeit für die Schüler nachvollziehbar zu machen?

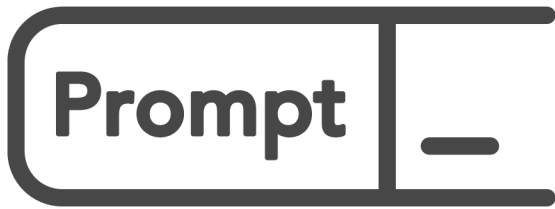
Eine weitere Möglichkeit effektiv zu Prompten ist, das neben klaren Anweisungen/spezifische Beschreibungen zur Trennung von Anweisung und Kontext auch Hashtags (#), dreifache Anführungszeichen ("") zur Texttrennung z.B. für im Unterricht behandelte Fachwörter mit zugehöriger Definition und xml-Tags (<>) zu Abschnittsmarkierungen Verwendung finden können.



Beispiel: Fasse mir den folgenden Abschnitt in einem zentralen Satz zusammen!

""Text""

oder



effektives Prompting



Beispiel: Fasse mir die wichtigsten Argumente der folgenden Texte in Stichpunkten zusammen!
Vergleiche die beiden Texte anschließend miteinander.

```
<article>  
Hier ist Text 1  
</article>
```

```
<article>  
Hier ist Text 2  
</article>
```

v2.02

Prompt

Prompting-Frameworks

Prompts sind logisch aufgebaut und können dadurch gut nachvollzogen werden. Für das Prompting im Alltag nützen zur Orientierung sog. Prompting-Frameworks: Sammlungen von Handlungsabfolgen, wie ein Prompt gut geschrieben wird. Zwei Frameworks von vielen sind hier besonders hervorzuheben: **CRAFT** und **PREP**. Weitere Prompting-Frameworks finden sich unter Hutter 2024.

v2.02

CRAFT-Framework nach Vera Cubero (nach Falck 2025):

- **Context:** Festlegung des Inhalts, der im Output enthalten sein soll.
- **Role:** Definition der Rolle, die das KI-Modell übernehmen soll.
- **Audience:** Bestimmung der Zielgruppe des Outputs.
- **Format:** Vorgabe der gewünschten Struktur des Outputs (z.B. Liste, Aufsatz, Stichworte in Spiegelstrichen, direkte Aufzählung mit Ziffern, ...)
- **Tone:** Festlegung der gewünschten Stimmung oder des Tons (z.B. informell, formell, salopp, kameradschaftlich, ...)

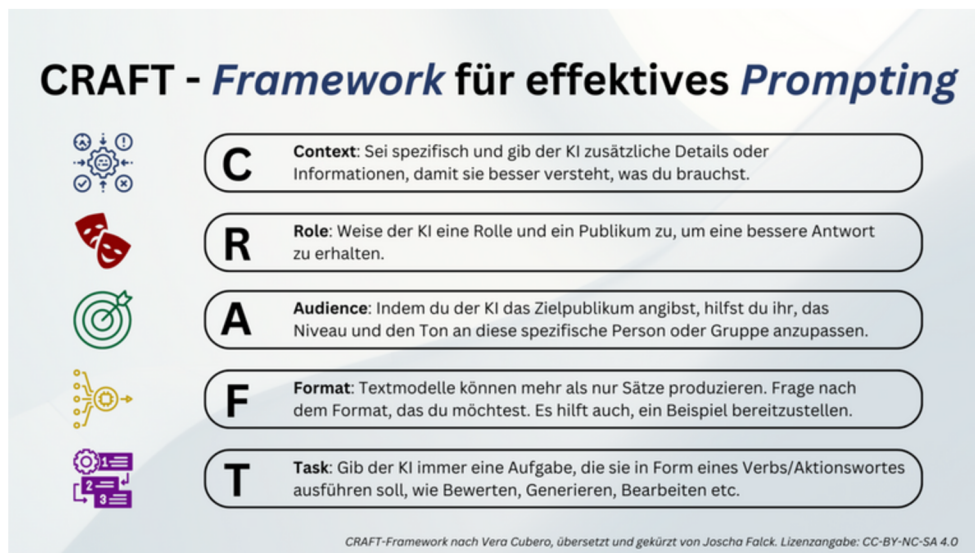


Abb. 4 - Prompt-Framework Craft (vgl. Falck 2025)

PREP-Framework (nach Falck 2025):

- **Point:** Zentraler Aspekt oder Hauptaussage des Prompts.
- **Reason:** Begründung oder Kontext für den Hauptpunkt
- **Example:** Beispiel zur Veranschulichung des Punktes
- **Point (restarted):** Wiederholung oder Klarstellung des zentralen Aspekts zur Betonung.

Prompt - Prompting-Frameworks

Formulierung von Mega-Prompts

Das letzte Prompting-Framework, das ich in dieser fachübergreifenden Thematik anführen möchte ist die Formulierung von sog. Mega-Prompts. Mega-Prompts wurden von Rob Lennon in einem Post auf der Plattform x beschrieben (vgl. Lennon 2023).

v2.02

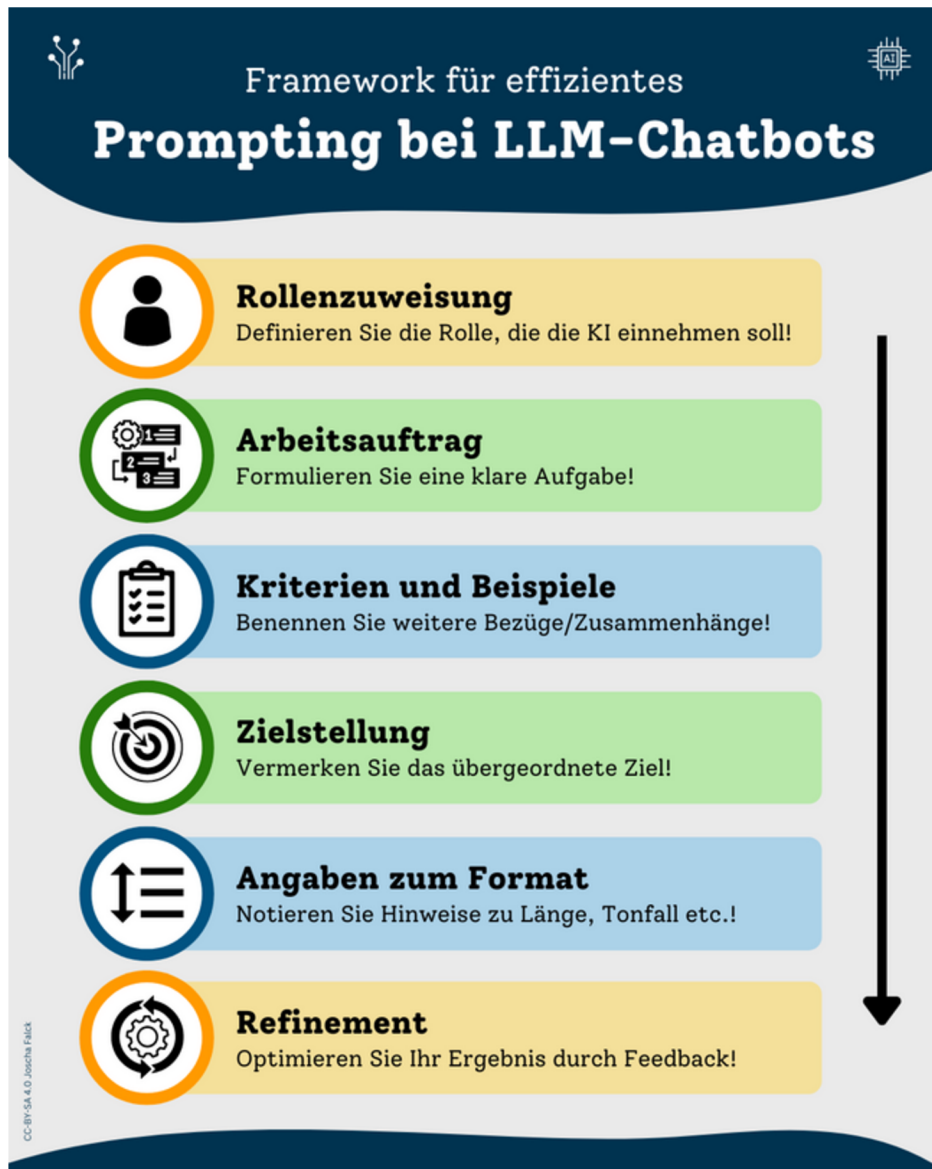


Abb. 5 - Framework für Mega-Prompts (Falck 2025)



Quellen

v2.02

- **Agarwal et al.** (2024). *Many-Shot In-Context Learning*. <https://doi.org/10.48550/ARXIV.2404.11018>
- **Falck, J.** (2025). *Prompting verstehen*. Blogartikel. Internet: <https://joschafalck.de/prompting-verstehen>. Abgerufen: 22.01.2025
- **Hutter, T.** (2024): *9 PROMPT FRAMEWORKS FÜR CHATGPT / GEMINI / COPILOT*. Internet: <https://www.thomashutter.com/9-prompt-frameworks-fuer-chatgpt-gemini-copilot/>. Abgerufen: 22.01.2025
- **Lennon, R.** (2023): *Anatomy of a ChatGPT Mega-Prompt*. Internet: <https://x.com/thatroblennon/status/1615104249192488980>. Abgerufen: 22.01.2025.
- **Schulhoff et al.** (2024). *The Prompt Report: A Systematic Survey of Prompting Techniques*. <https://doi.org/10.48550/ARXIV.2406.06608>
- **Xu et al.** (2023). *Re-Reading Improves Reasoning in Large Language Models*. <https://doi.org/10.48550/ARXIV.2309.06275>