

Nützliche Prompt-Techniken in der Zusammenfassung:

1) Zero-Shot Prompting / Few-Shot Prompting

Was es besonders macht:

Zero-Shot: Aufgabe nur mit Anweisung, ohne Beispiel.

Few-Shot: 2–5 Beispiele zeigen Zielstil/Format und reduzieren Varianz.

Beispiel (Few-Shot):

Beispiel 1: Frage → „Ist die Studie experimentell?“ Antwort → „Ja, wegen Randomisierung.“

Beispiel 2: Frage → „Ist die Studie experimentell?“ Antwort → „Nein, reine Beobachtungsstudie.“

Neue Eingabe: „Ist die folgende Studie experimentell? [Abstract] Antworte mit Ja/Nein + 1 Satz Begründung.“

Bewertung/Einordnung:

Zero-Shot schnell für Routine (Zusammenfassen, Umschreiben). Few-Shot empfehlenswert für konsistente Kategorisierung, Formatierung oder Stilvorgaben in Seminaren/Tutorien. Limit: Beispiele müssen gut gewählt sein; bei komplexen Aufgaben reichen wenige Shots nicht immer.

2) Reverse Prompt Engineering

Was es besonders macht:

Vom Output rückwärts denken: Aus einem Text/Stil ableiten, welcher Prompt ihn wahrscheinlich erzeugt hat; nützlich zum Analysieren und Replizieren von Ergebnissen.

Beispiel:

„Analysiere den folgenden KI-Text. Rekonstruiere einen wahrscheinlichen Prompt (Rolle, Ziel, Format, Constraints) und schlage 2 Verbesserungen vor. Text: ‘...’“

Bewertung/Einordnung:

Hilfreich, um Studierenden Prompt-Strukturen bewusst zu machen (Transparenz, Reproduzierbarkeit, Quellenkritik). Gefahr der Überanpassung an Stil; nicht verwechseln mit tatsächlicher Methodendokumentation.

3) Chain-of-Thought Prompting

Was es besonders macht:

Fordert explizite Zwischenschritte/Begründungen an, um logische Konsistenz zu erhöhen.

Beispiel:

„Löse die Statistikaufgabe schrittweise: 1) Hypothesen 2) Testwahl begründen 3) Test durchführen 4) Ergebnis interpretieren. Gib danach eine kurze Endzusammenfassung (2 Sätze).“

Bewertung/Einordnung:

Sehr geeignet zum Lehren von Problemlöseprozessen (Mathe, Methodik, Programmierung). In Prüfungen klare Regeln; ggf. nur kurze Begründungen statt ausführlicher Denkprotokolle verlangen.

4) Role Prompting

Was es besonders macht:

Weist dem Modell eine Fach-/Rollenperspektive zu, die Ton, Tiefe und Prioritäten steuert.

Beispiel:

„Du bist Studienberaterin. Erstelle für Erstsemester Informatik einen 10-Punkte-Plan zum Zeitmanagement, inkl. 2 evidenzbasierter Tipps und 1 Warnhinweis.“

Bewertung/Einordnung:

Niedrigschwellig und wirksam für Niveau-/Stilanpassung in Lehre, Beratung, Feedback. Qualität hängt von klaren Rollenmerkmalen ab.

5) Option-based Prompting

Was es besonders macht:

Gibt feste Antwortoptionen/Entscheidungsbäume vor; zwingt zu Auswahl und Begründung innerhalb definierter Alternativen.

Beispiel:

„Wähle genau eine Option (A–D) und begründe in 1–2 Sätzen: A) Definition unklar B) Methode unpassend C) Ergebnis unplausibel D) alles stimmig. Text: '...' Antworte im Format {wahl: 'A|B|C|D', begründung: '...'}.“

Bewertung/Einordnung:

Ideal für MC-ähnliche Übungen, Peer-Review-Raster, schnelle Screening-Entscheidungen. Reduziert Abschweifen; Gefahr: zu enge Optionen können Nuancen überdecken.

6) Self-Consistency

Was es besonders macht:

Mehrere unabhängige Lösungspfade/Antworten generieren und die häufigste/beste wählen; erhöht Robustheit.

Beispiel:

„Erzeuge 5 voneinander unabhängige Lösungsentwürfe für die Beweisidee. Wähle danach den konsistentesten Entwurf und gib nur die finale Lösung + 2-Satz-Begründung aus.“

Bewertung/Einordnung:

Gut für knifflige Aufgaben (Logik, Beweise, Quant-Methoden). Rechenaufwendiger; in der Lehre nutzbar, um Unsicherheit und Alternativen zu thematisieren.

7) Meta-Prompts & Prompt-Templates

Was es besonders macht:

Wiederverwendbare Schablonen mit Platzhaltern; standardisieren Qualität und Format, erleichtern Kollaboration.

Beispiel:

„Template: Rolle={...}; Ziel={...}; Eingabequelle={...}; Output-Format={Bullets|Tabelle|JSON}; Constraints={Länge, Zitation}.

Fülle das Template für 'Seminar-Abstract-Feedback' aus.“

Bewertung/Einordnung:

Sehr nützlich für Lehrteams (einheitliche Rubriken, Aufgabenstellungen, Feedbackformen) und für Studierende (Strukturhilfe). Pflege der Templates ist wichtig.

8) Tree of Thoughts (ToT)

Was es besonders macht:

Erzeugt mehrere Lösungsäste/Strategien, bewertet Zwischenschritte und wählt den besten Pfad (Divergenz → Konvergenz).

Beispiel:

„Entwickle drei alternative Forschungsfragen zu 'KI in der Pflege'. Für jeden Ast: (a) Frage (b) mögliche Methoden (c) Risiken. Bewerte die Äste nach Machbarkeit und Impact (je 1–5) und wähle den besten mit Begründung.“

Bewertung/Einordnung:

Stark für Ideation, Forschungsdesign, Projektplanung. Fördert reflektierte Entscheidung; etwas komplexer anzuleiten und zeitintensiver.

Kurzfasit für Hochschulkontexte

- Für Einstieg/Standardaufgaben: Role, Zero-/Few-Shot, Option-based, Templates.
- Für Problemlösen und Forschung: Chain-of-Thought, Self-Consistency, ToT.
- Für Qualität/Transparenz: Reverse Prompt Engineering (Analyse), Templates/Raster.