

Prompt Engineering für die Praxis

Sprachmodelle gezielt und reproduzierbar einsetzen

01 INTRO

Gute Ergebnisse mit Large Language Models entstehen nicht allein durch möglichst ausgefeilte Formulierungen. Entscheidend ist, eine Aufgabe so zu strukturieren, dass Ziel, Kontext, Randbedingungen und erwartetes Ergebnis für das Modell möglichst eindeutig sind.

Der Kurs vermittelt Prompt Engineering als systematische Arbeitsweise. Die Teilnehmer entwickeln Prompts für unterschiedliche Aufgaben, untersuchen das Verhalten von Sprachmodellen und lernen, komplexere Anforderungen schrittweise aufzubauen, Ergebnisse zu überprüfen und wiederverwendbare Prompt-Strukturen zu entwickeln.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Grundlegende Erfahrung mit ChatGPT, Claude oder vergleichbaren Sprachmodellen

02 TRAININGSPROGRAMM

Inhalte

Grundlagen des Prompt Engineering

- Prompt Engineering als Schnittstelle zwischen Mensch und Modell
- Prompt, Kontext und Modellantwort
- Instruktionen und Informationen unterscheiden
- Aufgaben eindeutig formulieren
- Einfluss von Modell und Kontext
- deterministische und probabilistische Systeme
- typische Ursachen schlechter Ergebnisse
- Grenzen des Prompt Engineering

Aufbau guter Prompts

- Ziel und Aufgabe definieren
- notwendigen Kontext bereitstellen
- Randbedingungen formulieren
- Rollen sinnvoll einsetzen
- gewünschte Ausgabe beschreiben
- Beispiele bereitstellen
- Prioritäten deutlich machen
- unnötige Informationen vermeiden

Instruktionen präzisieren

- mehrdeutige Anforderungen erkennen
- implizite Annahmen vermeiden
- positive und negative Vorgaben
- Muss-, Soll- und Kann-Anforderungen
- Bedingungen und Ausnahmen
- Abbruch- und Eskalationsbedingungen
- Umgang mit fehlenden Informationen
- Rückfragen gezielt ermöglichen

Zero-Shot und Few-Shot Prompting

- Aufgaben ohne Beispiele formulieren
- Beispiele als Orientierung verwenden
- Auswahl geeigneter Beispiele
- positive und negative Beispiele
- Format und Inhalt durch Beispiele steuern
- typische Few-Shot-Fehler
- Anzahl und Qualität der Beispiele
- Grenzen beispielbasierter Steuerung

Strukturierte Ein- und Ausgaben

- Listen und Tabellen
- Markdown
- JSON und andere strukturierte Formate
- vorgegebene Schemas
- Pflichtfelder und optionale Felder
- Ausgabeformate validierbar gestalten
- maschinelle Weiterverarbeitung vorbereiten
- robuste statt nur optisch plausible Ergebnisse

Kontext gezielt aufbauen

- relevanter und irrelevanter Kontext
- Reihenfolge von Informationen
- Kontextumfang
- lange Dokumente
- mehrere Informationsquellen
- Trennung von Daten und Instruktionen
- Kontext reduzieren und verdichten
- Auswirkungen großer Kontextmengen

Komplexe Aufgaben zerlegen

- große Aufgaben in Teilschritte zerlegen
- Zwischenprodukte definieren
- Analyse und Ergebnis trennen
- mehrstufige Prompt-Verfahren
- Ergebnisse eines Schrittes weiterverwenden
- Prüf- und Korrekturschritte einbauen
- wann ein einzelner Prompt nicht mehr sinnvoll ist
- Übergang zu Workflows

Iteratives Prompt Engineering

- mit einer einfachen Version beginnen
- Ergebnisse analysieren
- Fehlerklassen identifizieren
- Prompt gezielt verändern
- Änderungen einzeln untersuchen
- Varianten vergleichen
- Regressionen erkennen
- erfolgreiche Prompts stabilisieren

Prompts testen

- Testfälle für Prompts entwickeln
- typische und atypische Eingaben
- Grenzfälle
- unterschiedliche Formulierungen
- erwartete Eigenschaften der Antwort
- wiederholte Ausführung
- Modellvariabilität berücksichtigen
- Prompt-Änderungen systematisch vergleichen

Qualität bewerten

- fachliche Korrektheit
- Vollständigkeit
- Relevanz
- Konsistenz
- Einhaltung von Vorgaben
- Formatkonformität
- Nachvollziehbarkeit
- geeignete Bewertungskriterien definieren

Halluzinationen und Unsicherheit

- fehlendes Wissen und erfundene Antworten
- Modelle nicht zum Raten zwingen
- fehlende Informationen explizit behandeln
- Quellen und Belege anfordern
- Antworten gegen vorhandene Informationen prüfen
- Unsicherheit sichtbar machen
- Verifikation als Teil des Arbeitsablaufs
- Grenzen promptbasierter Absicherung

Wiederverwendbare Prompts

- Prompt Templates
- variable und feste Bestandteile
- Parameter und Platzhalter
- Prompts für unterschiedliche Anwendungsfälle
- Versionierung
- Dokumentation
- gemeinsame Prompt-Bibliotheken
- Wartung und Weiterentwicklung

System Prompts und Instruktionshierarchien

- Benutzer- und Systeminstruktionen
- unterschiedliche Instruktionsebenen
- dauerhafte Rahmenbedingungen
- Rollen und Verhaltensregeln
- Konflikte zwischen Instruktionen
- Grenzen der Steuerbarkeit
- System Prompts in Anwendungen
- Unterschiede zur normalen Chat-Nutzung

Prompt Injection und untrusted Input

- Prompt Injection als Problem
- direkte und indirekte Prompt Injection
- fremde Inhalte als potenzielle Instruktionen
- Dokumente, Webseiten und externe Daten
- Trennung von Instruktionen und Daten
- warum Prompt-Regeln allein keine Sicherheitsgrenze darstellen
- technische und organisatorische Schutzmaßnahmen
- Grenzen promptbasierter Sicherheitsmechanismen

Prompting für typische Aufgaben

- Informationen extrahieren
- Inhalte klassifizieren
- Texte zusammenfassen
- Texte überarbeiten
- Informationen vergleichen
- Daten strukturieren
- Varianten und Vorschläge erzeugen
- Analyseaufgaben unterstützen

Prompting mit Dokumenten

- Dokumente als Kontext bereitstellen
- gezielte Fragen zu Dokumenten
- Informationen extrahieren
- mehrere Dokumente vergleichen
- umfangreiche Dokumente zerlegen
- Quellenbezug erhalten
- fehlende Informationen erkennen
- Ergebnisse gegen Originalmaterial prüfen

Prompting mit multimodalen Modellen

- Text und Bild gemeinsam verwenden
- Screenshots analysieren
- Diagramme und Darstellungen beschreiben
- visuelle Informationen extrahieren
- Bild und Text als gemeinsamen Kontext behandeln
- präzise Aufgaben für multimodale Modelle
- typische Fehler
- Grenzen der visuellen Interpretation

Vom Prompt zum Workflow

- Grenzen einzelner Prompts
- Prompt Chains
- strukturierte Verarbeitungsschritte
- externe Daten einbeziehen
- Tools und Funktionen
- RAG im Überblick
- Agenten im Überblick
- wann Prompt Engineering zu AI Engineering wird

Praktische Übungen

- vorhandene Prompts analysieren
- unklare Prompts verbessern
- Anforderungen systematisch formulieren
- Zero-Shot und Few-Shot vergleichen
- strukturierte Ausgaben erzeugen
- komplexe Aufgaben zerlegen
- Prompts mit Testfällen prüfen
- eigene Anwendungsfälle der Teilnehmer bearbeiten

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	3 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit Zugang zu einem aktuellen Sprachmodell

Inhouse-Schulungen

Vorhandene Prompts, typische Aufgaben und unternehmensspezifische Anwendungsfälle können in die Übungen einbezogen werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Der Kurs richtet sich an Fachanwender, Entwickler, Projektmitarbeiter, Consultants und andere Nutzer generativer KI, die Sprachmodelle nicht nur spontan im Chat einsetzen, sondern Aufgaben systematisch und reproduzierbar formulieren möchten.

Er eignet sich insbesondere für Teilnehmer, die bereits erste Erfahrungen mit ChatGPT, Claude oder vergleichbaren Werkzeugen gesammelt haben und die Qualität ihrer Ergebnisse gezielt verbessern möchten.

Voraussetzungen

Grundlegende praktische Erfahrung mit einem Large Language Model wird vorausgesetzt. Programmierkenntnisse sind nicht erforderlich. Für Teilnehmer aus der Softwareentwicklung können Übungen und Beispiele entsprechend technisch ausgerichtet werden.

05 LERNZIELE

Was Sie nach dem Kurs können

Nach dem Kurs können die Teilnehmer:

- Aufgaben für Sprachmodelle strukturiert formulieren
- Ziel, Kontext und Randbedingungen klar voneinander unterscheiden
- Zero-Shot- und Few-Shot-Verfahren sinnvoll einsetzen
- strukturierte und maschinell weiterverarbeitbare Ausgaben erzeugen
- komplexe Aufgaben in geeignete Verarbeitungsschritte zerlegen
- Prompts iterativ entwickeln und systematisch testen
- Qualitätskriterien für Modellantworten definieren
- Halluzinationen und typische Fehlerquellen berücksichtigen
- wiederverwendbare Prompt Templates entwickeln
- System Prompts und Instruktionshierarchien einordnen
- Risiken durch Prompt Injection erkennen
- Prompts für Dokumente und multimodale Modelle entwickeln
- erkennen, wann ein Problem nicht mehr sinnvoll allein durch Prompt Engineering gelöst werden kann

06 FAQ

Häufige Fragen

Ist der Kurs auch für Teilnehmer mit wenig KI-Erfahrung geeignet?

Grundlegende Erfahrung mit ChatGPT, Claude oder einem vergleichbaren Sprachmodell sollte vorhanden sein. Tiefergehende technische Kenntnisse sind nicht erforderlich.

Muss ich programmieren können?

Nein. Die grundlegenden Methoden des Prompt Engineering sind unabhängig von einer Programmiersprache. Technische Beispiele können bei Bedarf ergänzt werden.

Werden fertige Prompt-Vorlagen vermittelt?

Der Kurs verwendet und entwickelt wiederverwendbare Prompt-Strukturen. Im Mittelpunkt steht jedoch das Verständnis, wie solche Prompts aufgebaut, geprüft und an neue Aufgaben angepasst werden, nicht das Sammeln allgemeiner „Magic Prompts“.

Geht es um bestimmte Modelle wie ChatGPT oder Claude?

Nein. Die grundlegenden Verfahren werden modellübergreifend behandelt. Unterschiede zwischen Modellen können in den praktischen Übungen untersucht werden.

Werden auch System Prompts behandelt?

Ja. System Prompts, Instruktionshierarchien und dauerhafte Rahmenbedingungen werden behandelt und gegenüber normalen Benutzerprompts abgegrenzt.

Wird Prompt Injection behandelt?

Ja. Der Kurs zeigt, warum externe Inhalte problematisch werden können und weshalb Prompting allein keine verlässliche Sicherheitsgrenze darstellt.

Werden Prompts auch systematisch getestet?

Ja. Testfälle, Varianten, Grenzfälle und Qualitätskriterien sind Bestandteil des Kurses. Prompt Engineering wird damit nicht nur als Formulierungsaufgabe, sondern als iterativer Entwicklungsprozess betrachtet.

Kann der Kurs an konkrete Anwendungsfälle angepasst werden?

Ja. Bei Inhouse-Schulungen können vorhandene Prompts, typische Aufgaben und unternehmensspezifische Anwendungsfälle in die Übungen einbezogen werden.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de

<https://www.uc-it.de/coaching/prompt-engineering/>