

Prompt Engineering: Taktiken & Strategien

Dieser Leitfaden fasst bewährte Methoden und Taktiken des Prompt Engineering zusammen, um Modelle der Künstlichen Intelligenz effektiver zu nutzen. Die folgenden Strategien und Taktiken bieten Ideen, wie Sie präzisere und relevantere Antworten erhalten können.

Strategie 1: Gib klare Anweisungen



Taktik A: Details hinzufügen, um relevantere Antworten zu erhalten

Um eine hochrelevante Antwort zu erhalten, stellen Sie sicher, dass Ihre Anfragen alle wichtigen Details oder Kontextinformationen enthalten. Andernfalls überlassen Sie es dem Modell, zu erraten, was Sie meinen.

Schlechter: *Wie addiere ich Zahlen in Excel?*

Besser: *Wie addiere ich eine Zeile von Dollarbeträgen in Excel? Ich möchte dies automatisch für ein ganzes Blatt mit Zeilen tun, wobei alle Summen in einer Spalte namens „Total“ auf der rechten Seite erscheinen sollen.*



Taktik B: Bitte das Modell, eine Persona anzunehmen

Die Systemnachricht kann verwendet werden, um die Persona festzulegen, die das Modell in seinen Antworten verwenden soll.

SYSTEM: Wenn ich um Hilfe beim Schreiben von etwas bitte, antwortest du mit einem Dokument, das in jedem Absatz mindestens einen Witz oder einen spielerischen Kommentar enthält.

USER: Schreibe eine Danksagung an meinen Stahlbolzenlieferanten dafür, dass er die Lieferung pünktlich und kurzfristig erledigt hat. Dies hat es uns ermöglicht, einen wichtigen Auftrag auszuliefern.



Taktik C: Verwende Trennzeichen (Delimiters)

Trennzeichen wie dreifache Anführungszeichen (``), XML-Tags oder Abschnittsüberschriften können helfen, Textabschnitte abzugrenzen, die unterschiedlich behandelt werden sollen.

USER: Fasse den Text, der durch dreifache Anführungszeichen begrenzt ist, in einem Haiku zusammen.

""""Text hier einfügen""""



Taktik D: Gib die Schritte an, die zur Erfüllung einer Aufgabe erforderlich sind

Manche Aufgaben lassen sich am besten als Abfolge von Schritten definieren. Das explizite Aufschreiben der Schritte kann es dem Modell erleichtern, ihnen zu folgen.

SYSTEM: Verwende die folgenden schrittweisen Anweisungen, um auf Benutzereingaben zu antworten.

Schritt 1 - Der Benutzer stellt dir Text in dreifachen Anführungszeichen zur Verfügung. Fasse diesen Text in einem Satz zusammen, der das Präfix "Summary:" trägt.

Schritt 2 - Übersetze die Zusammenfassung aus Schritt 1 ins Spanische, mit dem Präfix "Translation:".

USER: """"Text hier einfügen""""



Taktik E: Gib Beispiele (Few-Shot Prompting)

Das Bereitstellen von Beispielen kann einfacher sein, als eine Aufgabe explizit zu beschreiben. Dies ist besonders nützlich, wenn das Modell einen bestimmten Antwortstil nachahmen soll.

SYSTEM: Antworte in einem konsistenten Stil.

USER: Erkläre mir Geduld.

ASSISTANT: Der Fluss, der das tiefste Tal formt, entspringt einer bescheidenen Quelle; die größte Symphonie beginnt mit einer einzigen Note; der komplizierteste Wandteppich beginnt mit einem einzigen Faden.

USER: Erkläre mir den Ozean.



Taktik F: Gib die gewünschte Länge der Ausgabe an

Sie können das Modell bitten, Ausgaben einer bestimmten Länge zu produzieren, z. B. in Bezug auf die Anzahl der Wörter, Sätze, Absätze oder Aufzählungspunkte.

USER: Fasse den Text, der durch dreifache Anführungszeichen begrenzt ist, in 2 Absätzen zusammen.

"""Text hier einfügen"""

USER: Fasse den Text, der durch dreifache Anführungszeichen begrenzt ist, in 3 Aufzählungspunkten zusammen.

"""Text hier einfügen"""

Strategie 2: Zerlege komplexe Aufgaben in einfachere Unteraufgaben



Taktik G: Nutze Intent Classification, um relevante Anweisungen zu identifizieren

Bei Aufgaben, die viele unabhängige Anweisungssätze erfordern, um verschiedene Fälle zu behandeln, kann es von Vorteil sein, zuerst die Art der Anfrage zu klassifizieren.

UNTERAUFGABE 1

SYSTEM: Du erhältst Kundenanfragen. Klassifiziere jede Anfrage in eine primäre und eine sekundäre Kategorie. Stelle deine Ausgabe im JSON-Format mit den Schlüsseln "primary" und "secondary" bereit.

USER: Ich muss mein Internet wieder zum Laufen bringen.

Basierend auf der Klassifizierung kann eine Reihe spezifischerer Anweisungen für die nächsten Schritte bereitgestellt werden.



Taktik H: Fasse bei langen Konversationen den Dialog zusammen oder filtere ihn

Da Modelle eine feste Kontextlänge haben, kann ein Dialog nicht unbegrenzt fortgesetzt werden. Eine Lösung besteht darin, frühere Gesprächsrunden zusammenzufassen.

Strategie 3: Fasse lange Dokumente stückweise zusammen

Da Modelle eine feste Kontextlänge haben, können sie einen Text, der länger ist als der Kontext, nicht in einer einzigen Anfrage zusammenfassen. Sie können den Text in Abschnitte unterteilen, diese einzeln zusammenfassen und die Zusammenfassungen rekursiv zusammenfassen.

Strategie 4: Gib dem Modell Zeit zum "Nachdenken"



Taktik I: Weise das Modell an, eine eigene Lösung zu erarbeiten

Manchmal erzielen wir bessere Ergebnisse, wenn wir das Modell explizit anweisen, von den ersten Prinzipien aus zu argumentieren, bevor es zu einer Schlussfolgerung kommt.

SYSTEM: Erarbeite zuerst deine eigene Lösung für das Problem. Vergleiche dann deine Lösung mit der Lösung des Schülers und bewerte, ob die Lösung des Schülers korrekt ist oder nicht. Triff keine Entscheidung, ob die Lösung des Schülers korrekt ist, bevor du das Problem nicht selbst gelöst hast.



Taktik J: Frage das Modell, ob es in vorherigen Durchgängen etwas übersehen hat

Wenn das Modell Exzerpte aus einer großen Quelle auflistet, neigt es manchmal dazu, zu früh aufzuhören. Bessere Leistung kann erzielt werden, indem man das Modell mit Folgeanfragen auffordert, weitere übersehene Exzerpte zu finden.

Strategie 5: Nutze externe Tools



Taktik K: Nutze Code-Ausführung für genauere Berechnungen oder API-Aufrufe

Modelle können nicht darauf vertrauen, dass sie selbständig genaue Rechenaufgaben ausführen. In solchen Fällen kann ein Modell angewiesen werden, stattdessen Code zu schreiben und auszuführen.

```
SYSTEM: Du kannst Python-Code schreiben und ausführen, indem du ihn in dreifache Backticks einschließt, z. B. ```code here```. Verwende dies, um Berechnungen durchzuführen.
```



Taktik L: Gib dem Modell Zugang zu spezifischen Funktionen

Die Chat Completions API ermöglicht die Übergabe einer Liste von Funktionsbeschreibungen, wodurch Modelle Funktionsargumente generieren können, die dann für Funktionsaufrufe verwendet werden können.

Bereitzustellender Prompt zur Verbesserung:

```
"""
```

```
[HIER DER ZU VERBESSERENDE PROMPT]
```

```
"""
```

