

Prompt engineering

(Übersetzt mit dem Microsoft Bing Übersetzer am 20.12.2023. [Originaltext in Englisch](#) bei OpenAI)

Inhalt

Sechs Strategien, um bessere Ergebnisse zu erzielen	2
Schreiben Sie klare Anweisungen	2
Referenztext angeben	2
Teilen Sie komplexe Aufgaben in einfachere Teilaufgaben auf	2
Geben Sie dem Modell Zeit zum "Nachdenken"	3
Externe Tools verwenden	3
Änderungen systematisch testen	3
Taktik	4
Strategie: Schreiben Sie klare Anweisungen	4
Strategie: Referenztext bereitstellen	7
Strategie: Komplexe Aufgaben in einfachere Teilaufgaben aufteilen	8
Strategie: Den Modellen Zeit zum "Nachdenken" geben	10
Strategie: Externe Tools nutzen	12
Strategie: Änderungen systematisch testen	14
Weitere Ressourcen	17

In diesem Leitfaden werden Strategien und Taktiken vorgestellt, mit denen Sie mit großen Sprachmodellen (manchmal auch als GPT-Modelle bezeichnet) wie GPT-4 bessere Ergebnisse erzielen können. Die hier beschriebenen Methoden können manchmal in Kombination eingesetzt werden, um eine größere Wirkung zu erzielen. Wir ermutigen zum Experimentieren, um die Methoden zu finden, die für Sie am besten geeignet sind.

Einige der hier gezeigten Beispiele funktionieren derzeit nur mit unserem leistungsfähigsten Modell, . Wenn Sie feststellen, dass ein Modell bei einer Aufgabe fehlschlägt und ein leistungsfähigeres Modell verfügbar ist, lohnt es sich im Allgemeinen, es mit dem leistungsfähigeren Modell erneut zu versuchen.^{gpt-4} Sie können sich auch Beispielaufforderungen ansehen, die zeigen, wozu unsere Modelle in der Lage sind:

Beispiele für Eingabeaufforderungen

Sehen Sie sich Beispiele an, um zu erfahren, was GPT-Modelle leisten können

Sechs Strategien, um bessere Ergebnisse zu erzielen

Schreiben Sie klare Anweisungen

Diese Modelle können Ihre Gedanken nicht lesen. Wenn die Ausgaben zu lang sind, bitten Sie um kurze Antworten. Wenn die Ausgaben zu einfach sind, bitten Sie um das Schreiben auf Expertenniveau. Wenn Ihnen das Format nicht gefällt, zeigen Sie das Format, das Sie sehen möchten. Je weniger das Model erraten muss, was Sie wollen, desto wahrscheinlicher ist es, dass Sie es bekommen.

Taktik:

- [Fügen Sie Details in Ihre Abfrage ein, um relevantere Antworten zu erhalten](#)
- [Bitten Sie das Model, eine Persona anzunehmen](#)
- [Verwenden Sie Trennzeichen, um unterschiedliche Teile der Eingabe deutlich zu kennzeichnen](#)
- [Angeben der Schritte, die zum Ausführen einer Aufgabe erforderlich sind](#)
- [Beispiele nennen](#)
- [Geben Sie die gewünschte Länge der Ausgabe an](#)

Referenztext angeben

Sprachmodelle können selbstbewusst falsche Antworten erfinden, insbesondere, wenn sie nach esoterischen Themen oder nach Zitaten und URLs gefragt werden. Auf die gleiche Weise, wie ein Blatt mit Notizen einem Schüler helfen kann, bei einem Test besser abzuschneiden, kann die Bereitstellung von Referenztext zu diesen Modellen dazu beitragen, mit weniger Erfindungen zu antworten.

Taktik:

- [Weisen Sie das Modell an, mit einem Referenztext zu antworten](#)
- [Weisen Sie das Modell an, mit Zitaten aus einem Referenztext zu antworten](#)

Teilen Sie komplexe Aufgaben in einfachere Teilaufgaben auf

So wie es in der Softwareentwicklung üblich ist, ein komplexes System in eine Reihe modularer Komponenten zu zerlegen, gilt dies auch für Aufgaben, die einem Sprachmodell vorgelegt werden. Komplexe Aufgaben weisen tendenziell höhere Fehlerraten auf als einfachere Aufgaben. Darüber hinaus können komplexe Aufgaben oft als ein Workflow einfacherer Aufgaben neu definiert werden, in dem die Ausgaben früherer Aufgaben verwendet werden, um die Eingaben für spätere Aufgaben zu konstruieren.

Taktik:

- [Verwenden Sie die Absichtsklassifizierung, um die relevantesten Anweisungen für eine Benutzerabfrage zu identifizieren](#)

- [Für Dialoganwendungen, die sehr lange Gespräche erfordern, fassen Sie vorherige Dialoge zusammen oder filtern Sie sie](#)
- [Fassen Sie lange Dokumente stückweise zusammen und konstruieren Sie rekursiv eine vollständige Zusammenfassung](#)

Geben Sie dem Modell Zeit zum "Nachdenken"

Wenn du gebeten wirst, 17 mit 28 zu multiplizieren, weißt du es vielleicht nicht sofort, kannst es aber mit der Zeit trotzdem herausfinden. In ähnlicher Weise machen Modelle mehr Denkfehler, wenn sie versuchen, sofort zu antworten, anstatt sich Zeit zu nehmen, um eine Antwort zu erarbeiten. Die Frage nach einer "Gedankenkette" vor einer Antwort kann dem Modell helfen, sich zuverlässiger auf den Weg zu richtigen Antworten zu machen.

Taktik:

- [Weisen Sie das Modell an, seine eigene Lösung zu erarbeiten, bevor Sie voreilig zu einer Schlussfolgerung kommen](#)
- [Verwenden Sie einen inneren Monolog oder eine Sequenz von Abfragen, um den Denkprozess des Modells zu verbergen](#)
- [Fragen Sie das Modell, ob es bei früheren Durchläufen etwas übersehen hat](#)

Externe Tools verwenden

Kompensieren Sie die Schwächen des Modells, indem Sie es mit den Ergebnissen anderer Tools füttern. Beispielsweise kann ein Text-Retrieval-System (manchmal auch als RAG oder Retrieval Augmented Generation bezeichnet) dem Modell Informationen über relevante Dokumente geben. Eine Codeausführungs-Engine wie der Code Interpreter von OpenAI kann dem Modell helfen, Berechnungen durchzuführen und Code auszuführen. Wenn eine Aufgabe durch ein Tool zuverlässiger oder effizienter als durch ein Sprachmodell erledigt werden kann, lagern Sie sie aus, um das Beste aus beiden herauszuholen.

Taktik:

- [Verwenden Sie die einbettungsbasierte Suche, um eine effiziente Wissensbeschaffung zu implementieren](#)
- [Verwenden Sie die Codeausführung, um genauere Berechnungen durchzuführen oder externe APIs aufzurufen](#)
- [Gewähren des Modells Zugriff auf bestimmte Funktionen](#)

Änderungen systematisch testen

Die Verbesserung der Leistung ist einfacher, wenn Sie sie messen können. In einigen Fällen führt eine Änderung an einer Eingabeaufforderung zu einer besseren Leistung bei einigen isolierten Beispielen, führt jedoch zu einer schlechteren Gesamtleistung bei einer repräsentativeren Gruppe von Beispielen. Um sicherzustellen, dass sich eine Änderung positiv auf die Leistung auswirkt, kann es daher erforderlich sein, eine umfassende Testsuite (auch als "Evalu" bezeichnet) zu definieren.

Taktik:

- [Evaluieren Sie die Ergebnisse des Modells unter Bezugnahme auf Goldstandard-Antworten](#)

Taktik

Jede der oben aufgeführten Strategien kann mit bestimmten Taktiken instanziiert werden. Diese Taktiken sollen Ideen für Dinge liefern, die man ausprobieren kann. Sie sind keineswegs vollständig, und Sie sollten sich frei fühlen, kreative Ideen auszuprobieren, die hier nicht vertreten sind.

Strategie: Schreiben Sie klare Anweisungen

Taktik: Fügen Sie Details in Ihre Abfrage ein, um relevantere Antworten zu erhalten

Um eine hochrelevante Antwort zu erhalten, stellen Sie sicher, dass die Anfragen wichtige Details oder Kontext enthalten. Andernfalls überlassen Sie es dem Modell, zu erraten, was Sie meinen.

Schlimmer

Besser

Wie füge ich Zahlen in Excel hinzu?

Wie addiere ich eine Zeile von Dollarbeträgen in Excel?? Ich möchte dies automatisch für ein ganzes Blatt mit Zeilen tun, wobei alle Summen rechts in einer Spalte namens "Summe" landen.

Wer ist Präsident?

Wer war 2021 Präsident von Mexiko und wie oft finden Wahlen statt?

Schreiben Sie Code, um die Fibonacci-Folge zu berechnen.

Schreiben Sie eine TypeScript-Funktion, um die Fibonacci-Folge effizient zu berechnen. Kommentieren Sie den Code großzügig, um zu erklären, was jedes Stück tut und warum es so geschrieben ist.

Fassen Sie die Besprechungsnotizen zusammen.

Fassen Sie die Besprechungsnotizen in einem einzigen Absatz zusammen. Schreiben Sie dann eine Markdown-Liste der Referenten und ihrer wichtigsten Punkte. Listen Sie abschließend die nächsten Schritte oder Aktionspunkte

auf, die von den Referenten vorgeschlagen wurden, falls vorhanden.

Taktik: Bitten Sie das Model, eine Persona anzunehmen

Die Systemnachricht kann verwendet werden, um die Persona anzugeben, die das Modell in seinen Antworten verwendet.

SYSTEM

Wenn ich um Hilfe bitte, um etwas zu schreiben, antworten Sie mit einem Dokument, das in jedem Absatz mindestens einen Witz oder einen spielerischen Kommentar enthält.

BENUTZER

Schreiben Sie eine Dankesnachricht an meinen Stahlbolzenlieferanten für die pünktliche und kurzfristige Lieferung. Dadurch war es uns möglich, einen wichtigen Auftrag auszuliefern.

[Im Playground öffnen](#)

Taktik: Verwenden Sie Trennzeichen, um bestimmte Teile der Eingabe deutlich zu kennzeichnen

Trennzeichen wie dreifache Anführungszeichen, XML-Tags, Abschnittsüberschriften usw. können helfen, Textabschnitte abzugrenzen, die unterschiedlich behandelt werden sollen.

BENUTZER

Fassen Sie den durch dreifache Anführungszeichen begrenzten Text mit einem Haiku zusammen. ""Text hier einfügen""

[Im Playground öffnen](#)

SYSTEM

Sie erhalten zwei Artikel (getrennt durch XML-Tags) zum gleichen Thema. Fassen Sie zunächst die Argumente der einzelnen Artikel zusammen. Geben Sie dann an, wer von ihnen ein besseres Argument liefert, und erklären Sie, warum.

BENUTZER

<Artikel> ersten Artikel hier einfügen </Artikel> <Artikel> hier einen zweiten Artikel einfügen </Artikel>

[Im Playground öffnen](#)

SYSTEM

Sie erhalten ein Abstract der Abschlussarbeit und einen Vorschlag für einen Titel. Der Titel der Abschlussarbeit sollte dem Leser eine gute Vorstellung vom Thema der Arbeit geben, aber auch auffällig sein. Wenn der Titel diese Kriterien nicht erfüllt, schlagen Sie 5 Alternativen vor.

BENUTZER

Abstract: Abstract hier einfügen Titel: Titel hier einfügen

[Im Playground öffnen](#)

Bei einfachen Aufgaben wie diesen macht die Verwendung von Trennzeichen möglicherweise keinen Unterschied in der Ausgabequalität. Je komplexer eine Aufgabe jedoch ist, desto wichtiger ist es, die Aufgabedetails zu verdeutlichen. Lassen Sie das Modell nicht funktionieren, um genau zu verstehen, was Sie von ihnen verlangen.

Taktik: Geben Sie die Schritte an, die zum Abschließen einer Aufgabe erforderlich sind

Einige Aufgaben lassen sich am besten als Abfolge von Schritten angeben. Wenn Sie die Schritte explizit aufschreiben, kann es für das Modell einfacher sein, ihnen zu folgen.

SYSTEM

Verwenden Sie die folgenden Schritt-für-Schritt-Anweisungen, um auf Benutzereingaben zu reagieren. Schritt 1 - Der Benutzer stellt Ihnen den Text in dreifachen Anführungszeichen zur Verfügung. Fassen Sie diesen Text in einem Satz mit dem Präfix "Zusammenfassung: " zusammen. Schritt 2 - Übersetzen Sie die Zusammenfassung von Schritt 1 ins Spanische mit dem Präfix "Übersetzung:".

BENUTZER

""Text hier einfügen""

[Im Playground öffnen](#)

Taktik: Beispiele nennen

Das Bereitstellen allgemeiner Anweisungen, die für alle Beispiele gelten, ist im Allgemeinen effizienter als das Demonstrieren aller Permutationen einer Aufgabe anhand eines Beispiels, aber in einigen Fällen kann das Bereitstellen von Beispielen einfacher sein. Wenn Sie z. B. beabsichtigen, dass das Modell einen bestimmten Stil der Antwort auf Benutzerabfragen kopiert, der schwer explizit zu beschreiben ist. Dies wird als "Few-Shot"-Eingabeaufforderung bezeichnet.

SYSTEM

Antworten Sie in einem einheitlichen Stil.

BENUTZER

Lehre mich etwas über Geduld.

ASSISTENT

Der Fluss, der das tiefste Tal gräbt, entspringt einer bescheidenen Quelle; Die großartigste Symphonie entspringt einem einzigen Ton; Der komplizierteste Wandteppich beginnt mit einem einzigen Faden.

BENUTZER

Bring mir etwas über den Ozean bei.

[Im Playground öffnen](#)

Taktik: Geben Sie die gewünschte Länge der Ausgabe an

Sie können das Modell auffordern, Ausgaben mit einer bestimmten Ziellänge zu erzeugen. Die angestrebte Ausgabelänge kann in Form der Anzahl von Wörtern, Sätzen, Absätzen, Aufzählungspunkten usw. festgelegt werden. Beachten Sie jedoch, dass die Anweisung an das Modell, eine bestimmte Anzahl von Wörtern zu generieren, nicht mit hoher Genauigkeit funktioniert. Das Modell kann zuverlässiger Ausgaben mit einer bestimmten Anzahl von Absätzen oder Aufzählungspunkten generieren.

BENUTZER

Fassen Sie den durch dreifache Anführungszeichen getrennten Text in etwa 50 Wörtern zusammen. ""Text hier einfügen""

[Im Playground öffnen](#)

BENUTZER

Fassen Sie den durch dreifache Anführungszeichen getrennten Text in 2 Absätzen zusammen. ""Text hier einfügen""

[Im Playground öffnen](#)

BENUTZER

Fassen Sie den durch dreifache Anführungszeichen getrennten Text in 3 Aufzählungspunkten zusammen. ""Text hier einfügen""

[Im Playground öffnen](#)

Strategie: Referenztext bereitstellen

Taktik: Weisen Sie das Modell an, mit einem Referenztext zu antworten

Wenn wir ein Modell mit vertrauenswürdigen Informationen bereitstellen können, die für die aktuelle Abfrage relevant sind, können wir das Modell anweisen, die bereitgestellten Informationen zu verwenden, um seine Antwort zu verfassen.

SYSTEM

Verwenden Sie die bereitgestellten Artikel, die durch dreifache Anführungszeichen getrennt sind, um Fragen zu beantworten. Wenn die Antwort nicht in den Artikeln zu finden ist, schreiben Sie "Ich konnte keine Antwort finden".

BENUTZER

<Fügen Sie Artikel ein, die jeweils durch dreifache Anführungszeichen getrennt sind>

Frage: <Frage hier einfügen>

[Im Playground öffnen](#)

Da alle Modelle nur begrenzte Kontextfenster haben, benötigen wir eine Möglichkeit, Informationen, die für die gestellte Frage relevant sind, dynamisch nachzuschlagen. [Einbettungen](#) können verwendet werden, um eine effiziente Wissensbeschaffung zu implementieren. Weitere Informationen zur Implementierung finden Sie in der Taktik "[Verwenden der eingebetteten Suche zum Implementieren einer effizienten Wissensabrufung](#)".

Taktik: Weisen Sie das Modell an, mit Zitaten aus einem Referenztext zu antworten

Wenn der Input mit relevantem Wissen ergänzt wurde, ist es einfach, das Modell aufzufordern, seinen Antworten Zitate hinzuzufügen, indem es auf Passagen aus bereitgestellten Dokumenten verweist. Beachten Sie, dass Zitate in der Ausgabe dann programmgesteuert durch Zeichenfolgenabgleich innerhalb der bereitgestellten Dokumente überprüft werden können.

SYSTEM

Sie erhalten ein Dokument, das durch dreifache Anführungszeichen und eine Frage getrennt ist. Ihre Aufgabe ist es, die Frage nur anhand des bereitgestellten Dokuments zu beantworten und die Passage(n) des Dokuments zu zitieren, die zur Beantwortung der Frage verwendet wurde. Wenn das Dokument nicht die Informationen enthält, die zur Beantwortung dieser Frage erforderlich sind, schreiben Sie einfach: "Unzureichende Informationen". Wenn eine Antwort auf die Frage gegeben wird, muss diese mit einem Zitat versehen werden. Verwenden Sie das folgende Format, um relevante Passagen zu zitieren ({"Zitat": ...}).

BENUTZER

""<Dokument hier einfügen>"" Frage: <Frage hier einfügen>

[Im Playground öffnen](#)

Strategie: Komplexe Aufgaben in einfachere Teilaufgaben aufteilen

Taktik: Verwenden Sie die Absichtsklassifizierung, um die relevantesten Anweisungen für eine Benutzerabfrage zu identifizieren

Bei Aufgaben, bei denen viele unabhängige Befehlssätze benötigt werden, um verschiedene Fälle zu behandeln, kann es von Vorteil sein, zunächst den Abfragetyp zu klassifizieren und diese Klassifizierung zu verwenden, um zu bestimmen, welche Anweisungen benötigt werden. Dies kann erreicht werden, indem feste Kategorien definiert und Anweisungen fest codiert werden, die für die Bearbeitung von Aufgaben in einer bestimmten Kategorie relevant sind. Dieser Prozess kann auch rekursiv angewendet werden, um eine Aufgabe in eine Abfolge von Phasen zu zerlegen. Der Vorteil dieses Ansatzes besteht darin, dass jede Abfrage nur die Anweisungen enthält, die zum Ausführen der nächsten Phase einer Aufgabe erforderlich sind, was zu niedrigeren Fehlerraten führen kann, als wenn eine einzelne Abfrage zum Ausführen der gesamten Aufgabe verwendet wird. Dies kann auch zu geringeren Kosten führen, da die Ausführung größerer Eingabeaufforderungen mehr kostet ([siehe Preisinformationen](#)).

Nehmen wir z. B. an, dass Abfragen für eine Kundenserviceanwendung wie folgt klassifiziert werden könnten:

SYSTEM

Sie werden mit Kundendienstanfragen versorgt. Klassifizieren Sie jede Abfrage in eine primäre und eine sekundäre Kategorie. Stellen Sie Ihre Ausgabe im JSON-Format mit den Schlüsseln primär und sekundär bereit. Primäre Kategorien: Abrechnung, Technischer Support, Kontoverwaltung oder Allgemeine Anfrage. Sekundäre Kategorien für die Abrechnung: - Abmeldung oder Upgrade - Fügen Sie eine Zahlungsmethode hinzu - Erklärung für die Ladung - Anfechtung einer Abbuchung Sekundäre Kategorien des technischen Supports: -Fehlerbehebung - Geräte-Kompatibilität - Software-Updates Sekundäre Kategorien für die Kontoverwaltung: - Zurücksetzen des Passworts - Persönliche Informationen aktualisieren - Konto schließen - Sicherheit des Kontos Sekundäre Kategorien für allgemeine Anfragen: -Produktinformation -Auszeichnung - Feedback - Sprich mit einem Menschen

BENUTZER

Ich muss mein Internet wieder zum Laufen bringen.

[Im Playground öffnen](#)

Basierend auf der Klassifizierung der Kundenabfrage kann einem Modell eine Reihe spezifischerer Anweisungen zur Verfügung gestellt werden, damit es die nächsten Schritte ausführen kann. Angenommen, der Kunde benötigt Hilfe bei der "Fehlerbehebung".

SYSTEM

Sie erhalten Kundendienstanfragen, die eine Fehlerbehebung im Kontext des technischen Supports erfordern. Helfen Sie dem Benutzer, indem Sie: - Bitten Sie sie, zu überprüfen, ob alle Kabel zum/vom Router angeschlossen sind. Beachten Sie, dass es üblich ist, dass sich Kabel im Laufe der Zeit lösen. - Wenn alle Kabel angeschlossen sind und das Problem weiterhin besteht, fragen Sie sie, welches Routermodell sie verwenden - Jetzt werden Sie ihnen sagen, wie sie ihr Gerät neu starten können: -- Wenn die Modellnummer MTD-327J lautet, weisen Sie sie an, den roten Knopf zu drücken und ihn 5 Sekunden lang gedrückt zu halten, und warten Sie dann 5 Minuten, bevor Sie die Verbindung testen. -- Wenn die Modellnummer MTD-327S lautet, weisen Sie sie an, den Netzstecker zu ziehen und wieder einzustecken, und warten Sie dann 5 Minuten, bevor

Sie die Verbindung testen. - Wenn das Problem des Kunden nach dem Neustart des Geräts und einer Wartezeit von 5 Minuten weiterhin besteht, verbinden Sie ihn mit dem IT-Support, indem Sie {"IT-Support angefordert"} eingeben. - Wenn der Benutzer anfängt, Fragen zu stellen, die nichts mit diesem Thema zu tun haben, bestätigen Sie, ob er den aktuellen Chat zur Fehlerbehebung beenden möchte, und klassifizieren Sie seine Anfrage nach dem folgenden Schema: <Primär-/Sekundärklassifikationsschema von oben hier einfügen>

BENUTZER

Ich muss mein Internet wieder zum Laufen bringen.

[Im Playground öffnen](#)

Beachten Sie, dass das Modell angewiesen wurde, spezielle Zeichenfolgen auszugeben, um anzugeben, wann sich der Status der Konversation ändert. Dies ermöglicht es uns, unser System in eine Zustandsmaschine zu verwandeln, in der der Staat bestimmt, welche Anweisungen injiziert werden. Indem wir den Zustand verfolgen, welche Anweisungen in diesem Zustand relevant sind, und optional auch, welche Zustandsübergänge von diesem Zustand aus zulässig sind, können wir Leitplanken für die Benutzenerfahrung festlegen, die mit einem weniger strukturierten Ansatz nur schwer zu erreichen wären.

Taktik: Für Dialoganwendungen, die sehr lange Gespräche erfordern, vorherige Dialoge zusammenfassen oder filtern

Da Modelle eine feste Kontextlänge haben, kann der Dialog zwischen einem Benutzer und einem Assistenten, in dem die gesamte Konversation im Kontextfenster enthalten ist, nicht unbegrenzt fortgesetzt werden.

Es gibt verschiedene Problemumgehungen für dieses Problem, von denen eine darin besteht, vorherige Wendungen in der Konversation zusammenzufassen. Sobald die Größe der Eingabe eine vorgegebene Schwellenwertlänge erreicht, kann dies eine Abfrage auslösen, die einen Teil der Konversation zusammenfasst, und die Zusammenfassung der vorherigen Konversation kann als Teil der Systemmeldung enthalten sein. Alternativ könnte die vorherige Konversation während der gesamten Konversation asynchron im Hintergrund zusammengefasst werden.

Eine alternative Lösung besteht darin, die vorherigen Teile der Konversation, die für die aktuelle Abfrage am relevantesten sind, dynamisch auszuwählen. Weitere Informationen finden Sie in der Taktik "[Verwenden Sie die einbettungsbasierte Suche, um eine effiziente Wissensbeschaffung zu implementieren](#)".

Taktik: Fassen Sie lange Dokumente stückweise zusammen und konstruieren Sie rekursiv eine vollständige Zusammenfassung

Da Modelle eine feste Kontextlänge haben, können sie nicht verwendet werden, um einen Text zusammenzufassen, der länger als die Kontextlänge abzüglich der Länge der generierten Zusammenfassung in einer einzelnen Abfrage ist.

Um ein sehr langes Dokument, z. B. ein Buch, zusammenzufassen, können wir eine Sequenz von Abfragen verwenden, um jeden Abschnitt des Dokuments zusammenzufassen. Abschnittszusammenfassungen können verkettet und zusammengefasst werden, um Zusammenfassungen von Zusammenfassungen zu erstellen. Dieser Prozess kann rekursiv ablaufen, bis ein ganzes Dokument zusammengefasst ist. Wenn es notwendig ist, Informationen über frühere Abschnitte zu verwenden, um spätere Abschnitte zu verstehen, dann ist ein weiterer Trick, der nützlich

sein kann, eine fortlaufende Zusammenfassung des Textes einzufügen, der einem bestimmten Punkt im Buch vorangestellt ist, während der Inhalt an dieser Stelle zusammengefasst wird. Die Wirksamkeit dieses Verfahrens zum Zusammenfassen von Büchern wurde in früheren [Forschungen](#) von OpenAI anhand von Varianten von GPT-3 untersucht.

Strategie: Den Modellen Zeit zum "Nachdenken" geben

Taktik: Weisen Sie das Modell an, seine eigene Lösung zu erarbeiten, bevor Sie voreilig zu einem Schluss kommen

Manchmal erzielen wir bessere Ergebnisse, wenn wir das Modell explizit anweisen, von den Grundprinzipien auszugehen, bevor wir zu einer Schlussfolgerung kommen. Nehmen wir zum Beispiel an, wir möchten ein Modell, um die Lösung eines Schülers für ein mathematisches Problem zu bewerten. Der naheliegendste Weg, dies anzugehen, besteht darin, das Modell einfach zu fragen, ob die Lösung des Schülers richtig ist oder nicht.

SYSTEM

Stellen Sie fest, ob die Lösung des Schülers richtig ist oder nicht.

BENUTZER

Problemstellung: Ich baue eine Solaranlage und brauche Hilfe bei der Ausarbeitung der Finanzen. - Grundstück kostet 100 \$ / Quadratfuß - Ich kann Sonnenkollektoren für 250 \$ / Quadratfuß kaufen - Ich habe einen Wartungsvertrag ausgehandelt, der mich pauschal 100.000 US-Dollar pro Jahr und zusätzliche 10 US-Dollar pro Quadratmeter kosten wird Wie hoch sind die Gesamtkosten für das erste Betriebsjahr in Abhängigkeit von der Anzahl der Quadratmeter? Lösung des Schülers: Sei x die Größe der Installation in Quadratmetern. 1. Grundstückskosten: $100x$ 2. Kosten für Solarmodule: $250x$ 3. Wartungskosten: $100.000 + 100x$ Gesamtkosten: $100x + 250x + 100.000 + 100x = 450x + 100.000$

ASSISTENT

Die Lösung des Schülers ist richtig.

[Im Playground öffnen](#)

Aber die Lösung des Schülers ist eigentlich nicht richtig! Wir können das Modell dazu bringen, dies erfolgreich zu bemerken, indem wir das Modell auffordern, zuerst eine eigene Lösung zu generieren.

SYSTEM

Erarbeiten Sie zunächst Ihre eigene Lösung für das Problem. Vergleichen Sie dann Ihre Lösung mit der Lösung des Schülers und bewerten Sie, ob die Lösung des Schülers richtig ist oder nicht. Entscheide erst, ob die Lösung des Schülers richtig ist, wenn du die Aufgabe selbst gelöst hast.

BENUTZER

Problemstellung: Ich baue eine Solaranlage und brauche Hilfe bei der Ausarbeitung der Finanzen. - Grundstück kostet 100 \$ / Quadratfuß - Ich kann Sonnenkollektoren für 250 \$ / Quadratfuß kaufen - Ich habe einen Wartungsvertrag ausgehandelt, der mich pauschal 100.000 US-Dollar pro Jahr und zusätzliche 10 US-Dollar pro Quadratmeter kosten wird Wie hoch sind die Gesamtkosten für das erste Betriebsjahr in Abhängigkeit von der Anzahl der Quadratmeter? Lösung des Schülers: Sei x die Größe der Installation in Quadratmetern. 1. Grundstückskosten: $100x$ 2. Kosten für Solarmodule: $250x$ 3. Wartungskosten: $100.000 + 100x$ Gesamtkosten: $100x + 250x + 100.000 + 100x = 450x + 100.000$

ASSISTENT

Sei x die Größe der Anlage in Quadratfuß. 1. Grundstückskosten: $100x$ 2. Kosten für Solarmodule: $250x$ 3. Wartungskosten: $100.000 + 10x$ Gesamtkosten: $100x + 250x + 100.000 + 10x = 360x + 100.000$ Die Lösung des Schülers ist falsch. Sie haben einen Fehler bei der Berechnung der Wartungskosten gemacht, indem sie $100x$ anstelle von $10x$ verwendet haben. Die korrekten Gesamtkosten für das erste Betriebsjahr betragen $360x + 100.000$.

[Im Playground öffnen](#)

Taktik: Verwenden Sie einen inneren Monolog oder eine Abfolge von Abfragen, um den Argumentationsprozess des Modells zu verbergen

Die vorherige Taktik zeigt, dass es manchmal wichtig ist, dass das Modell ein Problem detailliert betrachtet, bevor es eine bestimmte Frage beantwortet. Bei einigen Anwendungen wäre es ungeeignet, den Argumentationsprozess, den ein Modell verwendet, um zu einer endgültigen Antwort zu gelangen, mit dem Benutzer zu teilen. Zum Beispiel möchten wir in Nachhilfeanwendungen die Schüler vielleicht ermutigen, ihre eigenen Antworten zu erarbeiten, aber der Denkprozess eines Modells über die Lösung des Schülers könnte dem Schüler die Antwort offenbaren.

Der innere Monolog ist eine Taktik, mit der dies abgemildert werden kann. Die Idee des inneren Monologs besteht darin, das Modell anzuweisen, Teile der Ausgabe, die vor dem Benutzer verborgen werden sollen, in ein strukturiertes Format zu bringen, das das Parsen erleichtert. Bevor die Ausgabe dem Benutzer präsentiert wird, wird die Ausgabe analysiert und nur ein Teil der Ausgabe sichtbar gemacht.

SYSTEM

Führen Sie die folgenden Schritte aus, um die Benutzerfragen zu beantworten. Schritt 1 - Erarbeiten Sie zunächst Ihre eigene Lösung für das Problem. Verlassen Sie sich nicht auf die Lösung des Schülers, da diese möglicherweise falsch ist. Schließen Sie Ihre gesamte Arbeit für diesen Schritt in dreifache Anführungszeichen (""") ein. Schritt 2 - Vergleichen Sie Ihre Lösung mit der Lösung des Schülers und bewerten Sie, ob die Lösung des Schülers richtig ist oder nicht. Schließen Sie Ihre gesamte Arbeit für diesen Schritt in dreifache Anführungszeichen (""") ein. Schritt 3 - Wenn der Schüler einen Fehler gemacht hat, legen Sie fest, welchen Hinweis Sie dem Schüler geben können, ohne die Antwort zu verraten. Schließen Sie Ihre gesamte Arbeit für diesen Schritt in dreifache Anführungszeichen (""") ein. Schritt 4 - Wenn der Schüler einen Fehler gemacht hat, geben Sie ihm den Hinweis aus dem vorherigen Schritt (ohne dreifache Anführungszeichen). Anstatt zu schreiben "Schritt 4 - ..." Schreiben Sie "Hinweis:".

BENUTZER

Problemstellung: <Problemstellung einfügen> Schülerlösung: <Schülerlösung einfügen>

[Im Playground öffnen](#)

Alternativ kann dies mit einer Sequenz von Abfragen erreicht werden, bei der alle außer der letzten ihre Ausgabe für den Endbenutzer verbergen.

Zuerst können wir das Modell bitten, das Problem selbst zu lösen. Da für diese anfängliche Abfrage keine Lösung des Schülers erforderlich ist, kann sie weggelassen werden. Dies bietet den zusätzlichen Vorteil, dass es keine Chance gibt, dass die Lösung des Modells durch den Lösungsversuch des Schülers verzerrt wird.

BENUTZER

<Problemstellung einfügen>

[Open in Playground](#)

Als nächstes können wir das Modell alle verfügbaren Informationen verwenden lassen, um die Richtigkeit der Lösung des Schülers zu bewerten.

SYSTEM

Vergleichen Sie Ihre Lösung mit der Lösung des Schülers und bewerten Sie, ob die Lösung des Schülers richtig ist oder nicht.

BENUTZER

Problemstellung: ""<Problemstellung einfügen>"" Ihre Lösung: ""<modellgenerierte Lösung einfügen>"" Lösung des Schülers: ""<Lösung des Schülers einfügen>""

[Im Playground öffnen](#)

Schließlich können wir das Modell seine eigene Analyse verwenden lassen, um eine Antwort in der Persona eines hilfreichen Tutors zu konstruieren.

SYSTEM

Sie sind Mathe-Nachhilfelehrer. Wenn der Schüler einen Fehler gemacht hat, geben Sie ihm einen Hinweis auf eine Weise, die die Antwort nicht preisgibt. Wenn der Schüler keinen Fehler gemacht hat, bieten Sie ihm einfach einen ermutigenden Kommentar an.

BENUTZER

Problemstellung: ""<Problemstellung einfügen>"" Ihre Lösung: ""<modellgenerierte Lösung einfügen>"" Lösung des Schülers: ""<Lösung des Schülers einfügen>""

Analyse: ""<modellgenerierte Analyse aus dem vorherigen Schritt einfügen>""

[Im Playground öffnen](#)

Taktik: Fragen Sie das Modell, ob es bei früheren Durchgängen etwas übersehen hat

Angenommen, wir verwenden ein Modell, um Auszüge aus einer Quelle aufzulisten, die für eine bestimmte Fragestellung relevant sind. Nach dem Auflisten jedes Auszugs muss das Modell entscheiden, ob es mit dem Schreiben eines weiteren Auszugs beginnen oder aufhören soll. Wenn das Quelldokument groß ist, kommt es häufig vor, dass ein Modell zu früh stoppt und nicht alle relevanten Auszüge auflistet. In diesem Fall kann häufig eine bessere Leistung erzielt werden, indem das Modell mit Folgeabfragen aufgefordert wird, alle Ausschnitte zu finden, die in früheren Durchläufen übersehen wurden.

SYSTEM

Sie erhalten ein Dokument, das durch dreifache Anführungszeichen getrennt ist. Ihre Aufgabe ist es, Auszüge auszuwählen, die sich auf folgende Frage beziehen: "Welche bedeutenden Paradigmenwechsel haben sich in der Geschichte der Künstlichen Intelligenz vollzogen?" Stellen Sie sicher, dass die Auszüge den gesamten relevanten Kontext enthalten, der für ihre Interpretation erforderlich ist - mit anderen Worten, extrahieren Sie keine kleinen Schnipsel, denen wichtiger Kontext fehlt. Geben Sie die Ausgabe im JSON-Format wie folgt an: [{"excerpt": "..."}, ... {"excerpt": "..."}]

BENUTZER

""<Dokument hier einfügen>""

ASSISTENT

```
[{"excerpt": "Das Model schreibt hier einen Auszug"}, ... {"excerpt": "Das Model schreibt hier einen weiteren Auszug"}]
```

BENUTZER

Gibt es relevantere Auszüge? Achten Sie darauf, Auszüge nicht zu wiederholen. Stellen Sie außerdem sicher, dass die Auszüge den gesamten relevanten Kontext enthalten, der für ihre Interpretation erforderlich ist - mit anderen Worten, extrahieren Sie keine kleinen Schnipsel, denen wichtiger Kontext fehlt.

[Im Playground öffnen](#)

Strategie: Externe Tools nutzen

Taktik: Verwenden Sie die auf Einbettungen basierende Suche, um eine effiziente Wissensbeschaffung zu implementieren

Ein Modell kann externe Informationsquellen nutzen, wenn es als Teil seines Inputs bereitgestellt wird. Dies kann dem Modell helfen, fundiertere und aktuellere Antworten zu generieren. Wenn ein Benutzer beispielsweise eine Frage zu einem bestimmten Film stellt, kann es nützlich sein, qualitativ hochwertige Informationen über den Film (z. B. Schauspieler, Regisseur usw.) zur Eingabe des Modells hinzuzufügen. Einbettungen können verwendet werden, um eine effiziente Wissensabfrage zu implementieren, sodass relevante Informationen zur Laufzeit dynamisch zur Modelleingabe hinzugefügt werden können.

Eine Texteinbettung ist ein Vektor, der die Verwandtschaft zwischen Textzeichenfolgen messen kann. Ähnliche oder relevante Zeichenfolgen liegen näher beieinander als nicht verwandte Zeichenfolgen. Diese Tatsache, zusammen mit dem Vorhandensein schneller Vektorsuchalgorithmen, bedeutet, dass Einbettungen verwendet werden können, um eine effiziente Wissensbeschaffung zu implementieren. Insbesondere kann ein Textkorpus in Blöcke aufgeteilt werden, und jeder Block kann eingebettet und gespeichert werden. Dann kann eine bestimmte Abfrage eingebettet werden und eine Vektorsuche durchgeführt werden, um die eingebetteten Textblöcke aus dem Korpus zu finden, die am meisten mit der Abfrage verwandt sind (d. h. im Einbettungsraum am nächsten beieinander liegen).

Beispielimplementierungen finden Sie im [OpenAI Cookbook](#). In der Taktik "[Weisen Sie das Modell an, abgerufenes Wissen zum Beantworten von Abfragen zu verwenden](#)" finden Sie ein Beispiel dafür, wie Sie mithilfe des Wissensabrufs die Wahrscheinlichkeit minimieren können, dass ein Modell falsche Fakten erfindet.

Taktik: Verwenden Sie die Codeausführung, um genauere Berechnungen durchzuführen oder externe APIs aufzurufen.

Sprachmodelle können sich nicht darauf verlassen, dass sie allein arithmetische oder lange Berechnungen genau durchführen. In Fällen, in denen dies erforderlich ist, kann ein Modell angewiesen werden, Code zu schreiben und auszuführen, anstatt eigene Berechnungen durchzuführen. Insbesondere kann ein Modell angewiesen werden, Code, der ausgeführt werden soll, in ein bestimmtes Format wie z. B. einen dreifachen Backtick einzufügen. Nachdem eine Ausgabe erstellt wurde, kann der Code extrahiert und ausgeführt werden. Schließlich kann bei Bedarf die Ausgabe der Codeausführungs-Engine (d. h. des Python-Interpreters) als Eingabe für das Modell für die nächste Abfrage bereitgestellt werden.

SYSTEM

Sie können Python-Code schreiben und ausführen, indem Sie ihn in dreifache Backticks einschließen, z. B. `code goes here`. Verwenden Sie diese Option, um Berechnungen durchzuführen.

BENUTZER

Finde alle reellwertigen Wurzeln des folgenden Polynoms: $3x^5 - 5x^4 - 3x^3 - 7x - 10$.

[Im Playground öffnen](#)

Ein weiterer guter Anwendungsfall für die Codeausführung ist der Aufruf externer APIs. Wenn ein Modell in die ordnungsgemäße Verwendung einer API angewiesen wird, kann es Code schreiben, der diese verwendet. Ein Modell kann in die Verwendung einer API

eingewiesen werden, indem es mit Dokumentation und/oder Codebeispielen versehen wird, die zeigen, wie die API verwendet wird.

SYSTEM

Sie können Python-Code schreiben und ausführen, indem Sie ihn in dreifache Backticks einschließen. Beachten Sie auch, dass Sie Zugriff auf das folgende Modul haben, mit dem Benutzer Nachrichten an ihre Freunde senden können: `python Nachricht importieren message.write(to="John", message="Hey, willst du dich nach der Arbeit treffen?") ```

[Im Playground öffnen](#)

WARNUNG: Das Ausführen von Code, der von einem Modell erzeugt wird, ist nicht von Natur aus sicher, und in jeder Anwendung, die dies anstrebt, sollten Vorsichtsmaßnahmen getroffen werden. Insbesondere ist eine Sandkasten-Codeausführungsumgebung erforderlich, um den Schaden zu begrenzen, den nicht vertrauenswürdiger Code verursachen könnte.

Taktik: Gewähren Sie dem Modell Zugriff auf bestimmte Funktionen

Die Chat-Abschluss-API ermöglicht das Übergeben einer Liste von Funktionsbeschreibungen in Anfragen. Auf diese Weise können Modelle Funktionsargumente gemäß den bereitgestellten Schemas generieren. Generierte Funktionsargumente werden von der API im JSON-Format zurückgegeben und können zum Ausführen von Funktionsaufrufen verwendet werden. Die von Funktionsaufrufen bereitgestellte Ausgabe kann dann in der folgenden Anforderung in ein Modell zurückgeführt werden, um die Schleife zu schließen. Dies ist die empfohlene Methode, OpenAI-Modelle zum Aufrufen externer Funktionen zu verwenden. Weitere Informationen finden Sie im [Abschnitt zum Funktionsaufruf](#) in unserem einführenden Leitfaden zur Textgenerierung und weitere [Beispiele für Funktionsaufrufe](#) im OpenAI Cookbook.

Strategie: Änderungen systematisch testen

Manchmal kann es schwierig sein, zu sagen, ob eine Änderung – z. B. eine neue Anweisung oder ein neues Design – Ihr System besser oder schlechter macht. Ein Blick auf einige Beispiele kann darauf hindeuten, was besser ist, aber bei kleinen Stichprobengrößen kann es schwierig sein, zwischen einer echten Verbesserung oder zufälligem Glück zu unterscheiden. Möglicherweise verbessert die Änderung die Leistung bei einigen Eingaben, beeinträchtigt jedoch die Leistung bei anderen.

Evaluierungsverfahren (oder "Evals") sind nützlich, um Systemdesigns zu optimieren. Gute Evaluierungen sind:

- Repräsentativ für die reale Nutzung (oder zumindest vielfältig)
- Enthalten viele Testfälle für eine größere statistische Aussagekraft (Richtlinien finden Sie in der Tabelle unten)
- Einfach zu automatisieren oder zu wiederholen

ZU ERKENNENDER UNTERSCHIED

STICHPROBENUMFANG FÜR 95 % KONFIDENZ ERFORDERLICH

30%	~10 KARTON
10%	~100
3%	~1.000
1%	~10.000

Die Auswertung der Ausgaben kann von Computern, Menschen oder einer Mischung aus anderen Quellen erfolgen. Computer können Evaluierungen mit objektiven Kriterien (z. B. Fragen mit einzelnen richtigen Antworten) sowie einigen subjektiven oder Fuzzy-Kriterien automatisieren, bei denen Modellausgaben von anderen Modellabfragen ausgewertet werden. [OpenAI Evals](#) ist ein Open-Source-Software-Framework, das Tools zum Erstellen automatisierter Evaluierungen bereitstellt.

Modellbasierte Evaluierungen können nützlich sein, wenn es eine Reihe möglicher Ausgaben gibt, die als gleich hochwertig angesehen werden (z. B. für Fragen mit langen Antworten). Die Grenze zwischen dem, was mit einer modellbasierten Evaluierung realistisch bewertet werden kann, und dem, was ein Mensch bewerten muss, ist verschwommen und verschiebt sich ständig, da die Modelle immer leistungsfähiger werden. Wir ermutigen zum Experimentieren, um herauszufinden, wie gut modellbasierte Evaluierungen für Ihren Anwendungsfall funktionieren können.

Taktik: Evaluieren von Modellausgaben unter Bezugnahme auf Goldstandard-Antworten

Angenommen, es ist bekannt, dass sich die richtige Antwort auf eine Frage auf einen bestimmten Satz bekannter Fakten beziehen sollte. Dann können wir eine Modellabfrage verwenden, um zu zählen, wie viele der erforderlichen Fakten in der Antwort enthalten sind.

Verwenden Sie z. B. die folgende Systemmeldung:

SYSTEM

Sie erhalten einen Text, der durch dreifache Anführungszeichen begrenzt ist und die Antwort auf eine Frage sein soll. Prüfen Sie, ob die folgenden Informationen direkt in der Antwort enthalten sind: - Neil Armstrong war der erste Mensch, der den Mond betrat. - Das Datum, an dem Neil Armstrong zum ersten Mal den Mond betrat, war der 21. Juli 1969. Führen Sie für jeden dieser Punkte die folgenden Schritte aus: 1 - Wiederholen Sie den Punkt. 2 - Geben Sie ein Zitat aus der Antwort an, das diesem Punkt am nächsten kommt. 3 - Überlegen Sie, ob jemand, der das Zitat liest und das Thema nicht kennt, den Punkt direkt ableiten könnte. Erkläre, warum oder warum nicht, bevor du dich entscheidest. 4 - Schreiben Sie "ja", wenn die Antwort auf 3 ja war, andernfalls schreiben

Sie "nein". Geben Sie abschließend an, wie viele "Ja"-Antworten es gibt. Geben Sie diese Anzahl als {"count": <insert count here>} an.

[Im Playground öffnen](#)

Hier ist eine Beispieleingabe, bei der beide Punkte erfüllt sind:

SYSTEM

<oben Systemmeldung einfügen>

USER

""Neil Armstrong is famous for being the first human to set foot on the Moon. This historic event took place on July 21, 1969, during the Apollo 11 mission.""

[Open in Playground](#)

Here's an example input where only one point is satisfied:

SYSTEM

<oben Systemmeldung einfügen>

BENUTZER

"Neil Armstrong schrieb Geschichte, als er die Mondlandefähre verließ und der erste Mensch wurde, der den Mond betrat."

[Im Playground öffnen](#)

Hier ist eine Beispieleingabe, bei der keine erfüllt ist:

SYSTEM

<oben Systemmeldung einfügen>

BENUTZER

""Im Sommer '69 war eine Reise groß, Apollo 11, kühn wie die Hand der Legende. Armstrong machte einen Schritt, die Geschichte entfaltete sich, »Ein kleiner Schritt«, sagte er, »für eine neue Welt.«

[Im Playground öffnen](#)

Es gibt viele mögliche Varianten dieser Art der modellbasierten Auswertung. Betrachten Sie die folgende Variante, die die Art der Überlappung zwischen der Kandidatenantwort und der Goldstandardantwort verfolgt und auch verfolgt, ob die Kandidatenantwort einem Teil der Goldstandardantwort widerspricht.

SYSTEM

Führen Sie die folgenden Schritte aus, um auf Benutzereingaben zu reagieren.

Wiederholen Sie jeden Schritt vollständig, bevor Sie fortfahren. z.B. "Schritt 1:

Begründung...". Schritt 1: Überlegen Sie Schritt für Schritt, ob die Informationen in der eingereichten Antwort im Vergleich zur Expertenantwort entweder: disjunkt, gleich, eine Teilmenge, eine Obermenge oder überlappend (d. h. eine Schnittmenge, aber keine Teilmenge/Obermenge) sind. Schritt 2: Argumentieren Sie Schritt für Schritt, ob die eingereichte Antwort einem Aspekt der Expertenantwort widerspricht. Schritt 3: Geben Sie ein JSON-Objekt aus, das wie folgt strukturiert ist: {"type_of_overlap": "disjunkt" oder "gleich" oder "Teilmenge" oder "Obermenge" oder "überlappend", "Widerspruch": wahr oder falsch}

[Im Playground öffnen](#)

Hier ist eine Beispieleingabe mit einer minderwertigen Antwort, die jedoch nicht im Widerspruch zur Expertenantwort steht:

SYSTEM

<oben Systemmeldung einfügen>

BENUTZER

Frage: ""Welches Ereignis ist Neil Armstrong am berühmtesten und an welchem Datum ereignete es sich? Nehmen Sie die UTC-Zeit an."" Eingereichte Antwort: ""Ist er nicht auf dem Mond spazieren gegangen oder so?"" Expertenantwort: ""Neil Armstrong ist vor

allem dafür bekannt, dass er der erste Mensch war, der den Mond betrat. Dieses historische Ereignis ereignete sich am 21. Juli 1969.

[Im Playground öffnen](#)

Hier ist eine Beispieleingabe mit Antwort, die der Expertenantwort direkt widerspricht:

SYSTEM

<oben Systemmeldung einfügen>

BENUTZER

Frage: ""Welches Ereignis ist Neil Armstrong am berühmtesten und an welchem Datum ereignete es sich? Nehmen Sie die UTC-Zeit an."" Am 21. Juli 1969 war Neil Armstrong nach Buzz Aldrin der zweite Mensch, der den Mond betrat." Expertenantwort: ""Neil Armstrong ist vor allem dafür bekannt, dass er der erste Mensch war, der den Mond betrat. Dieses historische Ereignis ereignete sich am 21. Juli 1969.

[Im Playground öffnen](#)

Hier ist eine Beispieleingabe mit einer richtigen Antwort, die auch etwas mehr Details als nötig enthält:

SYSTEM

<oben Systemmeldung einfügen>

BENUTZER

Frage: ""Welches Ereignis ist Neil Armstrong am berühmtesten und an welchem Datum ereignete es sich? Nehmen Sie die UTC-Zeit an."" Am 21. Juli 1969 um ca. 02:56 UTC betrat Neil Armstrong als erster Mensch die Mondoberfläche und stellte damit eine monumentale Leistung in der Geschichte der Menschheit dar." Expertenantwort: ""Neil Armstrong ist vor allem dafür bekannt, dass er der erste Mensch war, der den Mond betrat. Dieses historische Ereignis ereignete sich am 21. Juli 1969.

[Im Playground öffnen](#)

Weitere Ressourcen

Weitere Inspirationen finden Sie im [OpenAI Cookbook](#), das Beispielcode und Links zu Ressourcen von Drittanbietern enthält, wie z. B.:

- [Eingabeaufforderung für Bibliotheken und Tools](#)
- [Leitfäden für Eingabeaufforderungen](#)
- [Video-Kurse](#)
- [Veröffentlichungen über fortgeschrittenes Prompting zur Verbesserung des logischen Denkens](#)