

# Ask Smart to Get Smart: Mathematische Ausgaben generativer KI-Sprachmodelle verbessern durch gezieltes Prompt Engineering

Sebastian Schorcht, Lukas Baumanns, Nils Buchholtz, Judith Huget, Franziska Peters und Maximilian Pohl

*Der Beitrag beschäftigt sich mit den derzeitigen mathematischen Fähigkeiten des generativen KI-Sprachmodells ChatGPT in Bezug auf das Lösen mathematischer Probleme. Um für den schulischen Nutzen des Modells die Zuverlässigkeit und Nachvollziehbarkeit von mathematischen Ausgaben zu optimieren, stellen wir im Beitrag unterschiedliche Prompt-Techniken vor und untersuchen ihren Effekt in Modellvalidierungen anhand einer arithmetischen und einer algebraischen Problemlöseaufgabe.*

## Einleitung

Obwohl generative KI-Sprachmodelle wie ChatGPT (erreichbar unter der URL [chat.openai.com](https://chat.openai.com)) bereits unterschiedliche und auch komplexe mathematische Aufgaben korrekt bearbeiten können, treten auch bei zum Teil einfachen Aufgaben derzeit noch Schwierigkeiten in Bezug auf eine verlässliche korrekte Lösung auf. Die jeweiligen Outputs sollten stets auf ihre Richtigkeit hin überprüft werden (Buchholtz et al., 2023). Hier hat sich jedoch in den letzten Monaten einiges getan: Mittlerweile wurde das ChatGPT zugrunde liegende Sprachmodell von GPT-3.5 auf die neueste Version GPT-4 verbessert, was das Problem allerdings noch nicht gelöst hat (Schönthaler, 2023). Im Gegensatz zu der Vorgängerversion GPT-3.5 erzielt GPT-4 aber deutlich bessere Ergebnisse im mathematischen Bereich ([openai.com/research/gpt-4](https://openai.com/research/gpt-4))<sup>1</sup>. In diesem Beitrag wollen wir klären, wie es aktuell um die mathematischen Fähigkeiten von ChatGPT steht, um anschließend mit der neu entwickelten Methode des Chain-of-thought-Prompting (Liu et al., 2021; Wei et al., 2023) einen Weg aufzuzeigen, die mathematischen Argumentationsfähigkeiten durch die gezielte Modellierung von Eingaben (auch Prompt Engineering genannt) zu verbessern und die Outputs von ChatGPT so transparenter zu gestalten.

## Mathematische Fähigkeiten der aktuellen Version von ChatGPT

Am 14. März 2023 veröffentlichte das US-amerikanische Unternehmen Open AI mit GPT-4 (Gene-

rative Pretrained Transformer) die vierte Version seines generativen KI-Sprachmodells. Bei GPT-4 handelt es sich um ein auf künstlicher Intelligenz basierendes Large Language Modell, welches über die Schnittstelle ChatGPT Inhalte menschlicher Sprache (und inzwischen auch Bilder) versteht und diese anhand stochastischer Prozesse automatisch vervollständigt (Paaß, 2022). Das Modell GPT-4 wurde wie seine Vorgänger auf einer Basis riesiger Trainingsdaten aufgebaut. Die Analyse dieser Trainingsdaten verfolgt das Ziel der Muster- und Beziehungserkennung, um auf menschliche Eingaben passende (menschliche) Antworten zu generieren. Die Größe der Trainingsdaten des 2020 veröffentlichten Vorgängermodells GPT-3 beläuft sich auf 175 Milliarden Parameter (Floridi & Chiriatti, 2020), zum aktuellen Modell ist jedoch in dieser Hinsicht nichts bekannt.

Zur Ermittlung und zum Vergleich ihrer Fähigkeiten werden generative KI-Sprachmodelle nach Eingabe der Trainingsdaten u. a. verschiedenen, für Menschen entwickelten Tests unterzogen. So wurde GPT-4 von der Firma Open AI (2023) unter anderem mit dem SAT Evidence-Based Reading & Writing und dem SAT Math Test getestet. Bei beiden Tests handelt es sich um vor allem im US-amerikanischen Raum genutzte Prüfungen, um die Studierfähigkeit von Personen einzuschätzen. Im sprachlichen Test erlangte das generative KI-Sprachmodell eine Punktzahl von 710 von 800 möglichen Punkten; im Mathematiktest eine Punktzahl von 700 von 800 Punkten. Im Gegensatz zu den Ergebnissen von GPT-3.5 (SAT Reading & Writing: 670 von 800; SAT Math: 590 von 800) konnte GPT-4 hier besonders im Bereich der Mathematik seine ohnehin schon beachtliche Punktzahl noch einmal verbessern (Open AI, 2023).

Derzeit werden die Möglichkeiten und Herausforderungen von generativen KI-Sprachmodellen intensiv und kontrovers im schulischen Bildungskontext und in der Hochschullehre diskutiert (Buchholtz et al., 2023; Cherian et al., 2022; Dauscher, 2019; Floridi & Chiriatti, 2020; Lample & Charton, 2019; Lutz-Westphal, 2023; Oldenburg, 2023; Stre-

<sup>1</sup> GPT-4 ist aktuell in der Bezahlversion ChatGPT Plus zugänglich oder kostenlos in der neuesten Version der Suchmaschine Microsoft Bing.

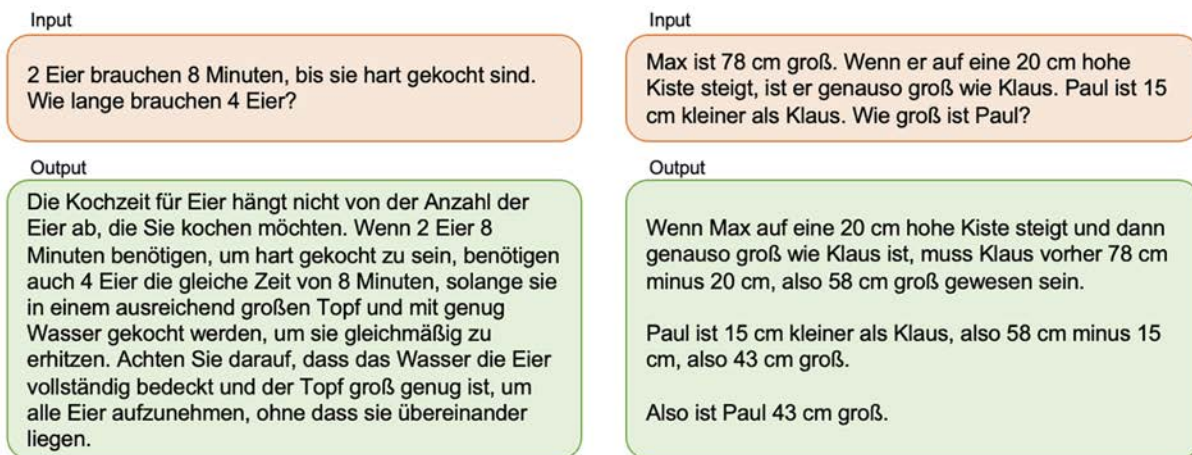


Abbildung 1. Prompts (Input) zur Eier-Aufgabe (links) sowie Körpergrößen-Aufgabe (rechts) und die generierte Antwort (Output) von GPT-4

cker, 2020; Weißels, 2023a). In den geführten Debatten zu den Potenzialen und den Gefahren der Technologie spielt unter anderem das sogenannte KI-Black-Box-Problem eine Rolle, das sich darauf bezieht, dass nach einer Eingabe eines sprachlichen Prompts nicht ersichtlich ist, wie die KI den Input mit Mitteln des maschinellen Lernens verarbeitet und letztlich einen Output generiert (Herm et al., 2021). Dies kann beispielsweise zur Reproduktion unbewusster Vorurteile und anderer (menschlicher) Denkfehler, sog. *Bias* führen, die das KI-Sprachmodell aus seinen Trainingsdaten übernimmt. Als Beispiel rufen wir die Eier-Aufgabe in Erinnerung, bei der GPT-3.5 in unserem Test noch eine durch die Fehlannahme von Proportionalität geprägte Ausgabe ausgab (vgl. Buchholtz et al., 2023). GPT-4 generiert nun im Versuch deutlich bessere Ergebnisse und löst die Aufgabe entsprechend (Abb. 1 links). Letztlich lässt sich durch das KI-Black-Box-Problem aber auch die Verwunderung über falsche Lösungen (oft) einfacher mathematischer Aufgaben ausdrücken. Manche Eingaben kann GPT-4 noch immer nicht einwandfrei und korrekt bearbeiten. Als Beispiel dient eine Aufgabe aus Ott (2018), in der die Körpergröße von Paul bestimmt werden soll (Abb. 1 rechts).

Vorbehalte gegen den Nutzen der Technologie bestehen für das Fach Mathematik trotz der Verbesserung der Modelle also nach wie vor durch die fehlende mathematische Exaktheit der durch die Modelle ausgegebenen Outputs und die fehlende Überprüfbarkeit der Lösungswege der ausgegebenen Antworten – was speziell bei schwierigen mathematischen Problemen relevant wird. Momentan ist es folglich auch mit GPT-4 nur in bestimmten Fällen möglich, die mathematische Antwort auf eine Aufgabe verlässlich zu bestimmen. Umso wichtiger erscheint daher ein produktiver

und kritischer Umgang mit KI-Sprachmodellen im Mathematikunterricht. Diese können zum jetzigen Zeitpunkt z. B. durchaus als Ideengeber verwendet werden (etwa im Rahmen des Problem-Posing), was allerdings nicht der notwendigen mathematischen Basiskompetenzen seitens der Schülerinnen und Schüler entbehrt, um die mathematischen Antworten anschließend auf ihre Korrektheit hin zu überprüfen. Weißels (2023b) schlägt vor, dass KI-Sprachmodelle daher als *Wegbereiter zu einem Endprodukt* verwendet werden können. Dementsprechend sind KI-Sprachmodelle durchaus als Hilfsmittel im Lösungsprozess, etwa beim Bearbeiten von Problemlöseaufgaben, nutzbar. Mit den Worten von Weißels (2023b, S. 5): Die „Maschine recherchiert und bereitet vor, [der] Mensch steuert den Prozess und ‚finalisiert‘ den Text.“ Dazu erscheint es uns insbesondere hilfreich, dem KI-Sprachmodell eine genauere Darlegung seiner mathematischen Lösungswege zu entlocken und so dem KI-Black-Box-Problem entgegenzuwirken.

### Produktiver Nutzen der Technologie durch Techniken des Prompt Engineering

Um die mathematischen Argumentationen von ChatGPT zu verbessern, stellen wir in diesem Beitrag Techniken vor, die dem Ansatz des Prompt Engineering des Natural Language Processing (NLP) entstammen. Dabei wird das *Zero-shot-Learning* (Brown et al., 2020; Kojima et al., 2022) und *Few-shot-Learning* in Kombination mit dem *Chain-of-thought-Prompting* (Liu et al., 2021; Wei et al., 2023) und einer Abwandlung des *Ask-me-Anything-Prompting* (Arora et al., 2023) genutzt.

In vielen Fällen können Prompt-Techniken ohne spezielles Training des generativen KI-Sprachmodells für Inputs verwendet werden und

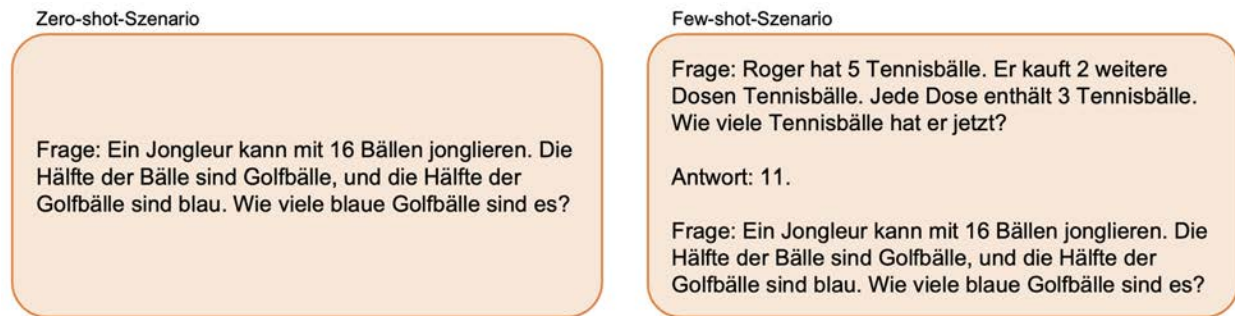


Abbildung 2. Beide Möglichkeiten eines Prompts im Zero-shot-Szenario (links) und Few-shot-Szenario (rechts) nach Liu et al. (2021)

liefern für einfache mathematische Fragen durchaus passable Outputs – wir nehmen an, dies entspricht dem Regelfall einer Eingabe etwa von Schülerinnen und Schülern. Diese Inputs werden als *Zero-shot-Szenario* bezeichnet, da die Prompt-Formulierung keine zusätzlichen Trainingsdaten für die betreffende Aufgabe enthält (Abb. 2). Das generative KI-Sprachmodell gibt dann anhand seiner ursprünglichen allgemeinen Trainingsdaten eine entsprechend wahrscheinliche Antwort. Es gibt allerdings auch Techniken, passgenaue Trainingsdaten in Prompts zu verwenden, um das Modell bereits mit dem Input zu trainieren und den Output zu optimieren. Diese Trainingsdaten können dabei durchaus schon aus einer sehr kleinen Anzahl von Beispielen zur Modellbildung bestehen, was als *Few-shot-Szenario* bezeichnet wird (Abb. 2).

Das generative KI-Sprachmodell nutzt in diesem Fall wie bei einem Worked-out-Example (Renkl, 2002) die eingegebenen Trainingsdaten im Prompt, um eine entsprechende Antwort mit größerer Genauigkeit auszugeben (sog. *Few-shot-Learning*). Ein *n-shot-Prompt* enthält in der Regel *n* Beispiele von Input-Output-Paaren, die als „Shots“ bezeichnet werden. Trainingsbasierte Prompt-Techniken sind also insbesondere nützlich, wenn es nicht genug Trainingsdaten gibt, um einen gewünschten Output auf einen Input mit einer verlässlichen Genauigkeit zu generieren. Daher ist die Verwendung von *Few-shot-Learning* besonders effektiv, um das Modell in die richtige Richtung zu lenken, und erste Studien zeigen tatsächlich eine höhere Genauigkeit bei der Beantwortung mathematischer Fragen (Drori et al., 2022; Liu et al., 2021). Große KI-Sprachmodelle wie GPT-4 bieten die Möglichkeit des kontextbezogenen Few-Shot-Learning innerhalb der Anwendung wie ChatGPT meist schon an. Anstatt für jede neue Eingabe das generative KI-Sprachmodell neu abzustimmen, kann das Modell einfach mit Shots „angewiesen“ werden (Brown et al., 2020). Der erste Shot des generativen KI-Sprachmodells beeinflusst somit die weiteren Ausgaben. Dies hat zum Vorteil, dass Anweisungen nicht jedes Mal neu formuliert

werden müssen, sondern diese vom generativen KI-Sprachmodell bei weiteren Ausgaben berücksichtigt werden. Beispielsweise kann eine erste Eingabe lauten: „Addiere mir die folgenden Zahlen: 20, 45, 234“. Das generative KI-Sprachmodell führt dies aus und gibt z. B. „Die Summe der Zahlen 20, 45 und 234 beträgt 299.“ aus. Im Folgenden können beliebige Zahlen eingegeben werden („36, 5078, 345, 21“), der Output ist, beeinflusst durch den ersten Shot, immer eine Summe dieser Zahlen (Output z. B. „Die Summe der Zahlen 36, 5078, 345 und 21 beträgt 5480.“).

Zusätzlich zu dieser Möglichkeit des Trainings können große KI-Sprachmodelle alternativ auch über den Input von sogenannten Follow-Up-Prompts zum Bilden einer Gedankenkette und damit zur strukturierten Wiedergabe eines Outputs geführt werden (eine Sammlung findet sich etwa unter [chatgpttrainer.co.uk/chatgpt-follow-up-prompts/](https://chatgpttrainer.co.uk/chatgpt-follow-up-prompts/)). Diese Form des Prompt Engineering wird *Chain-of-Thought-Prompting* (CoT) genannt und bezeichnet eine Serie von Zwischenschritten einer sprachlich formulierten Schlussweise, die zu einer endgültigen Ausgabe führt (Wei et al., 2023). Kojima et al. (2022) führen etwa das Beispiel an, *Zero-shot-Prompts* mit „Denken wir Schritt für Schritt“ zu ergänzen, Ramlochan (2023) behauptet, noch bessere Lösungen ergäben sich mit dem Zusatz „Lass uns Schritt für Schritt vorgehen, um sicherzugehen, dass wir die richtige Antwort haben“. Indem das generative KI-Sprachmodell die einfache Ausgabe des Ergebnisses in diesem Fall durch eine ausführliche Ausgabe eines Lösungsweges ersetzt, führt die Technik des Chain-of-Thought Promptings gerade bei mathematischen Fragen zu deutlich besseren und vor allem häufiger korrekten Lösungen (Wei et al., 2023).

Mit der Technik des *Ask-me-Anything-Prompting* schlagen Arora et al. (2023) vor, das generative KI-Sprachmodell zum Stellen von Fragen anzuregen. Dabei wird ein Kontext vorgegeben, eine Behauptung aufgestellt und die Behauptung vom KI-Sprachmodell als richtig oder falsch klassifiziert. In-

dem Arora et al. (2023) in einem Few-shot-Szenario abwechselnd eine Behauptung und eine Frage eingaben, wurde das System zum Stellen von Fragen trainiert. Die Ausgabe ist danach nicht mehr eine einfache Antwort, sondern eine Fragestellung bezüglich der Behauptung. Für das Lösen mathematischer Probleme sehen wir hierin eine gute Möglichkeit, mit GPT-4 in einen aktiven Lösungsprozess zu kommen. Anders als es Arora et al. (2023) in ihrem Few-shot-Prompting vorschlagen, ergänzen wir die Prompts mit der einfachen Aufforderung Fragen zu stellen: „Stelle notwendige Fragen, die du zur Beantwortung der Frage benötigst.“ Dies verändert die Nutzeranwendung dahingehend, dass nun nicht mehr die Benutzerin oder der Benutzer Fragen stellt, sondern das Modell seinerseits Fragen an die Benutzerin bzw. den Benutzer formuliert. Es entsteht eine Abfrageschleife, die wir im Folgenden als *Query-Loop-Prompting* bezeichnen möchten. Ein Query-Loop bezieht sich auf eine Schleife oder einen wiederholenden Prozess, bei dem eine Suchabfrage oder Anfrage wiederholt durchgeführt wird, bis bestimmte Bedingungen erfüllt sind oder ein gewünschtes Ergebnis erzielt wurde. Dies führt das KI-Sprachmodell durch unseren Zusatz im Prompt solange durch, bis sich Input und Output soweit annähern, dass der gewünschte Output generiert wird.

### Methodisches Vorgehen

Um die Leistungsfähigkeit des neuen KI-Sprachmodells GPT-4 im Hinblick auf mathematische Fähigkeiten zu untersuchen, haben wir im Rahmen von Modellvalidierungen verschiedene Varianten eines Prompts getestet und die entsprechenden Outputs bewertet. Dabei wurden vier Modellvarianten der trainingsbasierten Prompt-Techniken verwendet, die unterschiedliche Anreize zur Generierung eines Outputs setzen. Vor jedem neuen Input wurde ChatGPT mit einem neuen Chat gestartet, um dem integrierten *Few-shot-Learning* von GPT-4 vorerst entgegenzuwirken und das System mit jedem Test neu zu kalibrieren. Dadurch soll der Vergleich der Varianten innerhalb einer kalibrierten Umgebung gewährleistet werden. Getestet wurden die folgenden vier Varianten von Prompts zu zwei Problemlöseaufgaben:

1. In Variante A wurden die Aufgaben in einem Zero-shot-Szenario mit den Worten „Lass uns Schritt für Schritt vorgehen“ ergänzt, um eine zusätzliche Chain-of-Thought durch das generative KI-Sprachmodell zu provozieren.
2. Variante B erweitert die Anweisung unter Variante A mit den Worten „... um sicherzustellen, dass wir die richtige Antwort haben.“ Dies

soll dazu führen, dass GPT-4 in der selbst generierten Chain-of-Thought die Korrektheit der Schlussfolgerungen einbezieht.

3. Mit Variante C kann GPT-4 Fragen zur Aufgabenstellung stellen, um auf diese Weise die Schwierigkeiten zu erkennen, die das generative KI-Sprachmodell im Zusammenhang mit der zugewiesenen Aufgabe sieht. Hierzu wurde das Zero-shot-Szenario mit Chain-of-thought und einem zusätzlichen Query-Loop um die Eingabe „Stelle notwendige Fragen, die du zur Beantwortung der Frage benötigst.“ ergänzt.
4. In der letzten Variante D wurde GPT-4 neben der eigentlichen Aufgabe eine analoge Aufgabe gestellt und diese von uns beantwortet. Dieses Few-shot-Szenario setzt sich aus einer Aufgabe (Frage 1), einem Lösungsbeispiel (Antwort 1) in Form eines Chain-of-Thoughts und einer neuen Aufgabe (Frage 2) zusammen.

Für die Modellvalidierungen wurden ein arithmetisches und ein algebraisches Problem verwendet. Die Lösung des arithmetischen Problems setzt Basiskompetenzen in den Grundrechenarten und im Umgang mit dem Größenbereich Längen voraus, während das algebraische Problem, neben den Basiskompetenzen in den Grundrechenarten, auch die Mustererkennung und die Beschreibung einer mathematischen Struktur verlangt. Für das arithmetische Problem wurde die Aufgabe „Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul?“ aus Ott (2018) verwendet. Als algebraische Problemlöseaufgabe wurde die Aufgabe „Welche Differenzen besitzen benachbarte Palindromzahlen?“ aus Schorcht (2022) genutzt.

Die arithmetische Problemlöseaufgabe beinhaltet ein einfaches Gleichungssystem mit zwei Gleichungen und zwei Variablen, welches sich über das Gleichsetzungsverfahren lösen lässt.

$$\begin{aligned} 78 \text{ cm} + 20 \text{ cm} &= \text{Größe von Klaus} \\ \text{Größe von Paul} + 15 \text{ cm} &= \text{Größe von Klaus} \\ \text{Größe von Paul} + 15 \text{ cm} &= 78 \text{ cm} + 20 \text{ cm} \\ \text{Größe von Paul} &= 83 \text{ cm} \end{aligned}$$

Die algebraische Problemlöseaufgabe lässt sich nur lösen, wenn die mathematische Struktur der Stellenwerte im Zehnersystem in den Blick genommen wird. Als benachbarte Palindromzahlen sind solche Palindromzahlen zu verstehen, die in der Zahlenreihe direkt aufeinanderfolgen. Dies sind beispielsweise 1 und 2, 11 und 22, 101 und 111, 1551 und 1661, 23432 und 23532, aber auch 99 und 101 oder 1991 und 2002. Mögliche Differenzen dieser Beispiele sind 1, 11, 10, 110, 100 und 2. Dies wären einfache Antwortmöglichkeiten, die sich mit arith-



<p><b>Input</b></p> <p>Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul?</p>	<p><b>Input</b></p> <p>Welche Differenzen besitzen benachbarte Palindromzahlen?</p>
<p><b>Output</b></p> <p>Wenn Max auf eine 20 cm hohe Kiste steigt und dann genauso groß wie Klaus ist, muss Klaus vorher 78 cm minus 20 cm, also 58 cm groß gewesen sein.</p> <p>Paul ist 15 cm kleiner als Klaus, also 58 cm minus 15 cm, also 43 cm groß.</p> <p>Also ist Paul 43 cm groß.</p>	<p><b>Output</b></p> <p>Palindromzahlen sind Zahlen, die vorwärts und rückwärts gelesen das gleiche Ergebnis liefern. Um die Differenzen zwischen benachbarten Palindromzahlen zu bestimmen, können wir uns einige Beispiele ansehen:</p> <p>Zwischen den Palindromzahlen 11 und 22 liegt eine Differenz von 11. Zwischen den Palindromzahlen 22 und 33 liegt eine Differenz von 11. Zwischen den Palindromzahlen 99 und 101 liegt eine Differenz von 2. Zwischen den Palindromzahlen 101 und 111 liegt eine Differenz von 10.</p> <p>Man kann feststellen, dass die Differenzen zwischen aufeinanderfolgenden Palindromzahlen variieren. Es gibt keine feste Regel oder Muster für diese Differenzen, da sie von der Anzahl der Stellen der Palindromzahlen abhängen. Es ist jedoch möglich, dass die Differenzen sowohl gerade als auch ungerade Werte annehmen können.</p> <p>Es ist auch interessant zu beachten, dass es unendlich viele Palindromzahlen gibt. Daher sind die möglichen Differenzen zwischen benachbarten Palindromzahlen ebenfalls unendlich.</p>

Abbildung 3. Zero-shot-Prompt einer arithmetischen Problemlöseaufgabe (links) sowie einer algebraischen Problemlöseaufgabe (rechts) und die generierten Antworten aus GPT-4

metischen Basiskompetenzen erschließen lassen – allerdings ohne die Beispiele zu verallgemeinern. Betrachten wir die Muster in der Veränderung der Stellenwerte in Potenzschreibweise, kann die mathematische Struktur auch algebraisch erschlossen werden. Benachbarte zweistellige Palindromzahlen der Form  $a \cdot 10^1 + a \cdot 10^0$  und  $(a + 1) \cdot 10^1 + (a + 1) \cdot 10^0$  (mit  $a \neq 9$ ) besitzen die Differenz 11, da sich sowohl die Einer- als auch die Zehnerstelle um je 1 erhöht. Bei einer dreistelligen Palindromzahl der Form  $a \cdot 10^2 + b \cdot 10^1 + a \cdot 10^0$  (mit  $b \neq 9$ ) ist der direkte Nachbar  $a \cdot 10^2 + (b + 1) \cdot 10^1 + a \cdot 10^0$  und die Differenz beträgt 10, da sich der mittlere Stellenwert um 1 erhöht. Bei vierstelligen benachbarten Palindromzahlen der Form

$$a \cdot 10^3 + b \cdot 10^2 + b \cdot 10^1 + a \cdot 10^0 \quad (\text{mit } b \neq 9)$$

und

$$a \cdot 10^3 + (b + 1) \cdot 10^2 + (b + 1) \cdot 10^1 + a \cdot 10^0 \quad (\text{mit } b \neq 9)$$

beträgt die Differenz 110, da sich die beiden mittleren Stellenwerte um 1 erhöhen. Für die oben angenommenen Fälle, in denen die sich verändernden Stellenwerte 9 sind, ergibt sich jeweils eine Nachbarzahl mit Differenz 11 oder 2. Tatsächlich variieren die Differenzen bei Palindromzahlen genau dann, wenn sich neben den mittleren Stellenwerten auch die äußeren Stellenwerte verändern. Insgesamt ergeben sich dadurch immer Differenzen von 2 sowie Zehnerpotenzen faktorisiert mit 1 und 11.

Beide Probleme können von GPT-4 im Zero-shot-Szenario nicht vollständig gelöst werden und sind fehlerhaft (Abb. 3). Das generative KI-Sprachmodell erfasst in Teilen die arithmetische Vorgehensweise, kann diese aber im ersten Aufgabenbeispiel nicht korrekt umsetzen und in der zweiten Aufgabe nicht in eine mathematische Struktur überführen. Im Folgenden sollen die trainingsbasierten Prompt-Techniken verwendet werden, indem der Zero-shot ergänzt wird, um ein besseres Ergebnis in der Ausgabe von GPT-4 zu generieren.

#### Variante A: Zero-shot-Szenario und Anregung zum Chain-of-Thought

Unter Variante A (Abb. 4) liefert GPT-4 eine deutlich differenziertere Darstellung des Lösungsprozesses und gliedert den Output mit Teilüberschriften: „Schritt 1“, „Schritt 2“ und „Schritt 3“. Diese Teilschritte ermöglichen es dem generativen KI-Sprachmodell, eine korrekte Lösung auszugeben. Dies impliziert, dass der einfache Zusatz „Lass uns Schritt für Schritt vorgehen“ vermutlich schon für eine deutlich bessere Leistung im Bereich einfacher Problemlöseaufgaben führt, die die Kalkulation einfacher Rechenschritte erfordern. Durch die vorgenommene Veränderung am Prompt wird das System dazu gezwungen, die einzelnen Schritte auszuführen und generiert darüber vermutlich eine höhere Wahrscheinlichkeit für korrekte Outputs.

Der Output der algebraischen Problemlöseaufgabe unter Variante A (Abb. 4) liefert ebenfalls ei-

Input

Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul? Lass uns Schritt für Schritt vorgehen.

Output

Schritt 1: Berechne Max' Größe, wenn er auf der Kiste steht.

Max ist 78 cm groß und steigt auf eine 20 cm hohe Kiste. Um seine neue Größe zu berechnen, addieren wir seine Größe und die Höhe der Kiste zusammen:

$$78 \text{ cm (Max' Größe)} + 20 \text{ cm (Kistenhöhe)} = 98 \text{ cm}$$

Max ist also 98 cm groß, wenn er auf der Kiste steht.

Schritt 2: Bestimme Klaus' Größe.

Da Max auf der Kiste genauso groß ist wie Klaus, wissen wir, dass Klaus ebenfalls 98 cm groß ist.

Schritt 3: Berechne Pauls Größe.

Paul ist 15 cm kleiner als Klaus. Um Pauls Größe zu berechnen, subtrahieren wir 15 cm von Klaus' Größe:

$$98 \text{ cm (Klaus' Größe)} - 15 \text{ cm} = 83 \text{ cm}$$

Paul ist 83 cm groß.

Input

Welche Differenzen besitzen benachbarte Palindromzahlen? Lass uns Schritt für Schritt vorgehen.

Output

Um die Differenz zwischen benachbarten Palindromzahlen zu verstehen, betrachten wir zunächst, was eine Palindromzahl ist. Eine Palindromzahl ist eine Zahl, die vorwärts und rückwärts gelesen gleich ist. Beispiele für Palindromzahlen sind 121, 222, 353, 1001 usw.

Um die Differenzen zwischen benachbarten Palindromzahlen zu ermitteln, gehen wir Schritt für Schritt vor:

1. Zunächst listen wir einige Palindromzahlen in aufsteigender Reihenfolge auf: 11, 22, 33, 44, ..., 99, 101, 111, ..., 181, 191, 202, ..., 313, 323, ..., 999, 1001, 1111, ...
2. Nun berechnen wir die Differenzen zwischen aufeinanderfolgenden Palindromzahlen:
 
$$(22 - 11) = 11$$

$$(33 - 22) = 11$$

$$(44 - 33) = 11$$

...

$$(99 - 88) = 11$$

$$(101 - 99) = 2$$

$$(111 - 101) = 10$$

...

$$(191 - 181) = 10$$

$$(202 - 191) = 11$$

...

$$(313 - 303) = 10$$

$$(323 - 313) = 10$$

...

$$(999 - 989) = 10$$

$$(1001 - 999) = 2$$

$$(1111 - 1001) = 110$$

Es ist ersichtlich, dass die Differenzen zwischen benachbarten Palindromzahlen variieren und keine feste Regel existiert, die für alle Palindromzahlen gilt. In den meisten Fällen liegt die Differenz jedoch bei 10 oder 11, vor allem für Palindromzahlen mit zwei oder drei Stellen. Bei größeren Palindromzahlen können die Differenzen jedoch deutlich variieren.

Abbildung 4. Input der arithmetischen Problemlöseaufgabe (oben) und der algebraischen Problemlöseaufgabe (unten) unter der CoT-Variante A, sowie die generierten Antworten aus GPT-4

ne differenzierte Darstellung des Lösungsprozesses. Dazu gliedert GPT-4, anders als im Zero-shot-Szenario, die eigene Lösung in zwei Teile: Zunächst listet das generative KI-Sprachmodell einige Palindromzahlen bis 1111, um im zweiten Schritt die

Differenzen zwischen diesen Palindromzahlen zu bestimmen. Dadurch schafft es GPT-4, die neue Differenz 110 zu bestimmen, die in der ursprünglichen Lösung noch fehlte. Da die gewählten Beispiele sich allerdings nur im Zahlenraum 11 bis 1111 bewegen,

ist eine algebraische Mustererkennung erschwert und die Beschreibung der mathematischen Struktur entfällt dementsprechend. Der Versuch einer Beschreibung scheitert auf Grundlage dieser Auflistung. Der Blick auf die algebraische Struktur der Stellenwerte benachbarter Palindromzahlen wird von GPT-4 nicht in den Lösungsprozess integriert.

### Version B: Zero-shot-Szenario, Anregung zum erweiterten Chain-of-Thought

Unter Variante B und dem Zusatz „... um sicherzustellen, dass wir die richtige Antwort haben.“ liefert GPT-4 bei der arithmetischen Problemlöseaufgabe die gleiche Antwort, wie unter Variante A. Lediglich der Satz „Okay, lass uns das Problem Schritt für Schritt lösen.“ ist eine neue Ergänzung durch das KI-Sprachmodell. Die Lösung ist, wie wir erwartet hatten, korrekt (Abb. 5 links).

Die algebraische Problemlöseaufgabe erhält unter der leichten Veränderung der Variante eine neue Struktur (Abb. 5 rechts). GPT-4 generiert nun eine vierstufige Lösung, die sich, anders als unter Variante A, an den Zahlenräumen orientiert. Im ersten Schritt werden einstellige Palindromzahlen identi-

fiziert und die Differenz benachbarter Palindromzahlen mit 1 angegeben. Im zweiten Schritt tritt nun der zweistellige Zahlenraum in den Blick. Hier generiert GPT-4 eine vollständige Liste aller Palindromzahlen, welche unter Variante A noch fehlten. Korrekt bestimmt das generative KI-Sprachmodell die Differenz 11. Im dritten Schritt werden dreistellige Palindromzahlen betrachtet. Eine Auflistung aller Möglichkeiten erfolgt nicht. Die unterschiedlichen Differenzen 10 und 11, die GPT-4 unter Variante A noch konkret identifizieren konnte, werden hier nicht genannt. Das KI-Sprachmodell benennt allerdings das Phänomen, indem das Modell von einer Variation der Differenzen spricht, die mit den generierten Beispielen aber nicht untermauert werden kann. Diese Veränderungen in den Differenzen überträgt das System im vierten Schritt auf vierstellige und höhere Palindromzahlen. In einer Zusammenfassung bestätigt das generative KI-Sprachmodell dann die Veränderungen der Differenzen über alle Palindromzahlen hinweg. Differenzen, die hierbei als Lösung dienen, sind 1, 10 und 11. Die Lösungsmenge ist damit geringer als unter Variante A, allerdings gelingt es GPT-4, unter Variante B eine Fallunterscheidung vorzunehmen.

Input	Input
Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul? Lass uns Schritt für Schritt vorgehen, um sicherzustellen, dass wir die richtige Antwort haben.	Welche Differenzen besitzen benachbarte Palindromzahlen? Lass uns Schritt für Schritt vorgehen, um sicherzustellen, dass wir die richtige Antwort haben.
Output	Output
<p>Okay, lass uns das Problem Schritt für Schritt lösen.</p> <p>Schritt 1: Max ist 78 cm groß und steigt auf eine 20 cm hohe Kiste. Um herauszufinden, wie groß Max dann ist, addieren wir seine Größe mit der Höhe der Kiste.</p> $78 \text{ cm (Max' Größe)} + 20 \text{ cm (Kistenhöhe)} = 98 \text{ cm}$ <p>Schritt 2: Wenn Max auf der Kiste steht, ist er genauso groß wie Klaus. Das bedeutet, dass Klaus ebenfalls 98 cm groß ist.</p> <p>Schritt 3: Paul ist 15 cm kleiner als Klaus. Um Pauls Größe herauszufinden, subtrahieren wir 15 cm von Klaus' Größe.</p> $98 \text{ cm (Klaus' Größe)} - 15 \text{ cm} = 83 \text{ cm}$ <p>Paul ist 83 cm groß.</p>	<p>Palindromzahlen sind Zahlen, die vorwärts und rückwärts gelesen gleich sind. Zum Beispiel sind 121 und 131 Palindromzahlen. Um die Differenzen zwischen benachbarten Palindromzahlen zu finden, betrachten wir zunächst die kleinsten Palindromzahlen und ihre Differenzen:</p> <ol style="list-style-type: none"> <li>Einstellige Palindromzahlen: In diesem Fall sind alle einstelligen Zahlen Palindrome (1, 2, 3, 4, 5, 6, 7, 8 und 9). Die Differenzen zwischen diesen benachbarten Palindromzahlen sind immer 1.</li> <li>Zweistellige Palindromzahlen: Die zweistelligen Palindrome sind 11, 22, 33, 44, 55, 66, 77, 88 und 99. Auch hier betragen die Differenzen zwischen den benachbarten Palindromzahlen 11.</li> <li>Dreistellige Palindromzahlen: Hier wird es etwas komplizierter. Die kleinsten dreistelligen Palindrome sind 101, 111, 121, 131, ..., 191 und 202. Die Differenzen zwischen diesen benachbarten Palindromzahlen sind nicht konstant, sondern variieren. Zum Beispiel beträgt die Differenz zwischen 101 und 111 zehn, während die Differenz zwischen 111 und 121 ebenfalls zehn beträgt.</li> <li>Höhere Palindromzahlen: Bei vierstelligen und höheren Palindromzahlen variiert die Differenz zwischen benachbarten Palindromzahlen weiter.</li> </ol> <p>Zusammenfassend lässt sich sagen, dass die Differenzen zwischen benachbarten Palindromzahlen variieren und nicht konstant sind, insbesondere bei höheren Palindromzahlen.</p>

Abbildung 5. Input der arithmetischen Problemlöseaufgabe (links) und der algebraischen Problemlöseaufgabe (rechts) unter CoT-Variante B, sowie die generierten Antworten aus GPT-4



Der Blick auf die algebraische Struktur der Stellenwerte bleibt aber auch hier für GPT-4 verborgen.

### Version C: Zero-shot-Szenario, Anregung zum erweiterten Chain-of-Thought und Query-Loop

Variante C ermöglicht GPT-4 Rückfragen bezüglich des Aufgabenverständnisses und setzt hier beim ersten Schritt im Problemlöseprozess an. Interessanterweise wird eine ähnliche Lösung für das arithmetische Problem generiert und keine Frage gestellt (Abb. 6 links). Dies zeigt nochmals, dass Variante A bereits zur Unterstützung des Modells im Lösen von arithmetischen Problemlöseaufgaben auszureichen scheint.

Unter Variante C stellt GPT-4 allerdings bei der algebraischen Problemlöseaufgabe eine entscheidende Frage, die das Problem des generativen KI-Sprachmodells in der Verarbeitung algebraischer Strukturen offenlegt (Abb. 6 rechts): Nachdem GPT-4 einige Beispiele für Palindromzahlen generiert, stellt das System eine Anfrage bezüglich eines festzulegenden Intervalls bzw. einer diskreten Lösungsmenge, in dem die Differenzen bestimmt werden sollen: „Möchten Sie, dass ich die Differenzen zwischen benachbarten Palindromzahlen in einem bestimmten Zahlenbereich berechne oder für eine bestimmte Anzahl von Palindromzahlen?“ Wir entschieden uns, mit einem zweiten Input die

Sachlage zu verschärfen und keine diskrete Vorgabe zu machen, um das System zur Betrachtung der mathematischen Struktur zu bewegen. Mit dem Prompt „Für alle benachbarten Palindromzahlen!“ gab GPT-4 eine Beschreibung der eigenen Herausforderung mit der gestellten Aufgabe aus. Die einfache Berechnung von Differenzen benachbarter Palindromzahlen führt zu einem unendlichen Prozess von Operationen, die GPT-4 nicht leisten kann. Statt in diesem Fall auf Muster und Strukturen zu achten, generiert GPT-4 eine Antwort, die die Aufgabe als unlösbar klassifiziert. Es ist ersichtlich, dass GPT-4 angesichts der Aufgabe, einen Lösungsprozess für algebraische Probleme zu generieren, deutlich herausgefordert ist.

### Version D: Few-shot-Szenario mit Chain-of-Thought

Variante D des Prompts führt neben der eigentlichen Problemlöseaufgabe auch eine weitere Aufgabe mit Lösungsbeispiel an, um die Wahrscheinlichkeit für einen korrekten Output zu erhöhen. Für das arithmetische Problem wurde ein strukturgleiches Problem gewählt, das sich lediglich durch einen anderen Kontext und durch Veränderung der Namen und mathematischen Angaben unterscheidet: „Timo ist 1,78 m groß. Wenn er auf eine 10 cm hohe Stufe steigt, ist er genauso groß wie Peter. Tina ist

<p><b>Input</b></p> <p>Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul? Lass uns Schritt für Schritt vorgehen, um sicherzustellen, dass wir die richtige Antwort haben. Stelle notwendige Fragen, die du zur Beantwortung der Frage benötigst.</p> <p><b>Output</b></p> <p>Schritt 1: Berechnen, wie groß Klaus ist.</p> <p>Max ist 78 cm groß und wird genauso groß wie Klaus, wenn er auf eine 20 cm hohe Kiste steigt. Um Klaus' Größe herauszufinden, addieren wir Max' Größe (78 cm) und die Höhe der Kiste (20 cm) zusammen:</p> <p><math>78 \text{ cm (Max' Größe)} + 20 \text{ cm (Kistenhöhe)} = 98 \text{ cm}</math></p> <p>Klaus ist also 98 cm groß.</p> <p>Schritt 2: Berechnen, wie groß Paul ist.</p> <p>Es wurde gesagt, dass Paul 15 cm kleiner ist als Klaus. Um Pauls Größe herauszufinden, subtrahieren wir diese 15 cm von Klaus' Größe (98 cm):</p> <p><math>98 \text{ cm (Klaus' Größe)} - 15 \text{ cm} = 83 \text{ cm}</math></p> <p>Paul ist 83 cm groß.</p>	<p><b>Input #1</b></p> <p>Welche Differenzen besitzen benachbarte Palindromzahlen? Lass uns Schritt für Schritt vorgehen, um sicherzustellen, dass wir die richtige Antwort haben. Stelle notwendige Fragen, die du zur Beantwortung der Frage benötigst.</p> <p><b>Output #1</b></p> <p>Um die Differenzen zwischen benachbarten Palindromzahlen zu berechnen, müssen wir zunächst klären, welche Zahlen als benachbarte Palindromzahlen betrachtet werden sollen. Palindromzahlen sind Zahlen, die rückwärts gelesen dieselbe Zahlenfolge haben wie vorwärts. Zum Beispiel sind 121, 313 und 595 Palindromzahlen.</p> <p>Möchten Sie, dass ich die Differenzen zwischen benachbarten Palindromzahlen in einem bestimmten Zahlenbereich berechne oder für eine bestimmte Anzahl von Palindromzahlen?</p> <p><b>Input #2</b></p> <p>Für alle benachbarten Palindromzahlen!</p> <p><b>Output #2</b></p> <p>Es ist nicht möglich, Differenzen für alle benachbarten Palindromzahlen zu berechnen, da es unendlich viele Palindromzahlen gibt. Allerdings kann ich Ihnen einige Beispiele von benachbarten Palindromzahlen und ihren Differenzen geben. Bitte geben Sie mir einen Zahlenbereich oder eine Anzahl von Palindromzahlen, für die Sie die Differenzen berechnet haben möchten.</p>
---	---

Abbildung 6. Input der arithmetischen Problemlöseaufgabe (links) und der algebraischen Problemlöseaufgabe (rechts) unter CoT-Variante C, sowie die generierten Antworten aus GPT-4



Input

Frage: Timo ist 1,78 m groß. Wenn er auf eine 10 cm hohe Stufe steigt, ist er genauso groß wie Peter. Tina ist 12 cm kleiner als Peter. Wie groß ist Tina?

Antwort: Wenn Timo auf der Stufe steht, dann ist er 10 cm größer und genauso groß wie Peter. Daher ist Peter  $1,78 \text{ m} + 10 \text{ cm}$ , also 1,88 m. Wenn Tina nun 12 cm kleiner als Peter ist, dann ist sie  $1,88 \text{ m} - 12 \text{ cm}$  groß, also 1,76 m.

Frage: Max ist 78 cm groß. Wenn er auf eine 20 cm hohe Kiste steigt, ist er genauso groß wie Klaus. Paul ist 15 cm kleiner als Klaus. Wie groß ist Paul?

Output

Antwort: Wenn Max auf der Kiste steht, dann ist er 20 cm größer und genauso groß wie Klaus. Daher ist Klaus  $78 \text{ cm} + 20 \text{ cm}$ , also 98 cm. Wenn Paul nun 15 cm kleiner als Klaus ist, dann ist er  $98 \text{ cm} - 15 \text{ cm}$  groß, also 83 cm.

Abbildung 7. Input der arithmetischen Problemlöseaufgabe unter CoT-Variante D, sowie die generierte Antwort aus GPT-4

12 cm kleiner als Peter. Wie groß ist Tina?“ Gleichzeitig wird eine kurze Antwort geliefert, um dem System eine mögliche Antwortstruktur vorzugeben: „Wenn Timo auf der Stufe steht, dann ist er 10 cm größer und genauso groß wie Peter. Daher ist Peter  $1,78 \text{ m} + 10 \text{ cm}$ , also 1,88 m. Wenn Tina nun 12 cm kleiner als Peter ist, dann ist sie  $1,88 \text{ m} - 12 \text{ cm}$  groß, also 1,76 m.“ Anschließend wird im Prompt die ursprüngliche Problemlöseaufgabe gestellt. Diese Zugabe eines Shots liefert in unserem Versuch interessanterweise einen deutlich kürzeren korrekten Output in Relation zu den anderen Outputs unter Varianten A bis C (Abb. 7). Das generative KI-Sprachmodell orientiert sich offensichtlich an der vorgegebenen Antwort und führt die Lösung auch dementsprechend kürzer aus.

Für die algebraische Problemlöseaufgabe wurde eine ähnliche Problemstellung gewählt, die ebenfalls Differenzen in den Blick nimmt: „Welche Differenzen besitzen eine Zahl und ihre Spiegelzahl?“ Zur Lösung der Aufgabe wurde im Antwortbeispiel der Fokus bewusst auf die algebraische Struktur gelegt und die Hervorhebung der Stellenwerte durch den Hinweis auf die Potenzschreibweise der Zahl ermöglicht. Die Einführung von Variablen für die einzelnen Ziffern sollte das System dazu anregen, eine Lösung mit Variablen zu generieren. Anschließend erfolgte im Prompt die Eingabe der eigentlichen Problemlöseaufgabe (Abb. 8).

GPT-4 generiert eine Lösung, die die Stellenwerte berücksichtigt. Somit ist mit Variante D ein wesentlicher Schritt zur Beschreibung von Mustern gelungen, der in allen vorherigen Varianten fehlte. Allerdings verfehlt GPT-4 den Sprung von einer einfachen Beschreibung der Muster zu einer Formulierung der mathematischen Struktur. Das generative KI-Sprachmodell startet seinen Lösungsprozess mit der Erklärung für Palindromzahlen und nennt

drei Beispiele. Anschließend verweist GPT-4 auf die Komplexität der Aufgabe, da die Lösung von der Anzahl der Stellenwerte und den gewählten Ziffern abhängt. Danach erfolgt eine Fallunterscheidung zwischen Zahlen mit ungerader und gerader Zifferanzahl, ohne den Zahlenraum einzugrenzen. Hierbei schafft es GPT-4 erstmals korrekt eine allgemeine Beschreibung der Muster vorzunehmen und überwindet mithilfe der Variante D die Problematik der einzelnen Berechnungen der Differenzen. Korrekt führt das System aus, dass sich bei Palindromzahlen ungerader Zifferanzahl der mittlere Stellenwert verändert. Als mögliche Differenz generiert GPT-4 allerdings fälschlicherweise nur die Zahl 2. Für Palindromzahlen mit gerader Zifferanzahl generiert das System korrekt, dass die Veränderung der Ziffern an den beiden mittleren Stellenwerten erfolgt. Vermutlich aufgrund der Varianz der Differenzen bei Veränderungen mehrerer Stellenwerte (bspw. 2992 und 3003) hält GPT-4 fest, dass die Differenzen nicht konstant sind. Allerdings handelt es sich durch die wiederkehrende Differenz 11 um eine mögliche Lösung und könnte in eine Beschreibung der Lösungsmenge integriert werden. Auch die Ausnahmen beim Wechsel der Zifferanzahl werden in der Ausgabe durch GPT-4 explizit beschrieben. Zudem verweist das KI-Sprachmodell auf die Beschränkung der Aussagen auf das Zehnersystem. GPT-4 ist hier ein wesentlicher Schritt zur Erfassung von Mustern gelungen, der in unserem Versuch zuvor nicht möglich erschien. Allerdings verliert der Lösungsvorschlag von GPT-4 dabei die Angabe einer Lösungsmenge.

Während der Erprobung 14 verschiedener Ausführungen des Lösungsbeispiels unter Variante D konnte GPT-4 in keiner generierten Antwort zu einer angemessenen Lösung gelangen, die als Differenzen die 2 und faktorisierte Zehnerpotenzen

Input

Frage: Welche Differenzen besitzen eine Zahl und ihre Spiegelzahl?

Antwort: Eine Zahl und ihre Spiegelzahl bilden ein Zahlenpaar, deren Ziffern jeweils in umgekehrter Reihenfolge gelesen werden können, beispielsweise 123 und 321 oder 6789 und 9876. Wir verwenden für die einzelnen Ziffern Buchstaben a, b, c, usw. als Variablen. Zahlen können als Summen von Zehnerpotenzen geschrieben werden.

Für zweistellige Zahlen ergibt sich die Form  $10a+1b$ . Die Differenz aus zweistelligen Zahlen und ihrer Spiegelzahl ist dementsprechend  $(10a+1b)-(10b+1a)$ . Schreiben wir diese Differenz in die Form  $(10a+1b)-(9b+1b+1a)$  um, so ziehen wir von der Zahl  $10a+1b$  durch Subtraktion der Spiegelzahl immer ihre Quersumme  $1b+1a$  ab. Als Differenz bleibt  $9a-9b=9(a-b)$  und somit unabhängig von den Variablen eine Zahl, die durch 9 teilbar ist.

Dies gilt für alle Zahlenpaare im Zehnersystem, da immer die Quersumme von der gewählten Zahl subtrahiert wird und als Differenz eine durch 9 teilbare Zahl entsteht.

Frage: Welche Differenzen besitzen benachbarte Palindromzahlen?

Output

Antwort: Palindromzahlen sind Zahlen, die von vorne und von hinten gleich gelesen werden, wie zum Beispiel 121, 12321 oder 45654. Die Frage nach den Differenzen benachbarter Palindromzahlen ist etwas komplexer, da die Differenz zwischen zwei benachbarten Palindromzahlen von ihrer Stellenzahl und den spezifischen Ziffern abhängt.

Für ein einfacheres Verständnis der Frage können wir die Betrachtung auf Palindromzahlen mit ungerader Zifferanzahl beschränken. Die mittlere Ziffer dieser Zahlen ist immer diejenige, die sich verändert, um das nächste Palindrom zu erreichen. In diesem Fall beträgt die Differenz zwischen zwei aufeinanderfolgenden Palindromzahlen immer 2, wenn die Differenz zwischen den mittleren Ziffern betrachtet wird. Das liegt daran, dass, wenn die mittlere Ziffer um 1 erhöht wird, die resultierende Zahl immer noch ein Palindrom ist.

Allerdings ist dies nicht mehr der Fall, wenn wir Palindromzahlen mit gerader Zifferanzahl betrachten. In diesem Fall sind die Differenzen zwischen benachbarten Palindromzahlen nicht konstant, da eine Änderung an zwei Stellen (den beiden mittleren) notwendig ist, um das nächste Palindrom zu erreichen.

Es ist wichtig zu beachten, dass diese Regel nicht gilt, wenn wir den Übergang von einer Zahlenklasse zur nächsten betrachten (z.B. von der größten dreistelligen Palindromzahl 999 zur kleinsten vierstelligen Palindromzahl 1001, deren Differenz 2 ist).

Diese Analyse betrachtet ausschließlich Palindromzahlen im Dezimalsystem. In anderen Zahlensystemen könnten andere Muster gelten.

Abbildung 8. Input der algebraischen Problemlöseaufgabe unter CoT-Variante D, sowie die generierte Antwort aus GPT-4

mit 1 und 11 festhält. Die hier vorgestellte Antwort ist die bestmögliche Lösung, die das generative KI-Sprachmodell erstellen konnte.

### Fazit

Für die vorgestellten Beispiele, unter Verwendung des generativen KI-Sprachmodells GPT-4 und in Bezug auf die Lösung von Problemlöseaufgaben mit Basiskompetenzen im arithmetischen Bereich, erscheint Variante A (Zero-shot und Anregung zum Chain-of-Thought) zur Generierung einer korrekten Antwort ausreichend. Weitere Beispiele müssen zeigen, ob diese Prompt-Technik für eine Vielzahl von

Problemlöseaufgaben auf diesem Leistungsniveau effektiv ist. Da die Entwicklungen im Bereich der KI-Forschung deutlich an Geschwindigkeit zunehmen und Veränderungen mittlerweile im nahezu wöchentlichen Takt auftreten, könnten in zukünftigen Varianten die Prompt-Techniken bereits in die entsprechenden Schnittstellen integriert sein. Die weiteren Prompt-Techniken, wie die erweiterte Chain-of-Thought und der Query Loop, erzielten keine nennenswerte Verbesserung der Ergebnisse, als unter Variante A. Variante D, ein Few-shot-Szenario mit Chain-of-Thought, erzielt lediglich eine an die Nutzerin oder den Nutzer ausgerichtete

Ausgabe. Dies kann jedoch von Bedeutung sein, wenn die Form der Antwort von GPT-4 im schulischen Gebrauch entsprechende Vorgaben erfüllen soll.

Die vorgestellte algebraische Problemlöseaufgabe, die die Beschreibung eines Musters erfordert und eine Übertragung des Problems in eine mathematische Struktur, stellt nach wie vor eine deutliche Herausforderung für GPT-4 dar. Mit keiner der verwendeten Prompt-Techniken konnte GPT-4 eine vollständige und zufriedenstellende Lösung des Problems generieren. Gleichwohl zeigte sich im Versuch auch hier Variante A als sehr wirkungsvoll. Ein Zero-shot-Szenario mit der Anregung zu einer Chain-of-Thought führte hierbei zu einer deutlichen Verbesserung der Ausgabe. Unter Variante C konnte mit der Strategie des Query-Loops die zentrale Herausforderung für das generative KI-Sprachmodell identifiziert werden. Dies lieferte eine wertvolle Rückmeldung zur Verwendung eines passenden Lösungsbeispiels unter Variante D (Few-shot-Szenario mit Chain-of-Thought). Zwar schafft GPT-4 auch unter Variante D nicht die erwünschte Leistung, liefert allerdings eine erste Lösung, die die algebraischen Strukturen mehr in den Blick nimmt. Hier wird der große Einfluss des Lösungsbeispiels über das Few-shot-Learning des generativen KI-Sprachmodells ersichtlich – was auch für den schulischen Einsatz dieser Technik eine hohe Relevanz besitzt. Schülerinnen und Schüler müssen also über die notwendigen Problemlösefähigkeiten verfügen, um selbst ein passendes Lösungsbeispiel zu verwenden oder zu generieren. Selbst die Suche nach einer analogen Lösung im Internet macht in unserem Beispiel ein Verständnis für Problemlösestrategien und den Umgang mit algebraischen Strukturen unabdingbar. GPT-4 kann in seiner aktuellen Version zwar als Ideengeber dienen und eine Struktur vorgeben. Die letztendliche mathematische Tätigkeit und die Beurteilung der Ausgabe auf ihre Richtigkeit hin verbleiben jedoch bei den Anwenderinnen und Anwendern.

### Empfehlungen

Für die Verwendung großer generativer KI-Sprachmodelle wie ChatGPT existieren verschiedene trainingsbasierte Prompt-Techniken, die die Leistungsfähigkeit des Modells bei mathematischen Problemlöseaufgaben verbessern können. Dabei können Aufgaben sowohl direkt gestellt (Zero-shot-Szenario) werden, als auch mit Hilfe eines oder mehrerer Lösungsbeispiele (Few-shot-Szenario), was eine gewünschte Ausgabeform der Lösung provoziert. Die Prompts sollten die KI-Sprachmodelle zur Generierung einer logischen Gedankenkette (Chain-of-Thought) anregen, um deutlich bessere

Ergebnisse zu erzielen und „Denkfehler“ offenzulegen. Durch die Technik Query-Loop, welche dem Modell das Stellen von Fragen ermöglicht, können Herausforderungen für das Modell identifiziert werden.

Diese Schritt-für-Schritt-Anleitung skizziert unseren Prozess der effektiven Umsetzung der vorgestellten trainingsbasierten Prompt-Techniken (vgl. Ramlochan, 2023):

1. *Zero-shot-Szenario*: Formulieren Sie das Problem in einer logischen und sprachlich klaren Abfolge. Zerlegen Sie das Problem, falls notwendig, in Teilprobleme.
2. *Zero-shot-Szenario, Chain-of-Thought*: Regen Sie das KI-Sprachmodell zur Strukturierung der Ausgabe mit dem Zusatz „Lass uns Schritt für Schritt vorgehen.“ an.
3. *Zero-shot-Szenario, Chain-of-Thought, Query-Loop*: Ermöglichen Sie dem Modell mit dem Zusatz „Stelle notwendige Fragen, die Du zur Beantwortung der Frage benötigst.“ eine Rückmeldung zu geben.
4. *Few-shot-Szenario, Chain-of-Thought*: Konzipieren Sie mehrere Prompts mit kurzen Lösungsbeispielen, die die erforderliche Denkstruktur bei vergleichbaren Herausforderungen darstellen. Nutzen Sie in Ihren Lösungsbeispielen eine nachvollziehbare und logische Gedankenkette, um das Modell bei einer Chain-of-Thought zu unterstützen. Geben Sie das Lösungsbeispiel zusammen mit dem eigenen Problem ein.

Mit dieser Abfolge an Strategien konnten in den beiden aufgeführten Problemlöseaufgaben deutlich Verbesserungen der Leistungen des generativen KI-Sprachmodells GPT-4 erzielt werden. Inwiefern diese Prompt-Techniken auch auf zukünftige Modelle anwendbar sind, werden die weiteren technischen Entwicklungen zeigen müssen. Ebenso sind die Entwicklungen bei den kürzlich eingeführten Plugins (bspw. Wolfram) zu berücksichtigen. Es scheint uns allerdings grundlegend zu sein, diese Techniken zu beschreiben, um die Modelle im schulischen Bereich produktiv einzusetzen und die Qualität der Ausgaben der KI-Sprachmodelle insbesondere im mathematischen Bereich zu verbessern. Die Techniken könnten dann für Entwicklungen von KI-basierten Lernumgebungen im Mathematikunterricht verwendet werden, um einen kritischen und vor allem konstruktiven Umgang mit den KI-Werkzeugen der Zukunft zu fördern.

### Literatur

Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatta, K., Chami, I., & Re, C. (2023). Ask me anything: A simple strategy for prompting language models.



- Paper presented at ICLR 2023. [openreview.net/pdf?id=bhUPJnS2goX](https://openreview.net/pdf?id=bhUPJnS2goX)
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., & Amodei, D. (2020). Language models are few-shot learners. *NeurIPS*.
- Buchholtz, N., Baumanns, L., Huget, J., Peters, F., Schorcht, S. & Pohl, M. (2023). Herausforderungen und Entwicklungsmöglichkeiten für die Mathematikdidaktik durch generative KI-Sprachmodelle. *Mitteilungen der Gesellschaft für Didaktik der Mathematik*, 114, 19–26.
- Cherian, A., Peng, K.-C., Lohit, S., Smith, K., & Tenenbaum, J. B. (2022). Are deep neural networks SMARTer than second graders? DOI:10.48550/arXiv.2212.09993
- Dauscher, P. (2019). Künstliche Intelligenz selbst programmiert. *MNU Journal*, 6, 472–477.
- Drori, I., Zhang, S., Shuttleworth, R., Tang, L., Lu, A., Ke, E., Liu, K., Chen, L., Tran, S., Cheng, N., Wang, R., Singh, N., Patti, T. L., Lynch, J., Shporer, A., Verma, N., Wu, E., & Strang, G. (2022). A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32), 1–181.
- Floridi, L. & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds & Machines*, 30, 681–694.
- Herm, L.-V., Wanner, J., Seubert, F., & Janiesch, C. (2021). I don't get it, but it seems valid! The connection between explainability and comprehensibility in (X)AI research. In *European Conference on Information Systems (ECIS)* (S. 1).
- Kojima, T., Shane Gu, S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large Language models are zero-shot reasoners. [arXiv:2205.11916v1.pdf#page=1](https://arxiv.org/abs/2205.11916v1.pdf#page=1)
- Lample, G., & Charton, F. (2019). Deep learning for symbolic mathematics. DOI:10.48550/arXiv.1912.01412
- Larsen, M., & Weßels, D. (2022): Chain of Thought Prompting: KI-Transfer Hub SH. <https://kuenstliche-intelligenz.sh/de/chain-of-thought-prompting>
- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2021): Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. [arXiv:2107.13586.pdf](https://arxiv.org/abs/2107.13586.pdf)
- Lutz-Westphal, B. (2023). ChatGPT und der „Faktor Mensch“ im schulischen Mathematikunterricht. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 31(1), 19–21.
- Oldenburg, R. (2023). Künstliche und natürliche Intelligenz: Zur Gestaltung einer zukünftigen Bildung. *Forschung & Lehre*, 23(5), S. 346–347.
- OpenAI (2023). GPT-4 Technical Report. [arXiv:2303.08774.pdf](https://arxiv.org/abs/2303.08774.pdf)
- Ott, B. (2018). Mathematische Strukturen in Textaufgaben und grafischen Darstellungen. In Fachgruppe Didaktik der Mathematik der Universität Paderborn (Hrsg.), *Beiträge zum Mathematikunterricht 2018* (S. 1363–1366). WTM.
- Paaß, G. (2022). Sprachverstehen: GPT-3 & Co. texten überzeugend, aber nicht faktentreu. *c't*, 9, 65.
- Ramlochan, S. (2023). Master prompting concepts: Chain of thought prompting. [www.promptengineering.org/master-prompting-concepts-chain-of-thought-prompting/#introduction-to-chain-of-thought-cot-prompting](https://www.promptengineering.org/master-prompting-concepts-chain-of-thought-prompting/#introduction-to-chain-of-thought-cot-prompting)
- Renkl, A. (2002). Worked-out examples: Instructional explanations support learning by self-explanations. *Learning and Instruction*, 12(5), 529–556.
- Schönthaler, P. (2023). Schneller als gedacht: ChatGPT zwischen wirtschaftlicher Effizienz und menschlichem Wunschdenken. *c't*, 9, 126–131.
- Schorcht, S. (2022). Changes and rigors in systems of mathematical representations within gifted children's problem-solving process. Twelfth Congress of the European Society for Research in Mathematics Education (CERME12), Feb. 2022, Bolzano, Italy. hal-03765552
- Strecker, K. (2020). Ein kleines Neuronales Netz selbst programmieren. *MNU Journal*, 2, 92–96.
- Wei, J., Zhou, D. (2022): Language Models Perform Reasoning via Chain of Thought: Google AI Blog. [ai.googleblog.com/2022/05/language-models-perform-reasoning-via.html](https://ai.googleblog.com/2022/05/language-models-perform-reasoning-via.html)
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, Ed H., Le, Q. V., & Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models. [arXiv:2201.11903.pdf](https://arxiv.org/abs/2201.11903.pdf)
- Weßels, D. (2023a). ChatGPT: Ein Meilenstein der KI-Entwicklung. *Mitteilungen der Deutschen Mathematiker-Vereinigung*, 31(1), 17–19.
- Weßels, D. (2023b). Was ist ChatGPT und wie funktioniert es? Und welche ähnlichen Tools gibt es. Vortrag in der dghd-Reihe „KI in der Hochschullehre“ am 27. Januar 2023. [youtu.be/cMuBo\\_rH15c](https://youtu.be/cMuBo_rH15c)

Sebastian Schorcht, Technische Universität Dresden  
E-Mail: [sebastian.schorcht@tu-dresden.de](mailto:sebastian.schorcht@tu-dresden.de)

Lukas Baumanns, Universität zu Köln und Pädagogische Hochschule Freiburg  
E-Mail: [lukas.baumanns@uni-koeln.de](mailto:lukas.baumanns@uni-koeln.de)

Nils Buchholtz, Universität Hamburg  
E-Mail: [nils.buchholtz@uni-hamburg.de](mailto:nils.buchholtz@uni-hamburg.de)

Judith Huget, Universität Bielefeld  
E-Mail: [jhuget@math.uni-bielefeld.de](mailto:jhuget@math.uni-bielefeld.de)

Franziska Peters, Justus-Liebig-Universität Gießen  
E-Mail: [franziska.peters@math.uni-giessen.de](mailto:franziska.peters@math.uni-giessen.de)

Maximilian Pohl, Universität Duisburg-Essen  
E-Mail: [maximilian.pohl@uni-due.de](mailto:maximilian.pohl@uni-due.de)