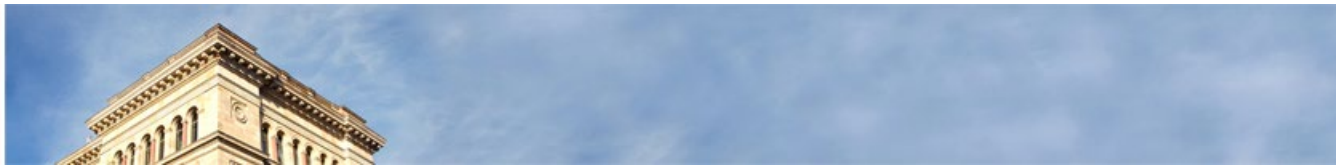




Wie sich erlernte von designer Technik unterscheidet: Die neue Agency generativer KI und das Erfordernis der strategischen Interaktion mit ihr



Overview

- (1) Warum Große Sprachmodelle (LLMs) grundlegende Fragen über Technik aufwerfen
- (2) Prompt-Techniken und Analogien zu menschlichen Strategien
- (3) Prompten als strategische Interaktion?

Original Research Article

Why generative AI is different from designed technology regarding task-relatedness, user interaction, and agency

Ingo Schulz-Schaeffer¹ 

Abstract

Generative AI requires us to rethink what constitutes technology, how technology is socially constructed, and how technology is used. Our common understanding of technology is mainly derived from designed technology. Generative AI, in contrast, is a learned technology and, moreover, a technology that mainly relies on unsupervised learning. The paper argues that this has far-reaching consequences with respect to the task-relatedness of generative AI systems, the user interaction with generative AI systems, and the agency of those systems. Designed technological behavior is usually designed with respect to particular tasks. AI systems based on unsupervised learning, in contrast, are not task-related in such a manner. Consequently, generative AI systems cannot be operated in the same way as designed technology. Rather than instructing them, operating them requires what can be described as strategic interaction. Interacting with generative AI systems leads to new actor roles and role relationships. The agency of technological artifacts that are designed and used for particular tasks tends to be the agency of a tool. In contrast, the new agency of generative AI systems lies in their capability to mobilize machine-learned versions of human experiential knowledge and thereby to become in some respects similar to a human interaction partner.

Keywords

Generative AI, agency of technology, task-relatedness, unsupervised learning, user interaction, experiential knowledge

Introduction

In this paper, I argue that generative AI is different from designed technology in three main respects: task-relatedness/toolness, user interaction, and agency. Designed technology is usually developed and put to use by establishing the artifacts as predefined means for particular ends and thus as tools. This mode of establishing technological artifacts requires the artifacts' features to be designed with respect to their task-related roles. Generative AI systems, in contrast, acquire their features by learning. To the extent that they are trained based on unsupervised learning, neither designers nor users initially know which features the systems possess and for which purposes they may turn out to be useful. In their pre-trained state, generative AI models are "task-agnostic" and lack the toolness of designed technology (Section, The task-relatedness of technological design and the task-agnosticism of unsupervised learning).

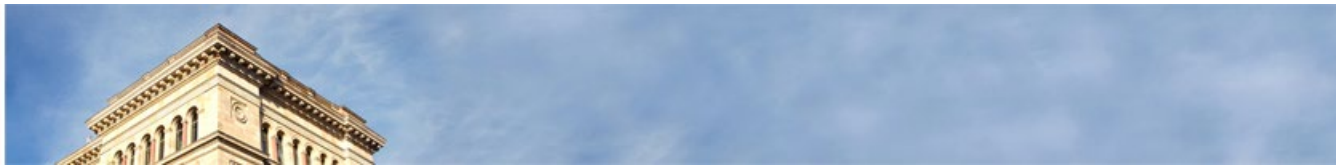
Technological artifacts are commonly used by operating and instructing them. Both activities presuppose that, to some extent, it is predefined which parts of the sequence of events that brings about the intended result are to be performed by the artifacts and which parts the users or other parties involved have to contribute. Operating a technological artifact means that the users have to perform their part of the task, and thereby enable the artifacts to play their part. Instructing refers to triggering the technology's contributions

¹Kate Hamburger Center for Advanced Study in the Humanities: Cultures of Research (KHK co/re), RWTH Aachen University, Aachen, Germany

Corresponding author:
Ingo Schulz-Schaeffer, Kate Hamburger Center for Advanced Study in the Humanities: Cultures of Research (KHK co/re), RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany. schulz-schaeffer@tu-berlin.de



Creative Commons Attribution-NonCommercial-NoDerivs 4.0 International License (https://creativecommons.org/licenses/by-nc-nd/4.0/) which permits any use, reproduction and distribution of the work as published without adaptation or alteration, provided the original work is attributed as specified on the SAGE and Open Access page (https://us.sagepub.com/en-us/nam/open-access-at-sage).



Große Sprachmodelle (LLMs) werfen grundlegende Fragen über Technik auf

Große Sprachmodelle (LLMs) stellen etablierte Vorstellungen über die Eigenschaften technischer Artefakte (mechanische oder digitale Geräte, Maschinen, Computer und Software) und über deren Verwendung infrage. Dies hat folgende Gründe:

- Ihre wesentlichen Funktionseigenschaften werden **nicht designt**, d.h. nicht mit Blick auf bestimmte Probleme entwickelt, **sondern erlernt**.
- Sie werden nicht genutzt, indem man sie **bedient oder Befehle gibt**, sondern indem man mit ihnen **interagiert**.
- In ihrem vortrainierten Zustand sind ihre Funktionseigenschaften **nicht aufgabenspezifisch**, sondern **aufgabenunabhängig** „(task agnostic)“.
- Zusammenfassend lässt sich sagen: Sie sind **keine Werkzeuge**.

Disclaimer: In diesem Vortrag geht es nicht um Biases von LLMs oder um deren grundlegende Undurchsichtigkeit und auch nicht um Auswirkungen generativer KI auf das Arbeitsleben – also nicht um die Fragen, mit denen sich die sozialwissenschaftliche Forschung bislang vorwiegend befasst hat.



Der Hintergrund, vor dem LLMs und deren Nutzung hervorstechen

Unser Verständnis von Technik und deren Nutzung basiert auf Artefakten, die entwickelt wurden bzw. entstanden sind, um als **Mittel für bestimmte Zwecke** zu dienen.

Solche Artefakte lassen sich als materielle Anordnungen charakterisieren, die **vorgefertigte Lösungen für vordefinierte Probleme** bieten. (Dies ist offensichtlich bei Techniken mit einem einzigen Zweck, wie beispielsweise einem Dokumentenscanner. Es gilt jedoch auch für vielfältig einsetzbare Techniken. Ein Beispiel hierfür ist der Elektromotor.)

Alle diese Artefakte sind **aufgabenbezogen**.

Die Nutzung dieser Art von Technologie erfordert zwei Arten von Wissen: Wissen darüber, welche Probleme das Artefakt lösen kann, und Wissen darüber, wie man den Beitrag des Artefakts zur Lösung hervorruft. Darum geht es wenn man mechanische Techniken **bedient** und digitalen Techniken **Befehle gibt**.

Zusammenfassend lässt sich sagen: Unser allgemeines Verständnis von Technik leitet sich von Artefakten ab, die als **Werkzeuge** dienen.

Der Zusammenhang von vorgefertigten Problemlösungen, Aufgabenbezogenheit und Werkzeugcharakter ist darin begründet, dass die entsprechenden Artefakte Bestandteil **geskripteter sozio-technischer Konstellationen** sind.



Warum LLMs anders sind

Erlernte Fähigkeiten statt vorgegebener Funktionen: LLMs werden unüberwacht vortrainiert. Ziel des Trainings ist es, alle Muster zu finden, die in den Trainingsdaten zu finden sind (Alpaydin, 2016: 111).

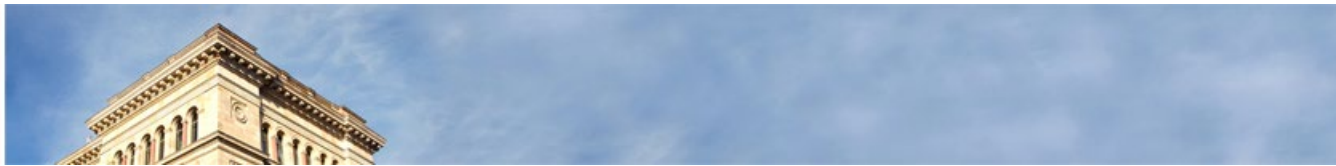
Die erlernten Muster vortrainierter LLMs sind keine vorgefertigten Lösungen für Probleme. Vielmehr sind die erlernten Fähigkeiten aufgabenunabhängig. Die grundlegende Funktionsweise von LLMs lautet **“ask me anything’ and ‘I may have a good answer’”** (Dwivedi et al., 2023).

Aufgabenspezifität und Werkzeugcharakter sind die Voraussetzungen dafür, dass wir technische Artefakte **bedienen oder ihnen Befehle geben** können: Sie zu bedienen oder ihnen Befehle zu geben bedeutet, die vorgefertigten Handlungsbeiträge der Artefakte abzurufen. Aus diesem Grund kann man vortrainierte LLMs nicht auf die übliche Weise bedienen oder ihnen Befehle geben.

Wie kann man dann LLMs dazu bringen, zur Lösung von Problemen beizutragen?

- (1) Aufgabenspezifisches Fine-Tuning (Fine-Tuning mit gelabelten Datensätzen und Reinforcement Learning from Human Feedback (RLHF))
- (2) Prompten und Prompt Engineering -> **Prompten als strategische Interaktion mit LLMs?**

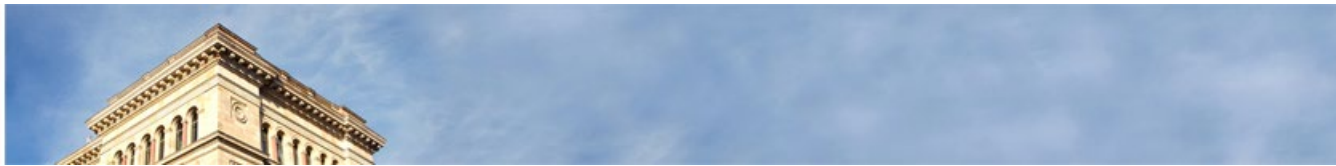
Vortrainierte LLMs repräsentieren **eine nicht geskripte Technologie**. Fine-Tuning und Prompten lassen sich als Formen des Skriptens interpretieren.



Prompt Engineering

“Prompt-engineering is the process of designing *natural language specifications of a task*, which are used to condition the LLM at inference time.” (Arora et al., 2022: 3).

“Now, as of this writing in 2021, we are in the middle of a second sea change, in which *the ‘pretrain, fine-tune’ procedure is replaced by one in which we dub ‘pre-train, prompt, and predict.’* In this paradigm, instead of adapting pre-trained LMs to downstream tasks via objective engineering, downstream tasks are reformulated to look more like those solved during the original LM training with the help of a textual prompt. For example, [...] if we choose the prompt “English: I missed the bus today. French: ”), then an LM may be able to fill in the blank with a French translation. In this way, by selecting the appropriate prompts we can manipulate the model behavior so that the pre-trained LM itself can be used to predict the desired output, sometimes even without any additional task-specific training [...]. The advantage of this method is that, given a suite of appropriate prompts, *a single LM trained in an entirely unsupervised fashion can be used to solve a great number of tasks.*” (Liu et al., 2023: 195:192)



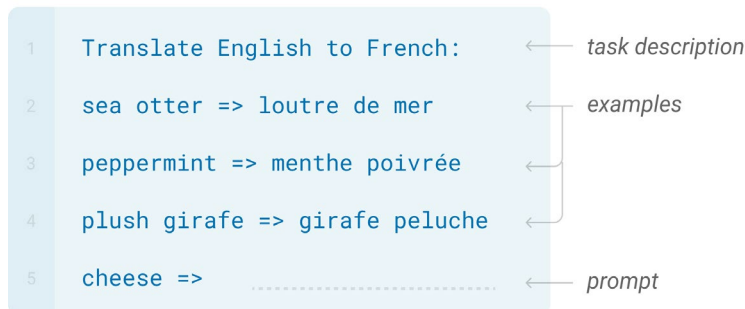
Prompt-Techniken: Grundlagen

1. Gib klare und spezifische Anweisungen.
2. Verwende explizite Vorgaben (Format, Länge, Umfang).
3. Stelle den Kontext und Beispiele bereit.
4. Berücksichtige den Unterschied zwischen Fragen, die direkt beantwortet werden können, und Fragen, die eine bewusstere, analytischere oder komplexere Problemlösung erfordern.
5. Steuere die Ausführlichkeit der Ausgabe (z. B. kurze oder detaillierte Antworten). (Ekin 2023: 3-6)

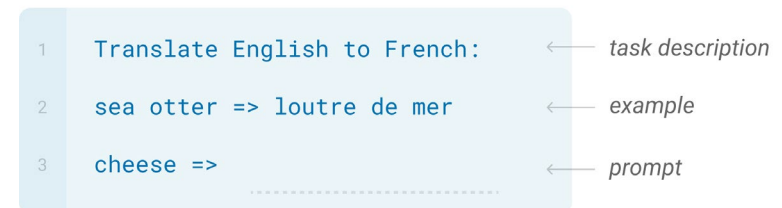
Prompt-Techniken: Few-shot Prompting

“Few-Shot (FS) is the term we will use in this work to refer to the setting where **the model is given a few demonstrations of the task** at inference time as conditioning [...]. As shown in Figure 2.1, for a typical dataset an example has a context and a desired completion (for example an English sentence and the French translation), and few-shot works by giving K examples of context and completion, and then one final example of context, with the model expected to provide the completion.” (Brown et al., 2020: 2, 7)

Few-shot



One-shot



“Few-shot prompting [Brown et al. 2020] refers to the idea that **language models do not need to be specifically trained for a downstream task** (e.g. classification, question answering, etc.). Rather, it is sufficient to train them on broad text-sequence prediction datasets [...] and to provide context in the form of examples when invoking them.” (Beurer-Kellner et al., 2023: 186:4–5)

Prompt-Techniken: Chain-of-Thought Prompting (CoT)

“A *chain of thought* is a series of **intermediate natural language reasoning steps** that lead to the final output [...] Our proposed approach is to augment each exemplar in few-shot prompting with a chain of thought for an associated answer” (Wei et al., 2023: 2-3)

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

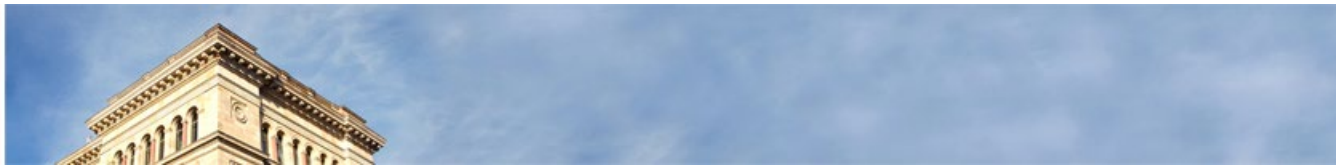
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

“chain-of-thought prompting via GPT-3 175B and PaLM 540B **compares favorably to prior state of the art**, which typically fine-tunes a task-specific model on a labeled training dataset.” (Wei et al., 2023: 5)



Prompt-Techniken: Zero-shot Chain-of-Thought Prompting (Zero-shot CoT)

“We propose Zero-shot-CoT, a zero-shot template-based prompting for chain of thought reasoning. It differs from the original chain of thought prompting [Wei et al., 2022] as it does not require step-by-step few-shot examples, and it differs from most of the prior template prompting [Liu et al., 2021b] as it is inherently task-agnostic [...]. The core idea of our method is simple [...]: **add Let's think step by step** [...] to extract step-by-step reasoning.” (Kojima et al., 2023: 3)

Standard Zero-shot:

Question: One of the Beatles was murdered. Which LP published the band 11 years earlier?

ChatGPT 4.0:

John Lennon, a member of The Beatles, was murdered on December 8, 1980. Eleven years earlier, in 1969, The Beatles released their album "Abbey Road" in September of that year. This LP was among their final works as a band before their breakup in 1970.

ChatGPT 5.2 Auto (Instant und Thinking fast identisch):

The murdered Beatle was John Lennon (killed in 1980). 11 years earlier, in 1969, the Beatles released the LP Abbey Road.

Zero-shot CoT:

Question: One of the Beatles was murdered. Which LP published the band 11 years earlier? Think step by step.

ChatGPT 4.0:

Let's break this down step by step:

1. Who was murdered?

John Lennon, a member of the Beatles, was tragically murdered on December 8, 1980.

2. What happened 11 years earlier?

Subtracting 11 years from 1980 takes us to 1969.

3. Which Beatles LP was released in 1969?

The Beatles released two LPs in 1969:

- "Yellow Submarine" in January 1969.
- "Abbey Road" in September 1969.

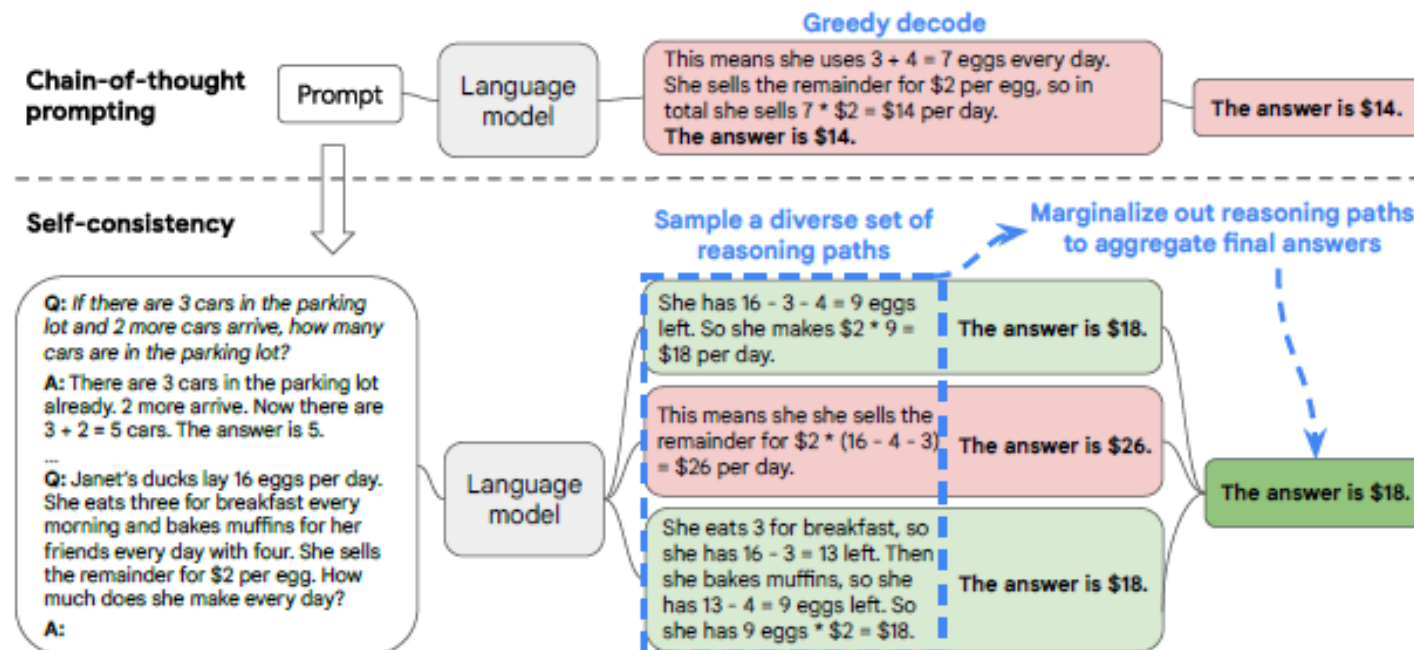
4. Conclusion:

If the event being referenced aligns with 11 years before John's death in December 1980, the relevant LP would likely be "Abbey Road" (September 1969).



Prompt-Techniken: Self-Consistency Prompting

“First, a language model is prompted with a set of manually written chain-of-thought exemplars (Wei et al., 2022). Next, we sample a set of candidate outputs from the language model’s decoder, **generating a diverse set of candidate reasoning paths**. [...]. Finally, we aggregate the answers by marginalizing out the sampled reasoning paths and **choosing the answer that is the most consistent** among the generated answers.” (Wang et al., 2023: 2–3)





Prompt-Techniken: Role Prompting

“Modern LLMs, with their **advanced role-playing capabilities** [...] can convincingly mimic various personas, ranging from fictional characters to historical and contemporary figures. The **assigned role provides context** about the LLM’s identity and background. By adopting the persona, the LLM can generate more natural, in-character responses tailored to that role. [...]

Beyond conversational applications, **role-playing also boosts LLM performance** on certain NLP tasks. For instance, when cast as a judge with a distinctive role, LLMs can effectively evaluate the quality of text summarization (Wu et al., 2023). More unconventionally, Chat-GPT demonstrates competency in processing Linux commands when prompted as a Linux terminal.” (Kong et al., 2024: 1-2)

Q: You are a primary school teacher. Explain to second graders what “or” means.

ChatGPT 4.0:

A: Hello, friends! Today, we're going to talk about a very small but very important word: "or". Can you all say "or"? (Pause for response.)

Great! Now, "or" is a word we use when we have to make a choice between two things. It's like when someone asks you, "Do you want an apple or a banana?" You can choose one of them, but not both.

Let's try a few fun examples: ...

Q: You are a philosophy professor specialized in formal logic. Explain to your students what “or” means.

ChatGPT 4.0:

A: In formal logic, the term "or" corresponds to the logical disjunction operator, denoted by the symbol \vee . It's crucial to understand what this means in logical terms, as it might differ slightly from how "or" is used in everyday language.

Inclusive vs. Exclusive "Or"

In formal logic:

- The logical "or" (\vee) is inclusive, meaning it is true if at least one of the statements it connects is true.
- In contrast, the "or" we sometimes use in ordinary speech can be exclusive, suggesting that only one of the statements is true, but not both (e.g., "You can have tea or coffee, but not both"). Exclusive "or" is a different operator in logic, sometimes written as \oplus or XOR. ...



Prompt-Techniken: Meta-Prompting (I)

Diese Prompt-Technik kombiniert das Zerlegen von Aufgaben aus dem Chain-of-Thought Prompting mit dem Role Prompting:

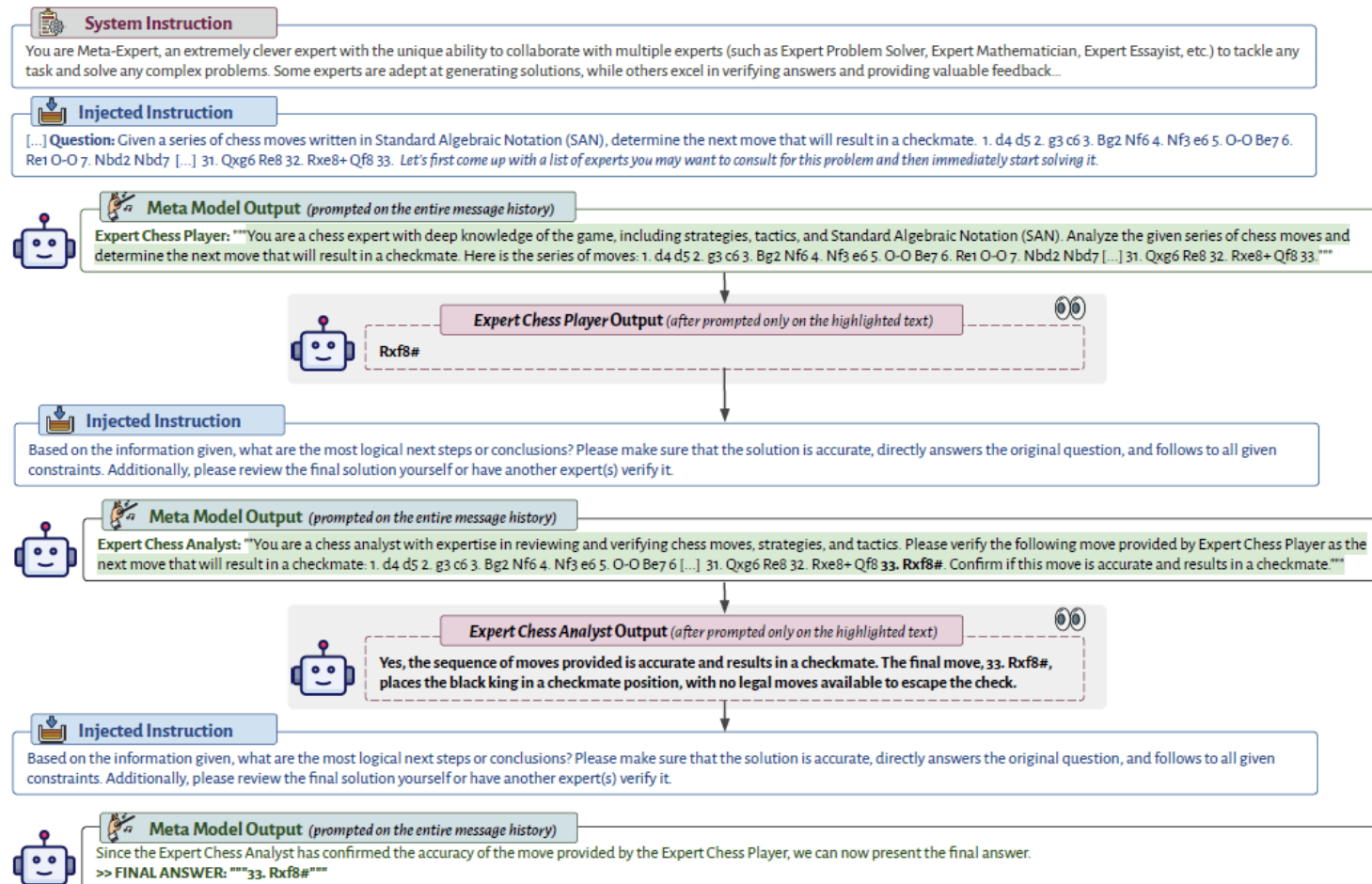
“This approach transforms a single LM into a multi-faceted conductor, adept at managing and integrating multiple independent LM queries. By employing high-level instructions, meta-prompting guides the LM to ***break down complex tasks into smaller, more manageable subtasks***.

These subtasks are then ***handled by distinct “expert” instances of the same LM***, each operating under specific, tailored instructions. Central to this process is the LM itself, in its role as the conductor, which ensures seamless communication and effective integration of the outputs from these expert models. It additionally employs its inherent critical thinking and robust verification processes to refine and authenticate the end result.

This ***collaborative prompting approach*** empowers a single LM to simultaneously act as a comprehensive ***orchestrator and a panel of diverse experts***, significantly enhancing its performance across a wide array of tasks.

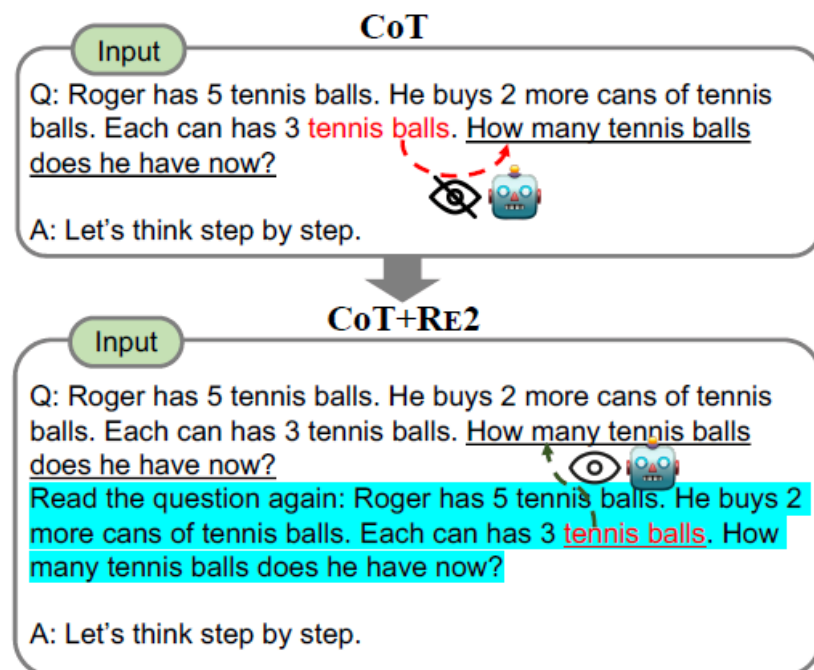
The ***zero-shot, task-agnostic*** nature of meta-prompting greatly simplifies user interaction by obviating the need for detailed, task-specific instructions. Furthermore, our research demonstrates the seamless integration of external tools, such as a Python interpreter, into the meta-prompting framework, thereby broadening its applicability and utility.” (Suzgun and Kalai, 2024: 1)

Prompt-Techniken: Meta-Prompting (II)



Prompt-Technik: Re-Reading

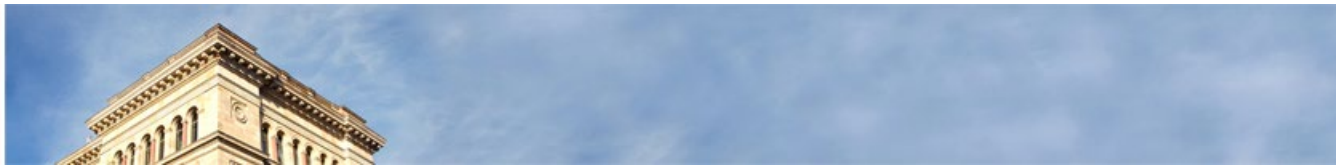
“we introduce a simple, yet general and effective prompting method, RE2, i.e., Re-Reading the question as input. [...] RE2 shifts the focus to the input by processing questions twice, thereby enhancing the understanding process. [...] Crucially, RE2 facilitates a "bidirectional" encoding in unidirectional decoder-only LLMs because the first pass could provide global information for the second pass.”(Xu et al., 2024: 1)





Analogien zu menschlichen Strategien und Verfahrensweisen

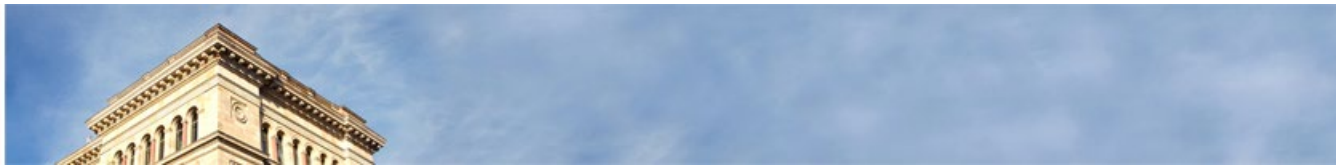
Die Autoren aller eben beschriebenen Prompt-Techniken sagen oder implizieren, dass sie sich von entsprechenden menschlichen Techniken und Strategien zur Problemlösung und zur Erläuterung von Aufgaben gegenüber anderen Menschen inspirieren ließen.



Analogien zu menschlichen Strategien: Grundlagen des Prompts

Grundlegende Strategien des Anweisungen-Gebens:

1. Gib klare und spezifische Anweisungen.
2. Verwende explizite Vorgaben (Format, Länge, Umfang).
3. Stelle den Kontext und Beispiele bereit.
4. Berücksichtige den Unterschied zwischen Fragen, die direkt beantwortet werden können, und Fragen, die eine bewusstere, analytischere oder komplexere Problemlösung erfordern.
5. Steuere die Ausführlichkeit der Ausgabe (z. B. kurze oder detaillierte Antworten). (Ekin 2023: 3-6)



Analogien zu menschlichen Strategien: Few-shot and One-shot Prompting

Beispielhafte Erläuterungen:

“*humans do not require large supervised datasets* to learn most language tasks – a brief directive in natural language (e.g. “please tell me if this sentence describes something happy or something sad”) or at most *a tiny number of demonstrations* (e.g. “here are two examples of people acting brave; please give a third example of bravery”) is often sufficient to enable a human to perform a new task to at least a reasonable degree of competence. [...] To be broadly useful, we would someday like our NLP systems to have this same fluidity and generality.” (Brown et al., 2020: 3–4)

“One-Shot (1S) [...] most closely matches *the way in which some tasks are communicated to humans*. For example, when asking humans to generate a dataset on a human worker service (for example Mechanical Turk), it is common to give one demonstration of the task. By contrast it is sometimes difficult to communicate the content or format of a task if no examples are given.” (Brown et al., 2020: 6)



Analogien zu menschlichen Strategien: Chain-of-Thought Prompting (CoT)

Nachahmung des menschlichen Denkprozesses:

“chain of thought *emulates the thought processes of human reasoners*” (Wei et al., 2023)

“Wei et al. (2022) have proposed chain-of-thought prompting, where a language model is prompted to generate a series of short sentences that *mimic the reasoning process a person might employ* in solving a task.” (Wang et al., 2023)

“Explanation-based prompting is *intuitively motivated by the reasoning steps humans typically employ* to solve a problem (Hausmann and Van Lehn, 2007).” (Jung et al., 2022)

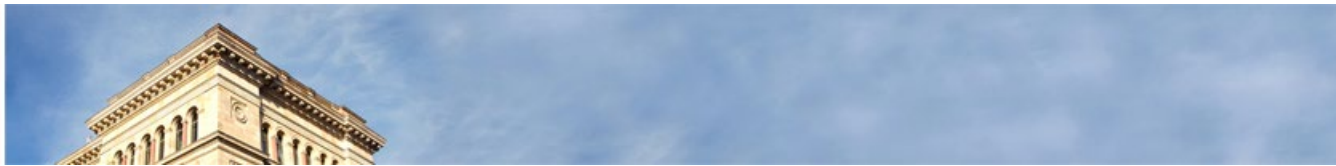


Analogien zu menschlichen Strategien: Self-Consistency Prompting

Mehrere Problemlösungswege verfolgen und vergleichen:

A *salient aspect of humanity is that people think differently*. It is natural to suppose that in tasks requiring deliberate thinking, there are likely several ways to attack the problem. We propose that such a process can be simulated in language models via sampling from the language model's decoder. (Wang et al., 2023: 2)

Generating and pursuing a diverse set of reasoning paths “is analogous to the *human experience that if multiple different ways of thinking lead to the same answer, one has greater confidence* that the final answer is correct..” (Wang et al., 2023: 1)

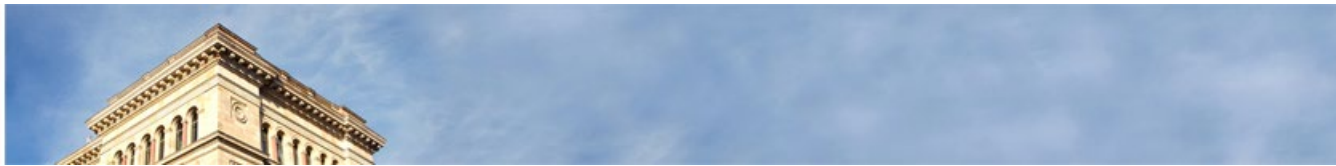


Analogien zu menschlichen Strategien: Meta-Prompting

Mit neuen Augen betrachten:

Meta-prompting: The conductor invokes new experts until the problem is solved.

“concept of fresh eyes [...] Grounded in *principles from cognitive psychology, fresh perspectives can lead to more creative problem-solving and error detection*. When individuals or models approach a problem without preconceived notions, they are more likely to consider alternative solutions and identify errors that might have been overlooked. Fresh eyes may help avoid cognitive biases such as anchoring, confirmation bias, as well as overconfidence.” (Suzgun and Kalai, 2024)



Analogien zu menschlichen Strategien: Re-Reading

Besseres Verstehen durch Wiederholung:

“many cognitive science studies have revealed that humans tend to re-read questions during learning and problem-solving to enhance the comprehension process (Dowhower, 1987, 1989; Ozek and Civelek, 2006). [...] Based on the observation *and inspired by the human strategy of re-reading*, we present a simple yet effective and general reasoning prompting strategy, RE2, i.e., Re-Reading the question as input” (Xu et al., 2024)



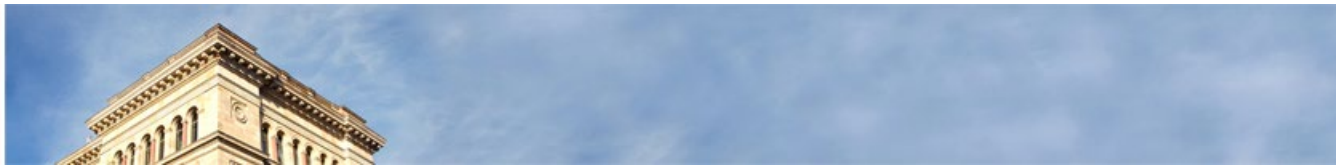
Instrumentelles, strategisches und kommunikatives Handeln

Habermas charakterisiert strategisches Handeln als eine von zwei Formen zweckrationalen Handelns, wobei **zweckrationales Handeln** erfolgsorientiertes Handeln ist, das darauf abzielt, gewünschte Ziele durch die Auswahl geeigneter Mittel zu erreichen.

„Eine erfolgsorientierte Handlung nennen wir **instrumentell**, wenn wir sie unter dem Aspekt der Befolgung technischer Handlungsregeln betrachten und den Wirkungsgrad einer Intervention in einem Zusammenhang von Zuständen und Ereignissen bewerten;

strategisch nennen wir eine erfolgsorientierte Handlung, wenn wir sie unter dem Aspekt der Befolgung von Regeln rationaler Wahl betrachten und den Wirkungsgrad der Einflußnahme auf die Entscheidungen eines rationalen Gegenspielers bewerten. [...]

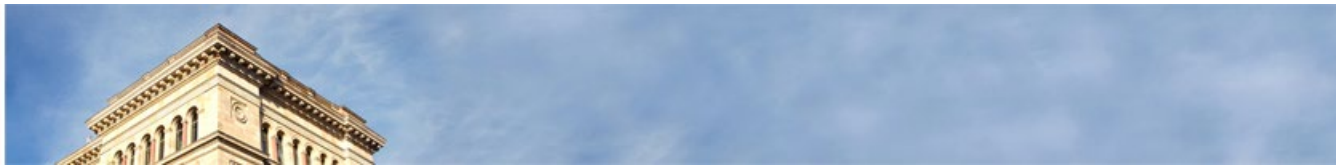
Hingegen spreche ich von **kommunikativen Handlungen**, wenn die Handlungspläne der beteiligten Akteure nicht über egozentrische Erfolgskalküle, sondern über Akte der Verständigung koordiniert werden. Im kommunikativen Handeln sind die Beteiligten nicht primär am eigenen Erfolg orientiert; sie verfolgen ihre individuellen Ziele unter der Bedingung, daß sie ihre Handlungspläne auf der Grundlage gemeinsamer Situationsdeutungen aufeinander abstimmen können.“ (Habermas 1981, Bd.1: 384-5.)



Prompten als instrumentelles, strategisches und kommunikatives Handeln?

Vier mögliche Interpretationen dessen, was Prompten ist, basierend auf Habermas' Unterscheidung:

- (1) Prompten ist instrumentelles Handeln. Es geht darum, technische Regeln zu erforschen und anzuwenden, die die beabsichtigten Reaktionen von LLMs hervorrufen.
- (2) Prompten ist strategisches Handeln. Es geht darum, LLMs auf effiziente Weise so zu beeinflussen, dass sie den eigenen Interessen entsprechen.
- (3) Prompten ist kommunikatives Handeln. Es geht sich auf der Grundlage gemeinsamer Situationsdefinitionen mit der LLM über die Problemlösung zu einigen.
- (4) Prompten ist etwas anderes.



Ist Prompten instrumentelles Handeln?

Einige Forscher:innen betrachten Prompten als eine Art des Programmierens, des Befehle-Gebens:

“Prompt engineering for a language model whose input and output are in natural language may be **conceived as programming in natural language**.” (Reynolds und McDonell 2021, S. 4)

“**Prompts are instructions** given to an LLM to enforce rules, automate processes, and ensure specific qualities (and quantities) of generated output. Prompts are also a form of programming that can customize the outputs and interactions with an LLM.” (White et al. 2023)

„Programmieren“ und „Befehle geben“ sind in diesem Zusammenhang irreführende Begriffe. Diese „Programme“ oder „Befehle“ in natürlicher Sprache sind keine Auslöser für das LLM. Sie lösen nicht automatisch die beabsichtigte Reaktion des Artefakts aus. Vielmehr behandeln LLMs sie wie jede andere Eingabe, für die sie nach passenden Mustern in ihren gelernten Mustern suchen.

Das Prompten ähnelt eher den **Anweisungen an menschliche Akteure** als der Art und Weise, wie wir Maschinen Befehle geben. Es handelt sich um eine Art des Anweisens, die das „Verständnis“ der Anweisung durch den Gegenüber berücksichtigen muss.

-> keine instrumentelle Handlung im Sinne von Habermas (technische Regeln werden einfach befolgt – kein Verständnis erforderlich)



Ist Prompten kommunikatives Handeln?

Beim kommunikativen Handeln verfolgen die Akteure „ihre individuellen Ziele unter der Bedingung, daß sie ihre Handlungspläne auf der Grundlage gemeinsamer Situationsdeutungen aufeinander abstimmen können.“ (Habermas 1981, Bd.1: 385.)

Eine **Situation** ist eine Konstellation von Faktoren, die als Kontext bestimmter Handlungen wahrgenommen wird. Die Wahrnehmung als Kontext basiert auf der Bedeutung der Konstellation von Faktoren für diese Handlungen. Die Vorstellungen über solche Handlungskontexte werden in der Soziologie als **Definitionen der Situation** bezeichnet. Sie existieren oft in Form von **geteilten Sinnmustern** im gesellschaftlichen Wissensbestand. (Schulz-Schaeffer, 2024: 459–463)

Beim Prompt Engineering geht es darum, Definitionen der Situation zu finden, die die Nutzer:in mit dem LLM „teilt“. Prompten ist erfolgreich, wenn es sich auf Definitionen der Situation bezieht, die den vom LLM gelernten Mustern ähneln.

Es macht jedoch keinen Sinn zu behaupten, dass Prompter und LLMs „ihre Handlungspläne auf der Grundlage gemeinsamer Situationsdefinitionen aufeinander abstimmen“ würden.

-> keine kommunikative Handlung im Sinne von Habermas, aber einige Aspekte davon



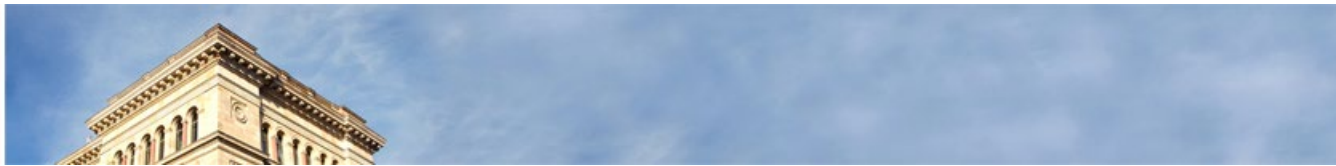
Ist Prompten strategisches Handeln?

Bei Prompt-Strategien geht es eindeutig um **effiziente Wege, das LLM zu beeinflussen** und zwar so, dass das Verhalten der LLMs den Interessen der Nutzer:innen entspricht. In dieser Hinsicht weisen sie Merkmale des strategischen Handelns nach Habermas auf.

Wie beim strategischen Handeln nach Habermas besteht das zugrunde liegende Problem darin, dass das Gegenüber – in diesem Fall das LLM – einer Aufforderung nicht freiwillig nachkommt, sondern dazu erst gebracht werden muss.

Im Gegensatz zu Habermas' strategischem Handeln sind LLMs keine rationalen Gegenspieler (z. B. können sie nicht durch positive oder negative Anreize beeinflusst werden).

Bei den Prompt-Strategien zur Beeinflussung von LLMs geht es vielmehr darum, „geteilte“ Situationsdefinitionen auszunutzen, d.h. **Prompt-Muster zu finden, die zu den erlernten Mustern der LLMs passen.**

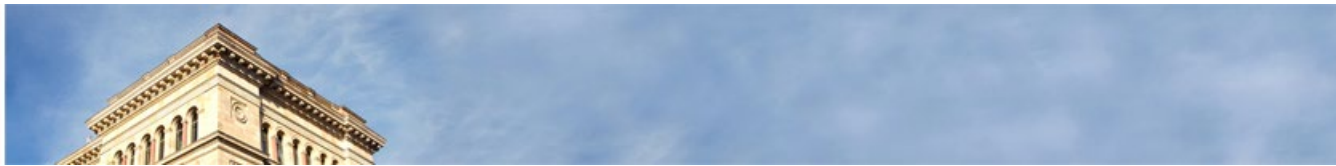


Prompten ist etwas anderes

Instrumenteller Aspekt: Prompt-Strategien ähneln eher der Anweisung an einen menschlichen Akteur als der Art und Weise, wie wir normalerweise Maschinen Befehlen geben. Prompten löst nicht einfach die beabsichtigte Reaktion aus, sondern muss das „Verständnis“ der Anweisung durch den Gegenüber berücksichtigen. Dies geschieht jedoch aus zweckrationalen Gründen.

Strategischer Aspekt: Prompt-Strategien zielen darauf ab, LLMs strategisch so zu beeinflussen, dass ihre Outputs den Zielen und Interessen der Nutzer:innen entsprechen.

Kommunikativer Aspekt: Prompten beeinflusst LLMs strategisch, indem es „geteilte“ Situationsdefinitionen ausnutzt, d. h. indem es Prompt-Muster findet und verwendet, die zu den gelernten Mustern der LLMs passen.



Prompten ist etwas anderes

Instrumenteller Aspekt: Prompt-Strategien ähneln eher der Anweisung an einen menschlichen Akteur als der Art und Weise, wie wir normalerweise Maschinen Befehlen geben. Prompten löst nicht einfach die beabsichtigte Reaktion aus, sondern muss das „Verständnis“ der Anweisung durch den Gegenüber berücksichtigen. Dies geschieht jedoch aus zweckrationalen Gründen.

Strategischer Aspekt: Prompt-Strategien zielen darauf ab, LLMs strategisch so zu beeinflussen, dass ihre Outputs den Zielen und Interessen der Nutzer:innen entsprechen.

Kommunikativer Aspekt: Prompten beeinflusst LLMs strategisch, indem es „geteilte“ Situationsdefinitionen ausnutzt, d. h. indem es Prompt-Muster findet und verwendet, die zu den gelernten Mustern der LLMs passen.