5. Oktober 2026

Wie promptet man richtig?

Systematisches **Prompt Engineering für Unternehmen**:
Von Heuristiken zu deterministischen Schnittstellen

Dr. Georg Hackenberg

Professor für Informatik & Industriesysteme

FH Oberösterreich · Campus Wels

https://hackenberg.tech

Der Fahrplan zum systematischen Prompting

Vom stochastischen Zufallstreffer zur **deterministischen Produktions-Schnittstelle**

1 **Stochastisches Fundament**
Autoregression, Vektorräume und **wie Sprachmodelle Tokens erzeugen**.

2 **Empirische Evidenz**
Wissensarbeit im Härtetest: Die **Harvard-Studie zur Jagged Frontier**.

3 **Mythen & Akronyme**
Warum Intuition trügt: Wir **prüfen Akronym-Frameworks** kritisch.

4 **Reasoning-Taxonomie**
Wissenschaftliche Methoden von **Few-Shot-Learning bis zu ReAct**.

5 **Enterprise Architecture**
Industrielle Standards mit **XML-Tags und JSON-Schemas**.

6 **Sicherheit & Evals**
Defensive Schutzschilde und **automatisierte Prompt-Evals**.

Das fundamentale Kernmissverständnis

Prompting ist kein **magisches Orakel-Voodoo**, sondern Spezifikation an stochastischen Schnittstellen

Spezifikation statt Orakel-Magie

Ingenieurs-Perspektive

Prompting ist kein **magisches Orakel-Voodoo**, sondern **formale Spezifikation an einer stochastischen Schnittstelle.**

Das Orakel-Missverständnis

Falsche Annahme: Das Modell **versteht Intentionen wie ein Mensch** – führt zu **instabilen, unberechenbaren Einzelergebnissen.**

Mathematische Raumkontrolle

Einschränkung des Lösungsraums durch **strikte Randbedingungen und Schemas** für **testbare, deterministische Software-Prozesse.**

KAPITEL 01 / 07 · THEORIE & GRUNDLAGEN

Das stochastische Fundament

Wie Sprachmodelle Tokens berechnen und warum **statistische Wahrscheinlichkeit keine Fakten kennt**

„Was geschieht mathematisch zwischen der Eingabeaufforderung und dem ersten generierten Wort?“

Von **Next-Token-Prediction über Vektor-Embeddings** bis zu Kontextgrenzen und Sampling-Parametern.

Next-Token-Prediction & Autoregression

Der mathematische Kernprozess: **Bedingte Wahrscheinlichkeiten im Textfluss**

Der stochastische Kern

KONDITIONALE WAHRSCHEINLICHKEIT

$$P(w_t \mid w_1, \ldots, w_{t-1})$$

Autoregressiver Loop: Token für Token

Das Modell maximiert fortlaufend die Plausibilität des Folge-Tokens.

In jedem Schritt berechnet das Netz eine Verteilung über das gesamte Vokabular und wählt das nächste Token deterministisch oder stochastisch.

Konsequenzen für das Prompting

Keine Wissensdatenbank im SQL-Sinn

Ein Modell schlägt keine Fakten in einer relationalen Datenbank nach; es erzeugt **statistisch plausible Textmuster** auf Basis gelernten Parameterwissens.

Kontext als Wahrscheinlichkeitsfilter

Jedes Zeichen im Prompt fungiert als mathematischer Operator: Es verändert **die Wahrscheinlichkeitsverteilung der nächsten Tokens** und engt den Lösungsraum ein.

Fehlerfortpflanzung

Wird ein fehlerhaftes Token generiert, **konditioniert es alle nachfolgenden Schritte negativ**, da das Modell seine eigene Halluzination als Kontext weiterschreibt.

Tokenisierung entzaubert

Warum Sprachmodelle **weder Buchstaben noch Zahlen nativ verstehen**

Subwort-Zerlegung (BPE)

Byte-Pair Encoding

Häufige Wörter sind 1 Token; Fachbegriffe werden in **mehrere isolierte Silben und Fragmente zerlegt**.

Das Zahlen-Dilemma

Fragmentierte Ziffern

Zahlen wie '3.14159' oder '98421' werden oft **in ungleichmäßige Zifferngruppen zerschnitten.**

Sensitivität auf Leerzeichen

Präfix-Leerzeichen

Präfixe wie ' Apfel' und 'Apfel' erzeugen **zwei völlig unterschiedliche Token-IDs** im internen Modell-Vokabular.

Multilingualer Overhead

Deutsch vs. Englisch

Deutsche zusammengesetzte Wörter erfordern **Tokens um bis zu 50 Prozent bei gleicher Wortzahl** mehr.

Vektorraum & Embeddings

Wie Sprache in **hochdimensionale Koordinaten im Hyperraum** übersetzt wird

Geometrie der Semantik

KONTINUIERLICHER VEKTORRAUM

$$\mathbb{R}^{1536} \ldots \mathbb{R}^{4096}$$

Hochdimensionale Repräsentation

Jedes Token ist ein Vektor mit **Tausenden kontinuierlichen Dimensionen.**

KOSINUS-ÄHNLICHKEIT

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

Navigation im latenten Raum

Prompt als Richtungsvektor

Worte definieren die **Koordinaten und das semantische Gravitationsfeld** der Antwort.

Kontextuelle Verschiebung

Das Wort 'Bank' führt je nach Kontext zu einer **semantischen Verschiebung.**

Empfindlichkeit für Terminologie

Präzise Fachtermini erzeugen **wesentlich schärfere semantische Cluster** als Umgangssprache.

Der Self-Attention-Mechanismus

Wie der Transformer **Relevanzbeziehungen zwischen Tokens** dynamisch berechnet

Mathematischer Kern [Vas17]

Scaled Dot-Product Attention

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Jedes Token projiziert **Suchanfragen, Schlüssel und Werte.**

All-to-All Vernetzung

Dynamische Kreuzkorrelation über die gesamte Sequenz, indem jedes Wort simultan auf **alle vorhergehenden Kontext-Tokens blickt** und semantische Bezüge gewichtet.

Bedeutung für die Prompt-Struktur

Aufmerksamkeitslenkung

Klare Trennzeichen und Schlüsselbegriffe **verstärken die Attention-Weights signifikant.**

Verwässerungsgefahr

Überflüssiger Fülltext führt dazu, dass **essenzielle Randbedingungen an Gewicht verlieren.**

Multi-Head Attention

Parallele Köpfe erfassen gleichzeitig **Syntax, Rollen und Sachzusammenhänge** über den gesamten Prompt.

Die Stellschrauben des Samplings

Wie Temperature, Top-p und Top-k die stochastische Varianz steuern

Temperature (T)

LOGIT-SKALIERUNG

$$z_i / T$$

Stauchung der Verteilung

Ein Wert von 0.0 **erzwingt nahezu deterministisches Greedy-Decoding** für reproduzierbare Ausgaben.

Enterprise-Faustregel

Gilt für strukturierte Daten: **für Enterprise-Pipelines ist null Komma null der Pflichtstandard.**

B2B-STANDARD

$$T = 0.0$$

Top-p (Nucleus)

KUMULIERTE MASSE

$$\sum P \geq p$$

Dynamischer Cut-off

Berücksichtigt **nur die wahrscheinlichsten Tokens bis zur kumulierten Schwelle p.**

Entropie-Adaption

Der Nucleus **passt den Wortschatz flexibel an die Modell-Sicherheit an** und filtert unwahrscheinliche Ränder.

B2B-STANDARD

$$p = 0.95$$

Top-k Sampling

FESTER SCHWELLENWERT

$$|V^{(k)}| = k$$

Kandidaten-Deckelung

Begrenzt die Ziehung strikt auf **einen festen Pool der k wahrscheinlichsten Folgekandidaten.**

Safety-Filter

Dient in Modellen **als harte Sicherheitsbarriere gegen absurde Ausreißer.**

B2B-STANDARD

$$k = 40$$

Kontextfenster & Lost-in-the-Middle

Warum riesige Kontextfenster **keine Garantie für fehlerfreies Retrieval** sind

Empirischer Befund [Liu24]

„Leistung ist am höchsten, wenn relevante Information **am Anfang oder Ende des Kontexts** steht – in der Mitte bricht sie dramatisch ein.“

U-Kurven-Effekt

Dokumente in der Mitte langer Prompts erleiden eine **signifikante Verschlechterung der Extraktionsgenauigkeit**. Transformer-Aufmerksamkeitsgewichte konzentrieren sich stark auf die Randpositionen.

Trotz großer Kontextweiten sinkt die Trefferrate in der Mitte ab: Große Kontextfenster ersetzen keine semantische Vorfilterung.

Architektonische Konsequenzen

Primacy & Recency

System-Instruktionen nach oben, **konkrete Aufgabenstellung unmittelbar ans Ende** des Prompts platzieren.

Kopf-Ende-Platzierung

Rohdaten (Dokumente, Tabellen) in die Mitte packen, **eingerahmt von strikten Regeln oben und unten**, damit kritische Constraints im Aufmerksamkeitsfokus bleiben.

Vermeidung von Kontextmüll

Jedes irrelevante Token im Kontext erhöht das Risiko für **Aufmerksamkeitszerstreuung und Halluzinationen** – präzises Chunk-Re-Ranking vor der Anfrage ist Pflicht.

KAPITEL 02 / 07 · EMPIRISCHE EVIDENZ

Empirische Evidenz & Unternehmensrealität

Was wissenschaftliche Feldexperimente über Produktivität und die Jagged Frontier enthüllen

„Wie verändert generative KI messbar Geschwindigkeit und Qualität von Fachkräften?“

Ergebnisse aus kontrollierten Feldstudien mit 758 Unternehmensberatern und **warum falsches Vertrauen zu dramatischen Fehlern führt**.

Die Harvard/BCG-Feldstudie

758 Consultants im kontrollierten Experiment: **Produktivitätssprung innerhalb der Modellgrenze**

KEY METRIC

25%

Schnellere Bearbeitung

Zeitreduktion bei Consulting-Aufgaben [Del23]

KEY METRIC

40%

Höhere Qualität

Signifikanter Anstieg der Ausarbeitungsnote

KEY METRIC

12%

Mehr Aufgaben gelöst

Erhöhte Gesamtproduktivität pro Zeiteinheit

Das zweischneidige Schwert [Del23]

„Liegt eine Aufgabe außerhalb der technologischen Grenze, **sinkt die menschliche Genauigkeit mit KI um 19 Prozentpunkte** unter das Niveau ohne KI.“

Die Jagged Technological Frontier

Warum KI an scheinbar komplexen Aufgaben brilliert, **aber an trivialen Details scheitert**

Grenzkartierung [Del23]

Centaur & Cyborg

Prompts fungieren als Tastwerkzeug zur **präzisen Kartierung von Modellgrenzen.**

Innerhalb der Frontier

Heterogene Textsynthese und **kreative Strategiekonzepte gelingen spielend leicht.**

Außerhalb der Frontier

Kausallogik und **versteckte quantitative Ausnahmebedingungen** scheitern überraschend.

Falling Asleep at the Wheel

Zu hohe Plausibilität führt zu **intellektuellem Kontrollverlust der Anwender.**

Der Nivellierungseffekt

Stanford/MIT-Studie: **Der größte Hebel liegt bei weniger erfahrenen Mitarbeitern**

Produktivitätsgewinn nach Skill-Level



Die NBER-Feldstudie [BLR23] belegt den empirischen Equalizer-Effekt: Generative KI hebt vor allem das Leistungsniveau leistungsschwächerer Mitarbeiter auf den Median an, während Top-Experten ihre Produktivität primär durch Spezialisierung und kritische Ergebnisvalidierung halten.

Der 'Equalizer'-Effekt

Wissenstransfer

Prompts übertragen Best Practices sofort auf Neueinsteiger.

Kompression der Lernkurve

Das Onboarding gelingt schneller, da Modelle **die Einarbeitungszeit neuer Fachkräfte drastisch verkürzen.**

Standardisierung des Outputs

Qualitätsunterschiede sinken durch **unternehmensweit einheitliche Prompt-Vorlagen** im Alltag.

B2B-Praxis: Dokumentenanalyse

Summaries & Kennzahlen: **Vom Fließtext zur verlässlichen Entscheidungsgrundlage**

Strukturierter Extraktions-Workflow

Bottom Line Up Front (BLUF)

Die wichtigste Erkenntnis steht ganz oben: **Sofortige Management-Entscheidungsfähigkeit.**

Tabellarische Extraktion

Finanzkennzahlen, Deadlines und Risiken in **eine strikte tabellarische Matrix überführen.**

Die Fußnoten-Falle

Kritische Vorbehalte im Kleingedruckten bergen **hohe Risiken für fatale Fehlinterpretationen.**

Exakter Quellennachweis

Das Modell muss jede Aussage durch **wörtliche Zitate und Seitenzahlen belegen.**

B2B-Praxis: Ticket-Triage & Support

Automatisierte Klassifikation: **Präzises Routing und strukturierte Vorqualifizierung**

Triage-Mechanik

Deterministische Taxonomie

Jedes Ticket wird in ein **eindeutiges Kategorien-Set klassifiziert** und **strukturiert vorqualifiziert.**

Dringlichkeits-Bewertung

Berechnung eines **objektiven Dringlichkeits-Scores von 1 bis 5** anhand definierter SLA-Signalwörter und Ausfallkriterien.

Automatisches Routing

Direkte Zuteilung zum **passenden First- oder Second-Level-Team** inklusive strukturierter Zusammenfassung ohne Medienbruch.

Sicherheits-Leitplanken

Verbindlichkeits-Schranke

Striktes **Verbot eigenmächtiger Kulanz- oder Preiszusagen** im automatisierten Dialog. Das System formuliert ausschließlich unverbindliche Vorprüfungen.

Human-in-the-Loop Eskalation

Bei eskalierendem Tonfall oder Frustration erfolgt die **sofortige Übergabe an menschliche Servicemitarbeiter**. Ein deterministischer Sentiment-Filter benachrichtigt direkt den Support-Lead.

Robuste Guardrails sichern die finale Entscheidungshoheit beim Fachpersonal.

Die Kosten ineffizienter Prompts

Warum schlechtes Prompting **enorme Latenzen, Token-Kosten und Nacharbeit verursacht**

KEY METRIC

5x

Kosten-Multiplikator

Durch Re-Prompts **verfünffachen sich API-Kosten und Antwortzeiten** bei jedem Arbeitsschritt.

Die drei verborgenen Kostentreiber

Re-Generation Loops

Endloses Neuformulieren bindet Arbeitszeit: **Mitarbeiter verlieren wertvolle Produktivstunden.**

Kontext-Aufblähung

Ungefilterte Dokumentenanhänge führen dazu, dass **tausende ungenutzte Tokens pro API-Call abgerechnet werden.**

Fehlende Formatgarantie

Freitext-Ausgaben erzwingen einen **enormen manuellen Kontroll- und Korrekturaufwand** vor der Weiterverarbeitung.

KAPITEL 03 / 07 · MYTHEN & HYPES

Dekonstruktion von Hype & Akronym-Modellen

Warum populäre Internet-Faustregeln Scheinsicherheit bieten und im Enterprise-Stack versagen

„Helfen Akronym-Modelle wie CLEAR oder CREATE wirklich in der industriellen Praxis?“

Kritische Einordnung von Rollenanweisungen und **warum menschliche Intuition beim Steuern stochastischer Systeme versagt.**

Akronym-Modelle im Härtetest

Von CLEAR über CREATE bis RTCE: **Nützliche Krücken mit harten industriellen Grenzen**

FRAMEWORK	TYPISCHER AUFBAU	DIDAKTISCHER NUTZEN	INDUSTRIELLE LIMITATION
C-L-E-A-R Concise, Logical, Explicit	**Kürze & Verständlichkeit**	Sensibilisiert für Unwichtiges	Keine formalen Schemas; **bleibt ohne formale Spezifikation vage.**
C-R-E-A-T-E Character, Request, Examples	**In-Context-Lernen**	Fördert Nutzung von Beispielen	Fokus auf Rollen; **vermischt aber Persona und Datenformate unsauber.**
R-T-C-E Role, Task, Context, Expectation	**Strukturierte Eingabe**	Guter logischer Rahmen	Fehlende Typisierung; **bietet jedoch weder Schema-Validierung noch Fallback-Pfade.**
Schema-First XML, JSON-Schema, Pydantic	**Software-Pipelines**	100% deterministische Syntax	Der Enterprise-Standard: **Typisierte Schnittstellen mit garantierter Struktur.**

Das Rollen-Paradoxon

Warum 'Handle als Experte' **den Tonfall verändert, aber keine Faktenkompetenz erzeugt**

Die Persona-Illusion

„Du bist ein weltführender Senior-Ingenieur mit 30 Jahren Erfahrung im Anlagenbau…“

Kein magischer Wissenszugriff

Die Persona **schaltet kein neues Faktenwissen im Vektorraum frei.**

Rein stilistische Verschiebung

Verändert wird lediglich das Vokabular: **Höhere Dichte an Fachausdrücken und steifere Tonalität.**

Gefahr der Sykophantie

Rollen-Prompts neigen zu Bestätigungsfehlern: **Modelle schmeicheln Annahmen des Nutzers.**

Richtiger Einsatzzweck

Setzen Sie Personas **gezielt zur Adressierung von Empfängern ein** (Vorstand vs. Monteur).

Höflichkeit, Drohungen & Voodoo

Warum Trinkgeld-Versprechen und emotionale Appelle **statistisch wirkungsloser Ballast** sind

Die Meme-Kultur im Check

„Ich gebe dir 100 $ Trinkgeld für eine perfekte Antwort, mein Job hängt davon ab!“

Statistisches Nullsummenspiel

Empirische Benchmarks zeigen: Trinkgelder und **fiktive Notlagen oder Strafandrohungen** bringen keinen signifikanten Genauigkeitsgewinn.

Kontext-Rauschen

Emotionale Floskeln erzeugen **unerwünschte Verteilungs-Verzerrungen** im Aufmerksamkeitsfokus.

Was Modelle wirklich steuert

Ingenieur-Prinzip

Stochastische Prozessoren reagieren auf Syntax und Semantik, **nicht auf Emotion oder soziale Normen.**

Strukturelle Spezifikation

Eindeutige Datenfelder und **strukturelle Randbedingungen, Typisierungen und Few-Shot-Beispiele** sichern reproduzierbar die Ausgabequalität und deterministische Validierbarkeit in produktiven Enterprise-Pipelines.

Die Kontroll-Illusion

Warum menschliche Intuition bei stochastischen Systemen **systematisch in die Irre führt**

Nicht-Lineare Dynamik

Schmetterlingseffekt

Ein einziges verändertes Synonym im Eingabe-Prompt kann **unvorhersehbare qualitative Sprünge auslösen.**

Die Stichproben-Falle

Lokale Scheinerfolge

Was bei drei manuellen Tests im Labor perfekt funktioniert, **bricht bei hundert realen Kundenfällen ein.**

Implizite Annahmen

Fehlender Commonsense

Modelle besitzen **weder implizites Alltagswissen noch Kontext** ohne explizite Spezifikation.

Die Ingenieurs-Lösung

Metrik-basierte Entwicklung

Subjektives Bauchgefühl weicht konsequent durch **testgetriebene Validierung und automatisierte Evals** ab.

KAPITEL 04 / 07 · REASONING-TAXONOMIE

Systematische Prompting-Taxonomie

Von In-Context Demonstration zu rekursiven Denk- und Werkzeugpfaden

„Welche formalen Prompting-Techniken heben nachweisbar die logische Reasoning-Tiefe?“

Wissenschaftliche Methoden von Few-Shot Learning über Chain-of-Thought bis ReAct **anhand empirischer Primärliteratur**.

Dr. Georg Hackenberg • FH OÖ Campus Wels

In-Context Learning: Zero- vs. Few-Shot

Der Urknall des modernen Promptings: **Lernen durch exemplarische Demonstration im Prompt**

Zero-Shot Prompting

Direkte Instruktion

Aufgabenstellung ohne Beispiele: **Das Modell verlässt sich rein auf Vortrainings-Assoziationen.**

Hohe Formatvarianz

Der Output variiert von Call zu Call und leidet unter **spezifischen unternehmensinternen Formatvorgaben.**

Einsatzbereich

Ideal für breites Brainstorming und **offene Textgenerierung ohne starre Schnittstellen.**

Few-Shot In-Context [Bro20]

Muster-Konditionierung

2–5 Input/Output-Paare fixieren **syntaktische und logische Muster sofort.**

Dramatischer Präzisionsgewinn

Demonstrationen lenken die Wahrscheinlichkeit auf **signifikant höhere Vorhersagegenauigkeit.**

Kein Modelltraining nötig

Funktioniert komplett in-memory im Kontextfenster **ohne aufwendiges Fine-Tuning oder Gewichts-Updates** des Sprachmodells.

Design robuster Few-Shot Demonstrationen

Vier Gesetze für verlässliche Beispiele: **Vermeidung von Biases und Reihenfolge-Effekten**

1. Klassen-Balance

Gleichverteilung der Klassen

Ungleiche Demonstrationen führen dazu, dass **das Modell Vorhersagen zugunsten der Mehrheit verzerrt**.

2. Reihenfolge-Effekte

Recency-Bias

Statistisch neigen LLMs dazu, **überproportional häufig das Label des allerletzten Beispiels** zu wählen.

3. Randfall-Abdeckung

Edge Cases einbetten

Triviale Beispiele genügen nicht; **seltene Ausnahmen und Grenzwerte robust verarbeitet** müssen demonstriert werden.

4. Format-Symmetrie

Identische Delimiter

Trennzeichen, Einrückungen und konsistente XML-Tags: **Jedes Beispiel muss die Zielstruktur exakt spiegeln**.

Chain-of-Thought (CoT) Reasoning

NeurIPS 2022: **Wie Zwischenrechenschritte die logische Tiefe revolutionieren**

Warum CoT funktioniert [Wei22]

Tokens als Rechenkapazität

Jedes geschriebene Zwischenwort ist **physische Rechenzeit im Transformer-Forward-Pass.**

Working Memory im Kontext

Das Modell schreibt **Schritt 1, Schritt 2 und Zwischenergebnisse** direkt in sein Kontextfenster.

Aufmerksamkeits-Speicher

Zwischenwörter dienen als Anker für **nachfolgende Self-Attention-Berechnungen.**

Dekomposition komplexer Logik

Schwierige Aufgaben werden in **eine Serie kleiner, lösbarer Einzelschritte zerlegt.**

Zero-Shot CoT: 'Schritt für Schritt'

Kojima et al. 2022: **Wie vier Worte Benchmark-Sprünge von 17 auf 78 Prozent auslösten**

reasoning-trigger.txt

```
User: Eine Fabrik produziert 120 Teile/h.
Bei 2 Schichten à 8h mit 5% Ausschuss:
Wie viele fehlerfreie Teile entstehen?
Trigger: Let's think step by step.

Model (Reasoning):
1. Gesamte Arbeitszeit: 2 Schichten * 8h = 16h
2. Brutto-Produktion: 16h * 120 Teile = 1920 Teile
3. Ausschuss berechnen: 1920 * 0.05 = 96 Teile
4. Netto-Ertrag: 1920 - 96 = 1824 Teile
Fazit: Es entstehen exakt 1.824 fehlerfreie Teile.
```

Empirischer Durchbruch [Koj22]

MultiArith Benchmark

Genauigkeitssprung von 17,7 % auf 78,7 %
ohne ein einziges Beispiel!

Zwei-Stufen-Architektur

Phase 1 generiert den Rechenweg; Phase 2 nutzt **einen Zwei-Phasen-Ansatz aus Reasoning-Schritt und Parsing-Schritt.**

Aktivierung latenter Logik

Der Trigger zwingt das Modell, **logische Zwischenzustände im latenten Raum zu aktivieren.**

Self-Consistency Sampling

Wang et al. 2022: **Mehrheitsentscheid über parallele stochastische Denkpfade**

GSM8K Benchmark [Wan23]

Standard CoT (k=1) (Einzelner Pfad) **+57 %**

Majority Vote (k=5) (Fünf Denkpfade) **+68 %**

Majority Vote (k=40) (Konsens-Plateau) **+74 %**

MAJORITY VOTING ÜBER K PFADE

$$\arg\max_a \sum_{i=1}^{k} \mathbb{I}(a_i = a)$$

Paralleles Sampling marginalisiert individuelle Ausreißer zuverlässig heraus und maximiert die Antwortstabilität bei komplexen Aufgaben.

Funktionsweise im Enterprise-Stack

Stochastisches Ensemble

Der Mehrheitsentscheid filtert zufällige Rechenfehler heraus.

Sampling mit Varianz (T ≈ 0.7)

Durch stochastische Streuung generiert das Modell **völlig unterschiedliche Lösungswege zum selben Problem.**

Marginalisierung des Pfads

Nicht der Einzelpfad zählt, sondern **ein hochgradig robustes Konsens-Ergebnis** über unabhängige Herleitungen.

Dekomposition: Least-to-Most

Zhou et al. 2022: **Rekursive Problemzerlegung für hochkomplexe Multi-Step-Aufgaben**

LOADING VISUAL...

1

Dekomposition

Zerlegung des Ziels in **eine geordnete Liste elementarer Sub-Probleme** [Zho22].

2

Basis-Lösung

Das Modell löst **das einfachste Teilproblem isoliert** und präzise.

3

Rekursive Übergabe

Die gefundene Teillösung wird **als fester Input in die nächste Subfrage übergeben**.

4

Gesamtsynthese

Finale **Synthese aller Zwischenergebnisse zur Gesamtlösung**.

ReAct: Reasoning & Acting

Yao et al. 2022: **Die Synergie aus internem Nachdenken und externen Werkzeugaufrufen**

Der ReAct-Zyklus [Yao22]

Interaktives Grounding

Das Modell agiert als logischer Controller, der **seine Argumentation auf Live-Schnittstellen erdet.**

Thought → Action → Observation

Überlegungen anstellen, **gezielte API- oder Tool-Calls absetzen** und **externe Messdaten in den Prompt zurücklesen.**

Überwindung parametrischer Grenzen

Aktuelle ERP-, CRM- oder Sensordaten werden **zur Laufzeit on-demand abgerufen.**

Dynamische Selbstkorrektur

Schlägt eine Aktion fehl, erkennt das Modell den Fehlercode und **formuliert selbstständig alternative Aufrufe.**

KAPITEL 05 / 07 · ENTERPRISE ARCHITECTURE

Enterprise Prompt Architecture

Produktionsstandards: Von Freitext-Skripten zu robuster Software-Infrastruktur

„Wie konstruieren wir Prompts, die zu 100 Prozent deterministisch in ERP- und CRM-Systeme integrierbar sind?“

Das 5-Säulen-Modell, XML-Kapselung, Context Caching **und garantierte Structured Outputs via JSON-Schema.**

Das 5-Säulen-Modell industrieller Prompts

Der Konstruktionsplan für **stochastisch kontrollierte Produktions-Prompts**

1

Rolle & Persona

Präziser Domänenfokus und **Rolle und Zielgruppen-Horizont** festlegen.

2

Task & Scope

Glasklare Verben definieren **eine glasklare, operative Aufgabenstellung.**

3

Kontext & Fakten

Dokumente und Daten als **den nicht-parametrischen Kontext als primäre Faktenquelle** bereitstellen.

4

Format-Schema

JSON, Tabellen oder Markdown für **strikte Formate und strukturierte Schemas.**

5

Constraints

Verbote, Grenzen und **negative Constraints und deterministische Abbruchkriterien.**

XML-Kapselung & Delimiter

Der Industriestandard für **semantische Trennschärfe und Parsing-Stabilität**

prompt-template.xml

```xml
<system_instructions>
  <rules>Antworte präzise auf Basis des Kontexts.</rules>
</system_instructions>
<context>
  [Technisches Prüfprotokoll Charge #4829...]
</context>
<user_query>
  Welche Grenztemperatur wurde überschritten?
</user_query>
```

Architektonischer Mehrwert

Strikte Grenzziehung

XML-Tags erzeugen eine **unmissverständliche Trennung von Logik und Daten.**

Schutz gegen Daten-Verwechslung

Das Modell verwechselt Dateninhalte nicht mit neuen Befehlen und bietet **effektiven Basisschutz gegen Prompt-Injections.**

Programmatische Parsbarkeit

Antworten können in <result>-Tags isoliert und **per Regex oder Parser fehlerfrei im Backend extrahiert werden.**

Closed-Source-Hebel: Caching & System

Wie moderne API-Features **Latenzen und Kosten um bis zu 90 Prozent senken**

Native System Messages

Architektonische Priorität

System-Instruktionen besitzen **höhere Priorität in der Modell-Attention.**

Resistenz gegen Überschreiben

Benutzer-Prompts können Systemregeln **deutlich schwerer kompromittieren oder überschreiben.**

Multi-Turn Stabilität

Die Rahmenbedingungen bleiben über **viele Chat-Runden hinweg unverändert wirksam.**

Prompt Caching Hebel

Wirtschaftlicher Durchbruch

Wiederholte Kontext-Präfixe **sparen bis zu 90 Prozent der Input-Kosten.**

Dramatische Latenzsenkung

Bereits berechnete Key-Value-Caches **beschleunigen Time-to-First-Token massiv.**

Große Kontext-Bibliotheken

Unternehmens-Handbücher verbleiben im Cache für **extrem kostengünstige Folgeabfragen.**

Structured Outputs & JSON-Schema

Constrained Decoding: **100 Prozent syntaktische Garantie für nachgelagerte Software-Systeme**

schema.py

```python
from typing import Literal, List
from pydantic import BaseModel, Field

class Anomaly(BaseModel):
    sensor_id: str
    severity: Literal["LOW", "HIGH", "CRIT"]
    confidence: float

class MachineAudit(BaseModel):
    machine_id: str = Field(description="Seriennummer")
    status: Literal["OK", "WARN", "FAIL"]
    temperature_c: float
    anomalies: List[Anomaly]
    action_required: bool
```

Constrained Decoding [Ope24]

Grammatik-Maskierung

Tokens, die das Schema verletzen, erhalten Wahrscheinlichkeit null.

Zero Parsing-Errors

Schluss mit fehlerhaften Kommas: **Syntaxfehler werden mathematisch unmöglich gemacht.**

Direkte DB-Pipeline

Antworten fließen direkt und **ohne manuelle Bereinigung in SQL-Datenbanken und ERP-Tabellen.**

In-Context Grounding vs. RAG

Die Architekturentscheidung: **Wann reines Prompting genügt und wann RAG zwingend ist**

Architektur-Entscheidung [Lew20]

Wann Caching vs. RAG?

Reines Caching bei Dossiers vs. RAG bei **Millionen Dokumenten und dynamischen Beständen.**

In-Context Grounding

Quelltext wird **direkt in den Prompt übergeben** – ideal für Prüfberichte ganz **ohne fehleranfällige semantische Chunk-Suche**.

Skalierbare Wissenskorpora

Verbindet strukturierte Vektorsuche mit dem Modell für **eine skalierbare Wissensbasis im Großunternehmen**.

Berechtigungs-Filterung (ACL)

Dokument-Chunks werden vorab **nach Mitarbeiter-Rollen und Mandanten gefiltert**.

B2B-Praxis: Compliance & Vertragsanalyse

Regulatorische Audits: **Rechtssichere Klauselprüfung und lückenlose Traceability**

Automatisierter Klausel-Check

Regelbasierte Gap-Analyse

Abgleich von Drittverträgen mit Standard-AGB und internen Klauselrichtlinien.

Risiko-Früherkennung

Sofortige Warnung bei **unbegrenzter Haftung oder verletzten Kündigungsfristen** im Vertragstext.

Compliance-Regeln

Prüfung auf gesetzliche Pflichtklauseln wie **DSGVO-Auftragsverarbeitung und Geheimhaltungsfristen** vor Freigabe.

Auditierbare Evidenz-Struktur

Lückenlose Zitation

Jedes gefundene Risiko wird mit **exakter Paragraphen-Nummer und wörtlichem Zitat** belegt, sodass Juristen jede Beanstandung direkt verifizieren können.

Konformitäts-Ampel

Klassifikation in einen **eindeutigen Rot-Gelb-Grün-Konformitätsstatus** für juristische Freigaben. Rote Klauseln erzwingen eine verbindliche Fachprüfung.

Die strukturierte Ausgabe garantiert revisionssichere Nachvollziehbarkeit im Audit.

KAPITEL 06 / 07 · SICHERHEIT & GUARDRAILS

Zuverlässigkeit, Sicherheit & Guardrails

Schutzschilde für Unternehmensdaten: Epistemische Grenzen, Fallbacks und Injection-Defense

„Wie verhindern wir Halluzinationen und das Kapern von Daten-Pipelines durch Prompt Injection?“

Epistemische Halluzinationsgrenzen, deterministische Fallback-Routinen **und defensive Sandwich-Architekturen**.

Das Spektrum der Halluzinationen

Epistemische Grenzen: **Warum Modelle bei Lügen genauso überzeugend klingen wie bei Fakten**

1. Fakten-Fabrikation

Plausible Fiktionen

Synthese sprachlich hochgradig plausibler, aber **völlig erfundener Normen, URLs oder Bauteilnummern.**

2. Faithfulness-Verletzung

Kontext-Widerspruch

Das generative Sprachmodell ignoriert oder **widerspricht explizit mitgeliefertem Dokumentenkontext.**

3. Sykophantie & Bestätigung

Konfliktscheue

Aus Höflichkeits-Alignment **bestätigt die KI falsche Vorannahmen des Nutzers** ungefiltert.

4. Konfidenz-Illusion

Fehlende Kalibrierung

Generative Sprachmodell-Architekturen besitzen **kein internes Bewusstsein über eigenes Nichtwissen.**

Negative Constraints & Fallback-Design

Die Lizenz zum Nichtwissen: **Wie explizite Abbruchklauseln Halluzinationen eliminieren**

Die Deterministische Notbremse

Expliziter Fallback-Befehl

„Falls die Information nicht im Kontext steht: Antworte exakt mit ‘NICHT_GEFUNDEN’.“

Befreiung vom Antwortzwang

LLMs versuchen 'hilfreich' zu fabulieren; der explizite Fallback **nimmt den Druck zur Zwangskonfabulation** und sichert verlässliche Nicht-Aussagen.

Ein standardisierter Fehler-String erlaubt nachgelagerten Systemen automatisierte Fallback-Routinen – reproduzierbares ‘Nicht gefunden’ schützt vor erfundenen Zahlen.

Strikte Negative Randbedingungen

Extrapolations-Verbot

Striktes **Verbot jeglicher freier Extrapolation** bei unvollständigen Kennzahlen. Fehlende Werte dürfen im Bericht unter keinen Umständen geschätzt oder interpoliert werden.

Quellen-Isolation

Das Modell errichtet eine **strikte Beschränkung auf mitgelieferte Datenquellen** und ignoriert konsequent diffuses Halbwissen aus dem allgemeinen Trainingskorpus.

Negative Constraints definieren die Leitplanken des Lösungsraums und eliminieren die häufigsten Fehlerquellen geschäftskritischer Prompts.

Direct & Indirect Prompt Injection

Greshake et al. 2023: **Wenn externe Dokumente zum unbemerkten Steuercode werden**

Angriffsvektoren & B2B-Risiko [Gre23]

Befehl vs. Daten Trennung fehlt

LLMs können **instruktiven Code nicht nativ von passiven Daten trennen.**

Indirekte Injection

PDFs und Mails schleusen Steuerbefehle ein: **externe Datenquellen manipulieren das Modell unbemerkt.**

Daten-Exfiltration

Angreifer stehlen vertrauliche Unternehmensdaten durch **die Exfiltration an externe Webhooks via Bildlinks.**

Zero-Trust im Kontext

Externe Dokumente erfordern eine **Zero-Trust-Behandlung aller externen Dokumente** mit strikter Isolation.

Defensive Prompt-Architektur

Vierstufige Härtung: **Die Sandwich-Defense und strikte Zero-Trust-Kapselung**

LOADING VISUAL...

1 **Input Sanitization**
Vorab-Filterung eliminiert **verdächtige Steuerzeichen und unsichtbare Unicodes.**

2 **XML Isolation**
Kapselung des Inputs in **strikt isolierende XML-Container als ungesicherte Daten.**

3 **Sandwich Defense**
Verankerung **aller Sicherheitsregeln unmittelbar vor der Generierung.**

4 **Output Guardrail**
Ein sekundärer Check **überprüft den finalen Output auf Datenlecks und Richtlinien.**

KAPITEL 07 / 07 · SKALIERUNG & AUSBLICK

Von Prompts zu Software-Pipelines

Automatisierung, Evals & DSPy: Der Wandel zur programmatischen Optimierung

„Wie überführen wir isolierte Prompts in automatisierte, getestete CI/CD-Pipelines?“

Test-Driven Evals, automatische Prompt-Compiler wie DSPy
und die fünf goldenen Gesetze für Unternehmen.

Dr. Georg Hackenberg • FH OÖ Campus Wels

Test-Driven Prompt Evals

Schluss mit Bauchgefühl: **Kuratierte Test-Datensätze und Regressionstests im CI/CD-Zyklus**

Systematisches Test-Harnisch

Golden Dataset & Regressionen

Prompts werden wie Code **anhand von fünfzig bis zweihundert Referenzfällen** gemessen, um **ungewollte Regressionen sofort zu stoppen.**

1. Deterministische Assertions

Automatisierte Code-Prüfung auf **JSON-Validität, Feld-Vollständigkeit und Regex-Muster**.

2. LLM-as-a-Judge

Ein starkes Schiedsrichter-Modell bewertet **Semantik, Relevanz und Markentonfall nach festen Rubriken.**

Programmatische Optimierung: DSPy

Khattab et al. 2023: **Warum wir Prompts in Zukunft kompilieren statt manuell zu tippen**

Das Fragilitäts-Dilemma

„Wir haben wochenlang gefeilt – **mit dem nächsten Modell-Update funktionierte nichts mehr.**“

Model-Drift

Jedes neue Modell-Checkpoint erfordert Anpassungen: **Mühsam ertestete Formulierungen brechen oft zusammen** und erzwingen manuelles Re-Engineering.

Hohe Wartungskosten

Ingenieure verbringen wertvolle Entwicklungszeit mit **endlosem sprachlichem Feinschliff statt Systemarchitektur** und robuster Pipeline-Logik.

Das DSPy-Kompilier-Paradigma [Kha23]

Deklarative Programmierung

Prompts werden **automatisch gegen eine Verlustfunktion kompiliert und optimiert.**

Typisierte Signaturen

Definition rein über I/O (z. B. `context, query -> answer`), **völlig losgelöst von konkreten Prompt-Strings.**

Automatisches Few-Shot Tuning

Der Compiler wählt selbstständig die besten Demonstrationen für **maximale Zielmetriken auf Ihren Daten.**

Agentische Entwurfsmuster

Vier Kernmuster für komplexe Workflows: **Vom Monolithen zu modularen Multi-Prompt-Systemen**

1. Routing (Verzweigung)

Spezialisierte Handler

Ein schlanker Router leitet Anfragen **über eine Klassifikations-Weiche an spezialisierte Prompts** weiter.

2. Prompt Chaining

Sequenzielle Pipelines

Aufgaben werden in Stufen zerlegt: **Jeder Schritt wird vor der Weitergabe deterministisch validiert.**

3. Parallelisierung

Divide & Conquer

Große Datenmengen werden **in simultan verarbeitete Teilabschnitte** aufgeteilt und synthetisiert.

4. Evaluator-Optimizer

Generator-Critic Schleife

Ein Modell generiert, während **ein Critic-Modell den Entwurf iteriert, bis Qualitätskriterien erfüllt sind.**

Die 5 goldenen Gesetze für Unternehmen

Der verbindliche Leitfaden: **Von Ad-hoc-Versuchen zu reproduzierbarer Exzellenz**

1

Prompts as Code

Versionierung in Git, Review-Prozesse und **jeden Prompt wie produktiven Quellcode** behandeln.

2

Schema First

Kein Freitext für Backends: Immer **strikte strukturierte Schemas statt Freitext** nutzen.

3

Kontext-Grounding

Kein blindes Vertrauen: **Fakten immer explizit im Kontext mit** Dokumenten absichern.

4

Fallback-Pflicht

Determinismus sichern durch **explizite 'Ich weiß es nicht'-Abbruchklauseln**.

5

Dekomposition

Keine Monster-Prompts: **komplexe Vorhaben in modulare Ketten statt Monster-Prompts** zerlegen.

Fazit & Synthese

Vom Experiment zur unternehmensweiten Skalierung: **Die Zukunft gehört getesteten, typisierten Pipelines**

Zentrale Erkenntnis

Ingenieurs-Reife

Vom Experiment zum System: **die neue Ingenieursdisziplin an stochastischen Schnittstellen.**

Vorsprung durch Robustheit

Nicht wer kreativ tippt gewinnt, sondern wer **einen uneinholbaren Vorsprung an Verlässlichkeit aufbaut** und systematisch testet.

Kontinuierliche Evolution: Die Reise führt von manuellen Prompts zu versionierten, automatisiert geprüften Agenten-Pipelines im Betrieb.

Ressourcen & Kontakt

Typisierte Referenzsysteme

Verifizierte Prompts und Schemas sichern reproduzierbare Pipelines im Produktivbetrieb.

Website & Slide-Engine

Forschungsberichte, interaktive Folien und Code-Beispiele sind **auf hackenberg.tech für weiterführende Materialien** frei abrufbar.

Vernetzung & Forschung

Dr. Georg Hackenberg · FH Oberösterreich · **Professor für Informatik & Industriesysteme Campus Wels** – Austausch und Kooperation.