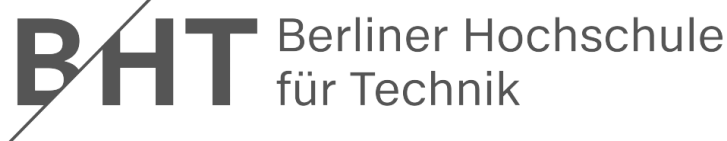


# Evaluierung und Optimierung von Large Language Models für die Entwicklung von Webanwendungen

Ein Ansatz zur Verbesserung des Entwicklungsprozesses bei  
Softwareprojekten



Masterthesis  
für den Master of Science Studiengang Medieninformatik

eingereicht von: Wilfried Pahl  
Matrikelnummer: 901932  
Studiengang: Online Medieninformatik  
Berliner Hochschule für Technik

betreut durch Prof. Dr. S. Edlich  
Berliner Hochschule für Technik

Temmen-Ringenwalde, der 18. Oktober 2024

## Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel „Evaluierung und Optimierung von Large Language Models für die Entwicklung von Webanwendungen (*Ein Ansatz zur Verbesserung des Entwicklungsprozesses bei Softwareprojekten*)“ selbstständig und ohne unerlaubte Hilfe verfasst habe. Alle benutzten Quellen und Hilfsmittel sind vollständig angegeben und wurden entsprechend den wissenschaftlichen Standards zitiert.

Ich versichere, dass alle Passagen, die nicht von mir stammen, als Zitate gekennzeichnet wurden und dass alle Informationen, die ich aus fremden Quellen übernommen habe, eindeutig als solche kenntlich gemacht wurden. Insbesondere wurden alle Texte und Textpassagen anderer Autoren sowie die Ergebnisse von Sprachmodellen wie OpenAI's GPT-3 entsprechend den wissenschaftlichen Standards zitiert und referenziert.

Ich versichere weiterhin, dass ich keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe und dass ich keine Teile dieser Arbeit in anderer Form für Prüfungszwecke vorgelegt habe.

Mir ist bewusst, dass eine falsche eidesstattliche Erklärung strafrechtliche Konsequenzen haben kann.

Temmen-Ringenwalde, den 18. Oktober 2024

Unterschrift

# ABSTRACT

Abstract in Englisch.

Entwurf

## ZUSAMMENFASSUNG

Zusammenfassung in Deutsch.

Entwurf

# INHALTSVERZEICHNIS

<b>Abstract</b>	<b>i</b>
<b>Abbildungsverzeichnis</b>	<b>v</b>
<b>Tabellenverzeichnis</b>	<b>vi</b>
<b>Listings</b>	<b>viii</b>
<b>Abkürzungsverzeichnis</b>	<b>x</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Hintergrund und Kontext	1
1.2 Problemstellung	2
1.3 Zielsetzung und Forschungsfragen	2
1.4 Aufbau der Arbeit	2
1.5 Abgrenzung	3
<b>2 Grundlagen</b>	<b>5</b>
2.1 Künstliche Intelligenz	5
2.1.1 Historisches	6
2.1.2 Maschinelles Lernen	7
2.1.3 Lernparadigmen des ML	8
2.1.4 Theoretische Grundlagen	9
2.1.5 Neuronale Netze	10
2.1.6 Deep Learning	10
2.1.7 Natural Language Processing	10
2.2 Large Language Model	10
2.2.1 Grundlagen	11
2.2.2 Historie der LLM	11
2.3 Orchestrierung von LLMs	11

2.4	Multi-Agenten-Systeme . . . . .	11
2.5	Prompt Engineering . . . . .	11
2.5.1	Prompt-Techniken . . . . .	12
2.5.2	Grenzen beim Prompt-Engineering für LLMs . . . . .	28
2.6	Grundlagen bei der Entwicklung von Webanwendungen . . . . .	28
<b>3</b>	<b>Stand der Forschung</b>	<b>29</b>
3.1	Methoden und Ansätze . . . . .	29
3.2	Forschungslücken und zukünftige Forschung . . . . .	29
3.2.1	Identifikation von Forschungslücken . . . . .	29
3.2.2	Zukünftige Forschungsrichtungen . . . . .	29
<b>4</b>	<b>Methodik</b>	<b>31</b>
4.1	Auswahl der LLM . . . . .	31
4.2	Prompt-Engineering . . . . .	31
<b>5</b>	<b>Implementierung</b>	<b>33</b>
5.1	Modelle lokal aufsetzen . . . . .	33
5.1.1	Install Ollama . . . . .	33
5.1.2	Open WebUI . . . . .	34
5.1.3	Python Client . . . . .	34
<b>6</b>	<b>Evaluation</b>	<b>35</b>
6.1	Einfache HTML Seite . . . . .	35
6.1.1	ChatGPT 3.5 . . . . .	35
<b>7</b>	<b>Anwendungsszenarien</b>	<b>37</b>
<b>8</b>	<b>Diskussion und Ausblick</b>	<b>39</b>
<b>9</b>	<b>Fazit</b>	<b>41</b>
	<b>Literatur</b>	<b>43</b>
	<b>Glossar</b>	<b>44</b>
	<b>Anhang</b>	<b>45</b>

# ABBILDUNGSVERZEICHNIS

2.1	Einordnung LLMs	5
2.2	KI Winterzyklen	6
2.3	Lernparadigmen des maschinellen Lernens	8
2.4	Baumstruktur der „Tree of Thoughts“ Technik	26

Entwurf



# TABELLENVERZEICHNIS

Entwurf

Entwurf

# LISTINGS

2.1	Zero-Shot Prompt als Python-String . . . . .	12
2.2	Antwort des Zero-Shot-Prompts . . . . .	12
2.3	Few-Shot Prompt als Python-String . . . . .	13
2.4	Antwort des Few-Shot-Prompts . . . . .	14
2.5	CoT Prompt als Python-String . . . . .	15
2.6	Antwort des CoT-Prompts . . . . .	15
2.7	Meta Prompt als Python-String . . . . .	17
2.8	Antwort des Meta-Prompts . . . . .	18
2.9	Chain Prompt Nr. 1 als Python-String . . . . .	20
2.10	Antwort des Chain-1-Prompts . . . . .	20
2.11	Chain Prompt Nr. 2 als Python-String . . . . .	21
2.12	Antwort des Chain-2-Prompts . . . . .	22
2.13	Chain Prompt Nr. 3 als Python-String . . . . .	23
2.14	Antwort des Chain-3-Prompts . . . . .	23
2.15	Ausgabe für DOMPDF Bibliothek . . . . .	25
2.16	Ausgabe für MPDF Bibliothek . . . . .	26
2.17	Ausgabe für TCPDF Bibliothek . . . . .	27
5.1	Ollama Hostanpassng für Netzwerkbetrieb . . . . .	33

Entwurf

## 1.1 Hintergrund und Kontext

Durch die zunehmende Globalisierung und Digitalisierung wird die Gesellschaft der Gegenwart und Zukunft geprägt. Der Ausbau von Hochgeschwindigkeitsnetze und die globale Corona-Pandemie haben diese Entwicklung noch einmal beschleunigt. Immer mehr Unternehmen erkennen die Potenziale der Digitalisierung und stellen ihre Prozesse um. Ganze Wertschöpfungsketten werden auf cloudbasierte Umgebungen umgestellt. Angefangen bei der Kommunikation, über Beschaffung und Produktion bis zum Verkauf der Waren und Dienstleistungen. In allen Stufen der Prozesse kommen webbasierte Anwendungen zum Einsatz, um die Kommunikation der Anwender mit den Systemen zu ermöglichen oder Schnittstellen für die Datenübertragung zwischen den Systemen zu gewährleisten. Durch wachsende Anzahl von Web-Anwendungen wächst auch der Druck an die Entwicklungsfirmen, ihre Anwendungen den schnell wechselnden Kundenanforderungen anzupassen (Beweis fehlt).

Durch diesen Prozess getrieben, müssen Entwicklungsfirmen in immer kürzeren Release-Zyklen Softwarekomponenten hinzufügen und vorhandene erweitern. Gleichzeitig wachsen aber auch die Anforderungen an Stabilität und Sicherheit der cloudbasierten Anwendungen, sowie der Bedarf an kostengünstigeren IT-Abläufen (Beweis fehlt). Ein weiteres Problem ist der wachsende Fachkräftemangel in der Wirtschaft und die damit verbundenen steigenden Gehälter der Entwickler (Beweis fehlt).

Die Verwendung künstlicher Intelligenz bei der Programmierung gewinnt immer mehr an Bedeutung. Eine Technologie die im besonderen Maße an dieser Entwicklung beteiligt ist, sind die Large Language Models. Insbesondere mit der Veröffentlichung vom ChatGPT wurde hier ein regelrechter Hype um die LLMs ausgelöst. Diese Modelle erlauben eine Softwareentwicklung mit natürlicher

Sprache. Tiefe Kenntnisse der verwendeten Programmiersprache sind nicht mehr in dem Maße erforderlich, wie ohne LLMs.

### 1.2 Problemstellung

So groß der Hype um Künstliche Intelligenz auch sein mag, zurzeit kann KI noch nicht alles. Dies sollte auch bei der Verwendung von KI generiertem Inhalten und Code beachtet werden.

*KI denkt nicht, KI trifft keine Entscheidungen. Eine KI antwortet auf eine Eingabe nicht mit der besten Antwort, sondern mit der Wahrscheinlichsten.*

Test Der Mensch muss die Ergebnisse prüfen, ehe generierte Programmcodestücke in vorhandene Programme eingefügt und in Produktionsumgebungen implementiert werden.

Viele Entwickler setzen auf ChatGPT zur Generierung von Code, wie eine Umfrage von stackoverflow vom Mai 2024 zeigt [1]. Gleichzeitig wachsen auch die technischen Schulden bei Softwareprojekten, da diese Modelle nicht für die Entwicklung von Software optimiert sind (Beweis fehlt).

### 1.3 Zielsetzung und Forschungsfragen

Diese Arbeit soll eine Auswahl von Modellen evaluieren und dessen Brauchbarkeit für die Softwareentwicklung aufzeigen. Die Modellauswahl wird von der Seite EvalPlus Leaderboard abgeleitet. Hier werden Modelle gewählt, welche erstere Plätze belegen, aber zum Vergleich auch Modelle aus dem Mittelfeld.

Des Weiteren soll gezeigt werden, ob die automatisierte Verwendung beider Techniken eine Effizienz und Effektivität des Entwicklungsprozesses gesteigert werden kann.

### 1.4 Aufbau der Arbeit

Ein paar Worte zum Aufbau dieser Arbeit. Im Kapitel 3 wird der aktuelle Stand der Forschung vorgestellt und Erkenntnisse anderer Arbeiten diskutiert. Die in dieser Arbeit verwendeten Methoden, werden im Kapitel 4 behandelt. Die Implementierung der Test LLMs wird in Kapitel 5 besprochen und in Kapitel 6 die Ergebnisse evaluiert. Bevor in Kapitel 9 auf mögliche Folgearbeiten eingegangen wird, gibt es in Kapitel 7 Anwendungsszenarien, die zu den Ergebnissen dieser Arbeit geführt haben.

## 1.5 Abgrenzung

Ausschluss anderer Anwendungsbereiche.

Rechtliche und ethische Überlegungen werden nur am Rande berücksichtigt.

Entwurf

Entwurf



Die hier besprochenen Grundlagen gehen nicht in eine Tiefe, um alle evtl. Fragen zu klären. Jedes einzelne Gebiet könnte eine Arbeit füllen. Stattdessen sollen lediglich einen kleinen Einblick geben.

## 2.1 Künstliche Intelligenz

Die Grafik 2.1 soll die Einordnung von LLMs im Bereich der künstlichen Intelligenz zeigen.

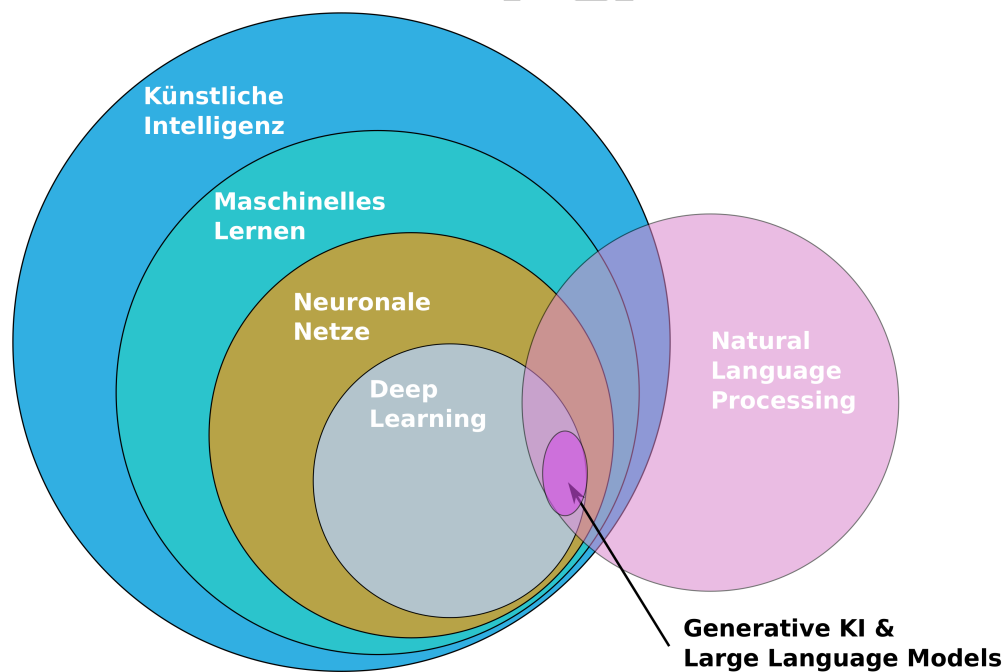


Abbildung 2.1: Einordnung LLMs

In den folgenden Kapiteln werden die wichtigsten Technologien und Begriffe erläutert.

Eine explizite Definition für künstliche Intelligenz ist zurzeit noch nicht erfolgt. Geschuldet ist diese Tatsache, da der Begriff Intelligenz nicht eindeutig definiert ist. Somit finden sich viele Versuche eine Definition für künstliche Intelligenz herzuleiten. In dieser Arbeit wird die Definition aus [2, 6 ff.] verwendet.

*Systeme der künstlichen Intelligenz (KI-Systeme) sind vom Menschen entwickelte Softwaresysteme (und gegebenenfalls auch Hardwaresysteme)<sup>3</sup>, die in Bezug auf ein komplexes Ziel auf physischer oder digitaler Ebene handeln, indem sie ihre Umgebung durch Datenerfassung wahrnehmen, die gesammelten strukturierten oder unstrukturierten Daten interpretieren, Schlussfolgerungen daraus ziehen oder die aus diesen Daten abgeleiteten Informationen verarbeiten, und über das bestmögliche Handeln zur Erreichung des vorgegebenen Ziels entscheiden. KI-Systeme können entweder symbolische Regeln verwenden oder ein numerisches Modell erlernen, und sind auch in der Lage, die Auswirkungen ihrer früheren Handlungen auf die Umgebung zu analysieren und ihr Verhalten entsprechend anzupassen.* (Wirtschaftliche Bedeutung ..., 26 ff.).

### 2.1.1 Historisches

An dieser Stelle eine kleine historische Exkursion in der Entwicklung der künstlichen Intelligenz.

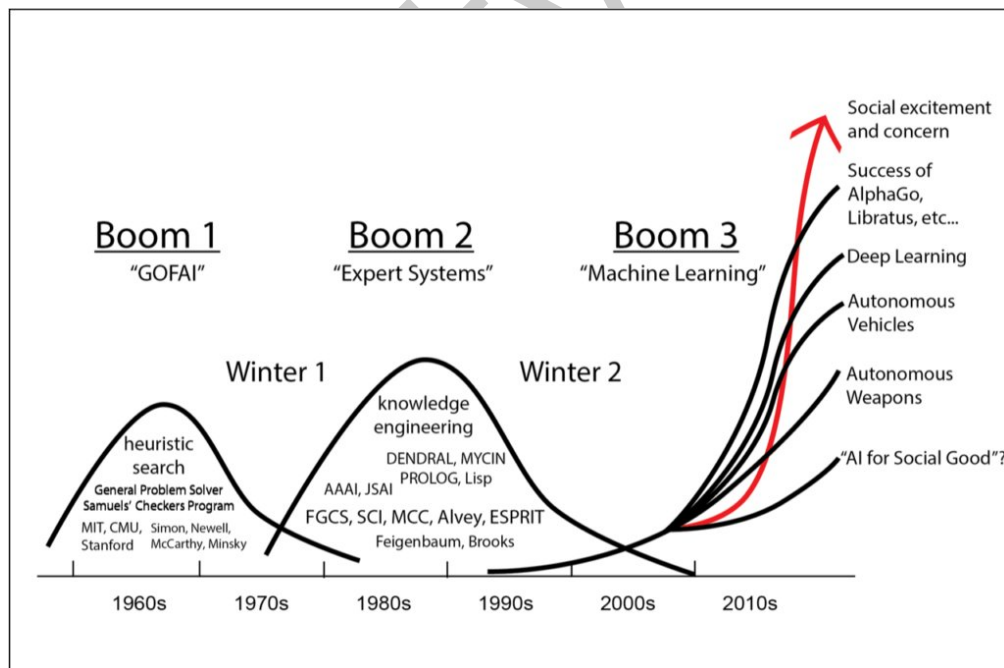


Abbildung 2.2: KI Winterzyklen

1966: Rückschläge bei der maschinellen Übersetzung

1969: Auswirkungen der Perzeptron-Kritik

1971-75: Die Finanzierungsherausforderungen der DARPA

1973: Der Fallout des Lighthill-Berichts

1973-74: Kürzung der DARPA-Mittel

1987: Zusammenbruch der LISP-Maschine

1988: Kürzungen bei der strategischen Datenverarbeitung

1990er Jahre: Niedergang von Expertensystemen

1990er Jahre: Ende des Projekts der fünften Generation

KI-Frühling des frühen 21. Jahrhunderts

### 2.1.2 Maschinelles Lernen

Das Gebiet um maschinelles Lernen (eng. Machine Learning) ist ein Teilgebiet der Künstlichen Intelligenz. Für das maschinelle Lernen wird in dieser Arbeit die allgemein gültige Definition nach Tom M. Mitchell verwendet.

*A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .*

Ein Computerprogramm lernt aus Erfahrung  $E$  in Bezug auf eine Klasse von Aufgaben  $T$  und ein Leistungsmaß  $P$ , wenn sich seine Leistung bei Aufgaben in  $T$ , gemessen an  $P$ , mit der Erfahrung  $E$  verbessert.

Mitchell, Tom M

Bei dieser Definition ist,

**E** die **Erfahrung**, die Daten aus denen das System lernt,

**T** beschreibt die **Aufgabe**, die das System erledigen soll und

**P** ist die **Leistungskennzahl** an dem der Erfolg des Systems die Aufgabe zu lösen gemessen wird.

Maschinelles Lernen (ML) und Künstliche Intelligenz (KI) sind nicht wirklich in der Lage selbstständig zu lernen oder denken, sie imitieren dies lediglich nach. Das maschinelle Lernen ist aber wohl in der Lage komplexe Muster und Funktionen in großen Datenmengen zu erkennen. Durch die Unfähigkeit zu lernen kann KI keine neuen Inhalte schaffen.

Es ist auch egal wie gut die Modelle trainiert werden, eine 100% Fehlerfreiheit gibt es nicht. Für die Praxis bedeutet dies, die Ausgaben von Modellen müssen durch Menschen immer wieder evaluiert werden, um dessen Richtigkeit sicherzustellen. Zudem sollten keine Modelle verwendet werden, wenn die Lösung eines Problems durch einen konkreten Algorithmus erfolgen kann. Hier können die Ergebnisse nachvollzogen werden und sind für den Menschen besser zu verstehen.

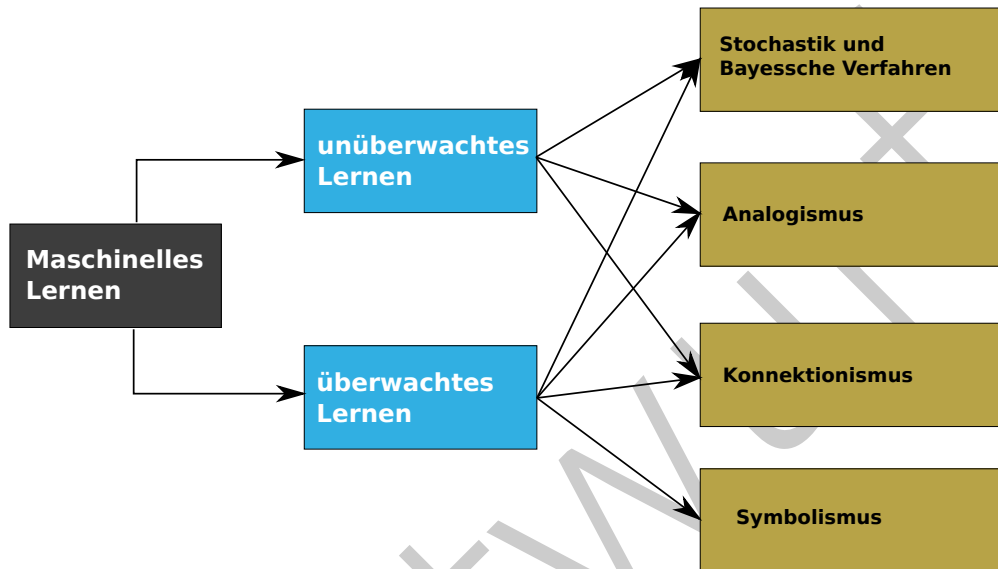


Abbildung 2.3: Lernparadigmen des maschinellen Lernens

### 2.1.3 Lernparadigmen des ML

Das maschinelle Lernen wird in zwei wichtige Formen der Lernparadigmen unterteilt. Dem überwachten und dem unüberwachten Lernen. Abbildung 2.3 zeigt die wichtigsten Lernparadigmen des maschinellen Lernens.

#### Überwachtes Lernen

Bei überwachtem Lernen sind für die Eingaben der Trainingsdaten dazugehörige Ausgaben (Labels) definiert. Das Ziel ist es eine Funktion zu trainieren um künftige Eingaben korrekt klassifizieren oder vorhersagen zu können.

### Unüberwachtes Lernen

Bei unüberwachten Lernen sind die gelabelten Ausgaben nicht vorhanden. Hierbei wird beispielsweise durch Clustering oder Dimensionsreduktion versucht Muster und Strukturen zu erkennen.

#### 2.1.4 Theoretische Grundlagen

In den folgenden Kapiteln werden vier wichtige Konzepte zum maschinellen Lernen besprochen.

#### Stochastik und Bayessche Verfahren

Als Teilgebiet der Mathematik befasst sich die Stochastik mit Wahrscheinlichkeiten und zufälligen Prozessen. Auf dem Gebiet des maschinellen Lernens werden mithilfe der Stochastik Prognosen erstellt.

Bei den Bayesschen Verfahren handelt es sich um stochastische Methoden, die auf dem Bayes-Theorem basieren. Hierbei werden neue Daten berücksichtigt, um die Wahrscheinlichkeit zu aktualisieren.

Beide Methoden finden im überwachten und unüberwachten Lernen Anwendung.

#### Analogismus

Dieser Lernansatz sucht nach Ähnlichkeiten in den Daten. Er basiert auf der Annahme, dass ähnliche Daten ähnliche Vorhersagen oder Klassifizierungen besitzen. Dieses Modell lernt, indem neue Daten mit bekannten verglichen und nach ähnlichen Strukturen und Mustern gesucht. Ein bekanntes Verfahren für diesen Lernansatz ist der k-Nearest Neighbors (k-NN).

Der Analogismus wird im überwachten Lernen als auch im unüberwachten Lernen angewandt, um Muster und Strukturen zuerkennen.

#### Konnektionismus

Der Lernansatz des Konnektionismus beruht auf kleinen Einheiten, die miteinander verbunden sind. Diese werden als Neuronen bezeichnet, die den Nervenzellen von Organismen nachempfunden sind. Die künstlichen neuronalen Netze sind die bekanntesten Vertreter, auf denen auch Deep Learning Modelle basieren.

Auch dieser Lernansatz wird im unüberwachten und überwachten Lernen angewandt.

### Symbolismus

Anders als beim Konnektionismus arbeiten die Einheiten beim Symbolismus mit explizite formale Regeln und Symbole, um das Wissen darzustellen. Der Symbolismus ist weniger flexible im Umgang mit unvollständigen Datensätzen und Unsicherheiten. Daher hat dieser Lernansatz weniger Relevanz als der Konnektionismus.

Dieser Lernansatz findet im überwachten Lernen Anwendung, als Beispiel sind hier Entscheidungsbäume zu nennen.

### 2.1.5 Neuronale Netze

KNN

#### Neuronen im neuronalen Netz

Neuronen

#### Arten der neuronalen Netzen

KNN Arten

#### Lernprozess im neuronalen Netz / Training

Training

### 2.1.6 Deep Learning

DL

### 2.1.7 Natural Language Processing

NPL

## 2.2 Large Language Model

Large Language Model

## 2.2.1 Grundlagen

Grundlagen

### Tokenisierung

Token

### Embedding

Embedding

### Vorhersage

Transformer

### Dekodierung

Dekodierung

## 2.2.2 Historie der LLM

Historie

## 2.3 Orchestrierung von LLMs

Orchestrierung

## 2.4 Multi-Agenten-Systeme

Multi-Agent-System

## 2.5 Prompt Engineering

Prompt Engineering optimiert die Antworten große Sprachmodelle, ohne Parameter, wie Bias und Gewichte des Models ändern zu müssen. Dieser Bereich hat in den letzten Jahren enorm an Bedeutung gewonnen und sich zu einer eigenen Disziplin im Bereich der Künstlichen Intelligenz entwickelt.

Ein Prompt oder Anweisung muss entweder als Anweisung oder als Frage gestellt werden. Dies kann, wie in [3] beschrieben, in Form von einer einfachen Anweisung bis hin zu detaillierten Beschreibungen oder spezifischen Aufgaben erfolgen.

[Hier Beispiel von ChatGPT oder Gemini einfügen, kann als Bild]

### 2.5.1 Prompt-Techniken

Siehe Prompting Techniques Hinweise für die Optimierung von Prompts. Die folgenden Techniken dienen dazu die Abfragen zu optimieren und somit eine bessere Antwort von den Sprachmodellen zu erhalten.

#### Zero-shot Prompting

Bei dieser Technik handelt es sich um das Senden einer einfachen, klaren und präzisen Anweisung, ohne Angabe von Beispielen und sonstigen zusätzlichen Informationen an die Modelle. Hierbei handelt es sich meistens um eine Domain-spezifische Anweisung. Es findet kein explizites Training vorher statt. Die Anweisung sollte ein klar definiertes Ziel haben

```
1 prompt_string: str = """
Generiere eine JavaScript-Funktion, die ein DIV Tag sucht und einen Text
einfügt.
"""
```

Listing 2.1: Zero-Shot Prompt als Python-String

Im Folgenden die Antwort vom Modell. Hierbei wurde das Modell *deepseek-coder-v2* verwendet und über die API abgefragt.

```
2 Natürlich! Hier ist ein Beispiel für eine JavaScript-Funktion, die ein '
DIV'-Element sucht und Text hinzufügt:
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
25
```



```

12     console.log("Das DIV-Element mit dem angegebenen Selektor wurde
        nicht gefunden.");
    }
}

// Beispielaufruf der Funktion
17 addTextToDiv('div', 'Hallo, dies ist ein eingefügter Text.');
```

'''

In diesem Code:

- 'selector' ist der CSS-Selektor für das 'DIV'-Element.
- 22 - 'text' ist der Text, den Sie dem 'DIV' hinzufügen möchten.
- Die Funktion verwendet 'document.querySelector(selector)' um das Element zu finden.
- Wenn das Element gefunden wird, fügt die Funktion 'innerText' des Elements den angegebenen Text hinzu.

Listing 2.2: Antwort des Zero-Shot-Prompts

Um bessere Lösungen von den Modellen zu bekommen, kann es sinnvoll sein, weitere Angaben, zum Beispiel zur Arbeitsweise oder die Definition, der zu verwendenden Bibliotheken hinzuzufügen. In [4] wird eine Methode vorgeschlagen, zur Verbesserung der Zero-shot Anweisungen.

### Few-shot Prompting

Bei komplexen Aufgaben liefert die Verwendung von Zero-shot Anweisungen oft unzureichende Ergebnisse. Hierfür finden Few-shot Prompts Verwendung. Bei dieser Technik werden ein oder mehrere Beispiele einer Antwort der Anweisung beigefügt, die als eine Art Antwortvorlage für das Modell dienen. Das Listings ?? zeigt beispielhaft den Prompt als Python-String, welcher an das Modell übertragen wurde.

```

1 prompt_string: str = """
Hier sind zwei Beispiele eine JavaScript Funktion die jeweils ein DIV mit
Text füllen:
Beispiel 1
'''javascript
// JavaScript Funktion, die ein DIV mit einer ID findet und Text hinzufügt
6 function insertTextInDiv() {
    let divElement = document.getElementById('myDiv');
    divElement.innerText = 'Hallo Welt!';
}
'''
"""

```

```

11 """
    Beispiel 2
    """
    // JavaScript Funktion, die das erste DIV-Element auf der Seite sucht und
    Text einfügt
    function insertTextInFirstDiv() {
16     let divElement = document.querySelector('div');
        divElement.innerText = 'Willkommen auf der Webseite!';
    }

    """
    Aufgabe: "Generiere eine JavaScript-Funktion, die ein DIV-Tag mit einer
    bestimmten Klasse sucht und einen Text einfügt."
21 """

```

Listing 2.3: Few-Shot Prompt als Python-String

Das Ergebnis vom Modell *deepseek-coder-v1:latest* ist in Listing 2.4 zu sehen.

```

Hier ist eine JavaScript Funktion, die ein DIV-Tag mit einer bestimmten
Klasse sucht und einen Text hinzufügt:

    """
    // JavaScript Funktion, die ein DIV mit einer bestimmten Klasse findet und
4     Text hinzufügt
    function insertTextInDivWithClass(className) {
        let divElement = document.querySelector('div.' + className);
        if (divElement) {
            divElement.innerText = 'Hallo von der Funktion!';
9         } else {
            console.log('Es wurde kein DIV mit der Klasse "' + className + '"
                gefunden.');
```

```

        }
    }
    """
14
Du kannst diese Funktion aufrufen und die gewünschte Klasse als Argument ü
bergeben:

    """
    javascript
    insertTextInDivWithClass('my-class'); // Ersetze 'my-class' durch deine

```

```
19   tatsächliche Klasse
    '''
```

Listing 2.4: Antwort des Few-Shot-Prompts

Wie erfolgreich diese Technik ist, wird in [5] beschrieben. Wie wichtig bei der Formulierung der Anweisungen das Format und die Beschriftung ist, zeigt [6] in seiner Studie. Wird ein Beispiel angegeben, kann dazu kommen, dass das Modell nicht die richtige Antwort findet. Dann sollten mehrere Beispiele an das Modell übergeben werden.

### Chain-of-Thought Prompting (CoT)

Wenn mit den Zero-shot und Few-shot Techniken nicht das gewünschte Ergebnis von den Modellen erzielt wird, könnte die Chain-of-Thought CoT Technik Verwendung finden. Bei dieser Technik wird das Modell aufgefordert, sein Vorgehen zu belegen. Mit dieser Technik kann besser nachvollzogen werden, wie im Modell der Lösungsversuch abläuft.

```
1  prompt_string: str = """
    Generiere eine JavaScript-Funktion, die ein DIV-Tag sucht und einen Text
    einfügt. Denke Schritt für Schritt darüber nach, wie du die Aufgabe
    angehst.
    """
```

Listing 2.5: CoT Prompt als Python-String

Die Antwort.

```
2  Um eine JavaScript-Funktion zu erstellen, die ein DIV-Tag sucht und einen
    Text einfügt, können wir folgendermaßen vorgehen:

    1. Auswahl des DOM-Elements: Wir müssen zunächst das DIV-Element
        finden, dem der Text hinzugefügt werden soll. Hierfür verwenden wir die
        Methode 'getElementById' oder andere Selektoren wie 'querySelector'.

    2. Einfügen des Textes: Sobald wir das DIV-Element gefunden haben, kö
        nnen wir den gewünschten Text in dieses Element einfügen. Dies
        erreichen wir mithilfe der Eigenschaft 'innerText' oder 'textContent'.

7  Hier ist eine detaillierte Implementierung der Funktion:

    '''javascript
```

```
function insertTextIntoDiv(divId, text) {
    // 1. Auswahl des DOM-Elements
12    var divElement = document.getElementById(divId);

    // Überprüfen, ob das Element existiert
    if (divElement) {
        // 2. Einfügen des Textes
17        divElement.innerText = text;
    } else {
        console.error("Das DIV-Element mit der ID " + divId + " wurde
            nicht gefunden.");
    }
}
22 '''

### Erklärung:
1. **getElementById**: Diese Methode gibt das Element mit der angegebenen
    ID zurück. Wenn das Element nicht existiert, wird 'null' zurückgegeben.
    Daher prüfen wir im Code, ob 'divElement' nicht 'null' ist, bevor wir
    versuchen, den Text einzufügen.
2. **innerText**: Diese Eigenschaft setzt den Text innerhalb des Elements.
    Wenn das Element bereits Inhalt enthält, wird dieser überschrieben.
27

### Verwendung:
Um die Funktion zu verwenden, rufen wir sie mit der ID des DIV-Elements
und dem gewünschten Text auf:

'''html
32 <!DOCTYPE html>
<html lang="de">
<head>
    <meta charset="UTF-8">
    <title>Beispiel</title>
37 </head>
<body>
    <div id="myDiv"></div>
    <script>
        // Funktion aufrufen
42    insertTextIntoDiv('myDiv', 'Hallo, Welt!');
```

```

    </script>
</body>
</html>
'''

```

47

In diesem Beispiel wird das DIV-Element mit der ID 'myDiv' gesucht und der Text "Hallo, Welt!" darin eingefügt.

Listing 2.6: Antwort des CoT-Prompts

Wie gut die Technik bei Modellen funktioniert, wird in [7] untersucht.

## Meta Prompting

Für Meta Prompting, sind nach [8] die wichtigsten Merkmale wie folgt,

- *Strukturierung* der Prompts beispielsweise bestimmen der Denkweise oder Reihenfolge vorgeben.
- *Syntax fokussiert*, dadurch wird die Syntax als Leitvorlage für die erwartete Antwort verwendet.
- *abstrakte Beispiele* die sich mit der Struktur des Prompt befassen, nicht mit der expliziten Lösung, wie die inhaltsgesteuert Few-Shot-Prompts.
- *vielseitig* lässt es zu, das der Prompt in vielen Bereichen anwendbar ist und geben Antwort auf eine Vielzahl von Problemen, sodass der Prompt nicht jedes Mal neu geschrieben werden muss.
- *kategorischer Ansatz* strukturiert den Aufbau der Prompts in logische Anordnung und Kategorisierung

Dem Modell wird wie bei der CoT Technik angewiesen, sein Vorgehen offen zulegen. Neben dem Ergebnis, wie beim CoT, soll hierbei auch der Ablauf und die Planung der Ergebnisfindung dargestellt werden. Ein Beispiel einer Anweisung könnte folgendermaßen aussehen und wird in Listing 2.7 gezeigt.

```
prompt_string: str = """
```

2

```

Bevor du den Code für die Erstellung einer PHP-Funktion schreibst, die
HTML in ein PDF speichert, erkläre die Wahl der Tools, die notwendigen
Installationsschritte und wie du die Funktion implementierst. Verwende
DomPDF und beschreibe den gesamten Prozess in klaren Schritten, bevor
du den Code präsentierst.
"""

```

Listing 2.7: Meta Prompt als Python-String

Das Listing 2.8 zeigt die ausführliche Antwort. Ebenfalls wurde hier das *deepseek-code-v2:latest* Modell angewandt.

```
Natürlich! Bevor ich dir den Code für eine PHP-Funktion zur Erstellung
    eines HTML-Inhalts zu PDF mit DomPDF schreibe, möchte ich zunächst die
    Wahl der Tools und deren notwendigen Installationsschritte erklären.
    Dann werde ich den gesamten Prozess in klaren Schritten beschreiben.

2  ### Wahl der Tools: DomPDF
    DomPDF ist eine leistungsstarke PHP-Bibliothek, die es ermöglicht, HTML-
    Inhalte direkt in PDF-Dateien zu konvertieren. Es basiert auf dem
    ebenfalls beliebten TCPDF und bietet einige Vorteile wie bequeme
    Integration und eine benutzerfreundliche API.

    ### Notwendige Installationsschritte:
7  1. **Serverkonfiguration**: Stellen Sie sicher, dass Ihr Server die
    notwendigen Erweiterungen (z.B. 'php-dompdf') aktiviert hat. Dies kann
    über die '.htaccess'-Datei oder eine Konfigurationsdatei in PHP
    erfolgen.
    2. **Bibliotheken herunterladen**: Laden Sie die DomPDF-Bibliothek von der
    offiziellen Website oder einem gängigen Paketmanager wie Composer
    herunter.
    3. **Composer verwenden**: Wenn du Composer verwendest, füge dies in
    deiner 'composer.json' Datei hinzu:
    12  '''json
    {
        "require": {
            "dompdf/dompdf": "~1.0"
        }
    }
    17  '''
    Dann führe 'composer install' aus.
    4. **Autoloader einbinden**: In deiner PHP-Datei, die DomPDF verwendet,
    binde den Autoloader von Composer ein:
    22  '''php
    require_once 'vendor/autoload.php';
    '''

    ### Implementierung der Funktion:
    1. **Initialisieren Sie DomPDF**: Erstellen Sie eine neue Instanz von
```

DomPDF in Ihrer PHP-Datei.

2. **\*\*HTML-Inhalt bereitstellen\*\***: Übergeben Sie den HTML-Code oder die URL des HTML-Dokuments an die DomPDF-Instanz.
3. **\*\*PDF generieren\*\***: Rufen Sie die Methode auf, um das PDF zu erzeugen.
- 27 4. **\*\*Speichern oder Anzeigen des PDFs\*\***: Speichern Sie das PDF in einer Datei oder geben Sie es direkt an den Browser aus.

Hier ist ein Beispiel für eine einfache PHP-Funktion, die HTML-Inhalt zu einem PDF konvertiert:

```
32 <?php
require_once 'vendor/autoload.php';

use Dompdf\Dompdf;

37 function generatePDF($html) {
    // Instanz von Dompdf erstellen
    $dompdf = new Dompdf();

    // HTML-Inhalt setzen
42 $dompdf->loadHtml($html);

    // (Optional) Konfigurieren Sie Dompdf, um die Größe des Blattes und
    // andere Einstellungen anzupassen
    $dompdf->setPaper('A4', 'portrait');

47 // Rendern Sie das PDF
    $dompdf->render();

    // Speichern oder Ausgeben des PDFs
    $output = $dompdf->output();
52 file_put_contents('example.pdf', $output);
}

// Beispiel-HTML-Inhalt
$html = '<h1>Hallo, Welt!</h1><p>Dies ist ein Test für die Konvertierung
    von HTML nach PDF.</p>';
57 generatePDF($html);
```

```
?>
'''

### Zusammenfassung:
62 1. **Installiere DomPDF und andere benötigte Bibliotheken**.
   2. **Initialisiere eine Instanz von DomPDF** in deiner PHP-Datei.
   3. **Setze den HTML-Inhalt oder die URL des HTML-Dokuments**.
   4. **Rendern Sie das PDF** und speichern oder geben Sie es aus.
67 Mit diesem Prozess kannst du einfach HTML-Inhalte in professionelle PDFs
   umwandeln, die dann gespeichert oder weiter verwendet werden können.
```

Listing 2.8: Antwort des Meta-Prompts

Meta-Prompts sind Token-Effizient und verringern die benötigte Anzahl an Token, da der Schwerpunkt wie beschrieben auf der Struktur liegt, nicht auf den expliziten Inhalt.

### Prompt Chaining

Hierbei wird eine komplexe Aufgabe in Unteraufgaben zerlegt. Die Antwort einer Unteraufgabe dient als Eingabe für die nächste Unteraufgabe. Diese Zerlegung ist hilfreich, um Komplexität einer Aufgabe zu verringern und eine Überforderung der Modelle zu verhindern. Durch diese Technik ist eine schrittweise Näherung an die Gesamtlösung der Aufgabe möglich.

Im Beispiel soll das Sprachmodell wieder eine PHP Funktion schreiben, die eine HTML Zeichenkette als PDF speichert.

```
prompt_string: str = """
Nenne mögliche PHP Bibliotheken, mit der PDF Dateien erstellt werden kö
nnen.
3  """
```

Listing 2.9: Chain Prompt Nr. 1 als Python-String

```
Es gibt mehrere PHP Bibliotheken, die es ermöglichen, PDF-Dateien zu
erstellen. Hier sind einige beliebte und weit verbreitete Bibliotheken
:
2
1. **TCPDF**:
   - TCPDF ist eine leistungsstarke Open-Source-Bibliothek zur Erstellung
     von PDF-Dokumenten in PHP. Es bietet umfassende Unterstützung für
     verschiedene Funktionen wie Texte, Grafiken, Tabellen und mehr.
```



```
7  - [Offizielle Website](https://tcpdf.org/)
   - [GitHub-Repository](https://github.com/tecnickcom/tc-lib-pdf)

12 2. **MPDF**:
   - MPDF ist eine weitere leistungsstarke PHP Bibliothek zur PDF-
     Generierung, die sehr beliebt ist und viele Funktionen bietet,
     einschließlich der Unterstützung für Unicode-Zeichen.
   - [Offizielle Website](https://mpdf.github.io/)
   - [GitHub-Repository](https://github.com/mpdf/mpdf)

17 3. **FPDF**:
   - FPDF ist eine einfache und leichtgewichtige Bibliothek zur PDF-
     Generierung in PHP, die auf der FPDF-Bibliothek basiert. Es ermö-
     glicht das Erstellen von PDF-Dokumenten mit Text, Grafiken und
     vielen anderen Elementen.
   - [Offizielle Website](https://tcpdf.org/downloads/)
   - [GitHub-Repository](https://github.com/fpdf/fpdf)

22 4. **DomPDF**:
   - DomPDF ist eine PHP-Implementierung von PDFLib, einer
     leistungsstarken Bibliothek zur Erstellung von PDF-Dokumenten in C
     ++. Es bietet auch die Möglichkeit, HTML in PDF umzuwandeln.
   - [Offizielle Website](https://dompdf.github.io/)
   - [GitHub-Repository](https://github.com/dompdf/dompdf)

27 5. **SnappyPdf**:
   - SnappyPdf ist eine PHP Bibliothek, die auf der wunderbaren Pdfcrowd
     API basiert und ermöglicht das Erstellen von PDF-Dokumenten aus HTML
     oder Textinhalten.
   - [Offizielle Website](https://github.com/Knplabs/snappy)
   - [GitHub-Repository](https://github.com/Knplabs/snappy)

Diese Bibliotheken bieten verschiedene Funktionen und Flexibilität, sodass
Sie je nach Anwendungsfall die passende Wahl treffen können.
```

Listing 2.10: Antwort des Chain-1-Prompts

```
2 prompt_string: str = ""
  Welche der folgenden Bibliotheken bieten die beste Unterstützung der
  Entwickler,
```

```
7 * TCPDF
  * MPDF
  * FPDF
  * DomPDF
  * SnappyPdf
  ""
```

Listing 2.11: Chain Prompt Nr. 2 als Python-String

```
2 Um zu beurteilen, welche der angegebenen Bibliotheken die beste Unterstüt-
  zung für Entwickler bietet, ist es wichtig, verschiedene Kriterien zu
  berücksichtigen. Diese können unter anderem die Funktionalität, die
  Aktualität der Dokumentation, das Umfeld der Community, die Unterstüt-
  zung von Drittanbietern und die allgemeine Benutzerfreundlichkeit
  umfassen.

  Hier ist eine kurze Bewertung einiger dieser Bibliotheken:

  1. **TCPDF**: TCPDF (HTML2FPDF) ist eine leistungsstarke PDF-Bibliothek,
    die hauptsächlich in PHP entwickelt wurde und auf HTML2FPDF basiert. Es
    bietet eine Vielzahl von Funktionen für das Erstellen und Bearbeiten
    von PDF-Dokumenten. Die Dokumentation ist recht ausführlich, und es
    gibt eine aktive Community.

  7 2. **MPDF**: MPDF (Multi-Purpose PDF Library) ist ebenfalls in PHP
    geschrieben und bietet ähnliche Funktionen wie TCPDF. Es unterstützt
    HTML sowohl im Text- als auch im Modus, was es für Entwickler attraktiv
    macht, die mit Webtechnologien arbeiten. Die Dokumentation ist gut und
    die Community ist aktiv.

  3. **FPDF**: FPDF (Free Pascal PDF) ist eine andere weit verbreitete PHP-
    Bibliothek zum Erstellen von PDF-Dokumenten. Es ist in Free Pascal
    geschrieben, aber auch viele PHP-Entwickler bevorzugen es für seine
    einfache Verwendung und die Fähigkeit, HTML zu rendern. Die
    Dokumentation ist gut verständlich.

  4. **DomPDF**: DomPDF ist eine portierte Version von MPDF in PHP, die auf
    der DOM-Basis arbeitet. Es unterstützt das Rendern von HTML und bietet
    eine hohe Flexibilität bei der Erstellung komplexer PDF-Dokumente. Die
    Dokumentation ist gut, aber die Community ist kleiner als bei anderen
```

- 12 Bibliotheken.
5. **\*\*SnappyPdf\*\***: SnappyPdf verwendet die C-Bibliothek "snappy" von Google für das Komprimieren und Dekomprimieren der PDF-Dateien. Es unterstützt viele verschiedene Backends wie Ghostscript, Qpdf oder MuPDF. Die Verwendung erfordert jedoch evtl. zusätzliche Abhängigkeiten und ist nicht in allen Sprachen verfügbar.
- \*\*Empfehlung\*\***: Wenn Sie eine Bibliothek für PHP-Projekte suchen, bei denen HTML unterstützt wird und Sie eine reaktionsfähige Unterstützung der Entwickler sowie eine aktive Community benötigen, würde ich **\*\*TCPDF\*\*** oder **\*\*MPDF\*\*** empfehlen. Beide Bibliotheken sind gut dokumentiert und haben aktive Communities. FPDF ist ebenfalls ein guter Ersatz, vor allem wegen seiner guten Leistung und der größeren Verbreitung im PHP-Umfeld.
- 17 **\*\*SnappyPdf\*\*** sollte nur dann in Betracht gezogen werden, wenn Sie speziell eine plattformübergreifende Lösung benötigen und bereits über die notwendigen Abhängigkeiten verfügen.

Listing 2.12: Antwort des Chain-2-Prompts

```
prompt_string: str = """
Erstelle eine PHP Funktion, die eine HTML Zeichenkette in einer PDF Datei
speichert. Dafür soll die PHP Bibliothek 'DomPDF' verwendet werden.
3 """
```

Listing 2.13: Chain Prompt Nr. 3 als Python-String

- Um eine PHP Funktion zu erstellen, die eine HTML-Zeichenkette in einer PDF-Datei speichert, kann die PHP Bibliothek 'DomPDF' verwendet werden. Hier ist ein Beispiel, wie Sie dies tun können:
- 2
1. Installieren Sie zuerst die DomPDF Bibliothek über Composer:
- ```
'''bash
composer require dompdf/dompdf
'''
```
- 7
2. Erstellen Sie eine PHP Funktion, die die HTML-Zeichenkette enthält und die PDF speichert:

```
12  ‘‘‘php
    <?php
require_once __DIR__ . '/vendor/autoload.php';

use Dompdf\Dompdf;

function saveHtmlToPdf($html) {
17     // Erstellen Sie eine Instanz von Dompdf
    $dompdf = new Dompdf();

    // Setzen Sie die HTML-Inhalte in Dompdf
    $dompdf->loadHtml($html);
22

    // (Optional) Konfigurieren Sie die PDF Optionen
    $dompdf->setPaper('A4', 'portrait');

    // Rendern Sie das PDF
27     $dompdf->render();

    // Speichern Sie das PDF in einer Datei
    $output = $dompdf->output();
    file_put_contents('output.pdf', $output);
32 }

// Beispiel HTML-Inhalt
$html = '<!DOCTYPE html>
<html lang="de">
37 <head>
    <meta charset="UTF-8">
    <title>Beispielseite</title>
</head>
<body>
42     <h1>Hallo Welt!</h1>
    <p>Dies ist ein Beispiel für die Verwendung von DomPDF in PHP.</p>
</body>
</html>';

47 // Rufen Sie die Funktion auf, um das HTML als PDF zu speichern
```

```
saveHtmlToPdf($html);  
?>  
'''
```

52 In diesem Beispiel:

1. Wir laden zuerst die DomPDF Bibliothek über Composer.
2. Eine PHP Funktion 'saveHtmlToPdf' wird definiert, die eine HTML-Zeichenkette als Parameter enthält.
3. Innerhalb der Funktion erstellen wir eine Instanz von Dompdf und setzen den in der Funktion übergebenen HTML-Inhalt.
4. Wir rendern das PDF und speichern es als 'output.pdf'.

57

Sie können diese Funktion aufrufen und die gewünschte HTML-Zeichenkette an sie übergeben, um eine PDF-Datei zu erstellen und zu speichern.

Listing 2.14: Antwort des Chain-3-Prompts

### Tree of Thoughts (ToT)

Diese Technik wurde von [9] und [10] vorgeschlagen. (ToT) kommt bei komplexen Anforderungen zum Einsatz, wenn einfache Techniken, die zuvor genannt wurden, nicht mehr ausreichen. Auch bei dieser Technik wird die Anforderung in keine Aufgaben zerlegt. Dann werden mehrere Lösungen pro Aufgabe erstellt und im Anschluss bewertet. Dabei entsteht eine Baustruktur, von der die besten Lösungen ausgesucht werden.

Im Folgenden ein Beispiel einer Teilaufgabe, bei der drei mögliche Lösungsvorschläge vom Modell erstellt wurden. Die Listings 2.15, 2.16 und 2.17 zeigen die Ergebnisse des Modells. Es wurden jeweils andere PHP Bibliotheken für die Lösung vorgeschlagen. Als Modell wurde *deepseek-coder-v2* verwendet. Als Nutzereingabe wurde folgendes an das Modell übergeben,

*Erstelle drei verschiedene Methoden in PHP, die eine HTML Zeichenkette in ein PDF umwandeln und es als Datei, mit angegebenen Namen, speichern.*  
Die erste Lösung beinhaltet die Bibliothek *Dompdf*.

```
<?php  
2 require 'vendor/autoload.php';  
  
use Dompdf\Dompdf;  
  
function createPDF($html, $filename)  
7 {
```

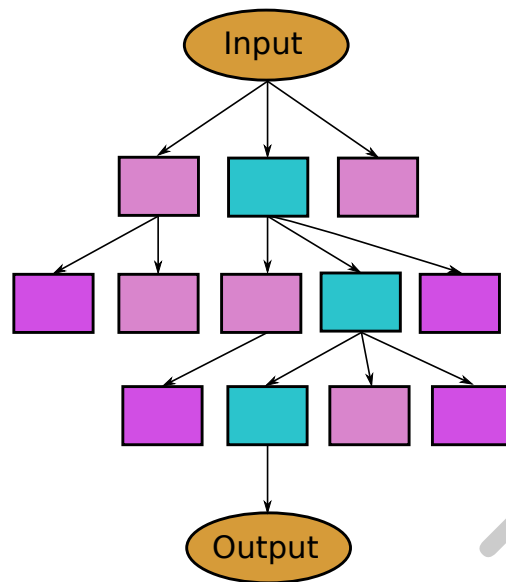


Abbildung 2.4: Baumstruktur der „Tree of Thoughts“ Technik

```

$dompdf = new Dompdf();
$dompdf->loadHtml($html);
$dompdf->setPaper('A4', 'portrait');
$dompdf->render();
12 file_put_contents($filename, $dompdf->output());
}

// ... more.

```

Listing 2.15: Ausgabe für DOMPDF Bibliothek

```

<?php
require_once('mpdf/autoload.inc.php');

use Mpdf\Mpdf;
5
function createPDF($html, $filename)
{
    $mpdf = new Mpdf();
    $mpdf->WriteHTML($html);
10 $mpdf->Output($filename, 'F');
}

```

```
// ... more.
```

Listing 2.16: Ausgabe für MPDF Bibliothek

Als dritte Ausgabe liefert coder,

```
<?php
2 require_once('tcpdf/tcpdf.php');

class CustomPDF extends TCPDF
{
    public function Header()
7    {
        $this->SetFont('helvetica', 'B', 20);
        $this->Cell(0, 15, 'Beispiel-PDF', 0, false, 'C', 0, '', 0, false, 'M'
            , 'M');
    }
}
12

function createPDF($html, $filename) {
    $pdf = new CustomPDF(PDF_PAGE_ORIENTATION, PDF_UNIT, PDF_PAGE_FORMAT,
        true, 'UTF-8', false);
    $pdf->SetDefaultMonospacedFont(PDF_FONT_MONOSPACED);
    $pdf->SetMargins(PDF_MARGIN_LEFT, PDF_MARGIN_TOP, PDF_MARGIN_RIGHT);
17 $pdf->SetHeaderMargin(PDF_MARGIN_HEADER);
    $pdf->SetFooterMargin(PDF_MARGIN_FOOTER);
    $pdf->setPrintHeader(false);
    $pdf->setPrintFooter(false);
    $pdf->AddPage();
22 $pdf->writeHTML($html, true, false, true, false, '');
    $pdf->Output($filename, 'F');
}

// ... more.
```

Listing 2.17: Ausgabe für TCPDF Bibliothek

### 2.5.2 Grenzen beim Prompt-Engineering für LLMs

Trotz der bemerkenswerten linguistischen Leistung, stoßen große Sprachmodelle an ihre Grenzen, unter anderem wie in [3] beschrieben,

## 2.6 Grundlagen bei der Entwicklung von Webanwendungen

Webanwendung

Entwurf



## STAND DER FORSCHUNG

### 3.1 Methoden und Ansätze

### 3.2 Forschungslücken und zukünftige Forschung

#### 3.2.1 Identifikation von Forschungslücken

#### 3.2.2 Zukünftige Forschungsrichtungen

Entwurf

# METHODIK

## 4.1 Auswahl der LLM

## 4.2 Prompt-Engineering

Entwurf

Entwurf

# IMPLEMENTIERUNG

## 5.1 Modelle lokal aufsetzen

Als Server dient ein Debian 12 System.

### 5.1.1 Install Ollama

Ein Skript ausführen

```
curl -fsSL https://ollama.com/install.sh | sh
```

Ein Model laden und im Anschluss starten, Beispiel

```
ollama pull deepseek-coder-v2:16b \
ollama run deepseek-coder-v2:16b
```

Config Ollama

Set correct IP and Post in /etc/systemd/system/ollama.service

```
diff --git a/ollama.service b/ollama.service
--- a/ollama.service
+++ b/ollama.service
4 @@ -10,3 +10,4 @@
RestartSec=3
Environment="PATH=/usr/local/bin:/usr/bin"
-
+ Environment="OLLAMA_HOST=0.0.0.0"
9 +
```

Listing 5.1: Ollama Hostanpassng für Netzwerkbetrieb

### 5.1.2 Open WebUI

Optional kann ein grafisches Tool, zum Testen und verwalten vom Ollama-Server im Netzwerk installiert werden. Der Aufruf der UI, kann mittel Browser erfolgen. Hier wird die IP und der Port 8080 angegeben. Beispiel `http://192.168.2.45:8080`.

```
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
-o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] \
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io \
docker-buildx-plugin docker-compose-plugin
docker run -d --network=host -v open-webui:/app/backend/data \
-e OLLAMA_BASE_URL=http://127.0.0.1:11434 --name open-webui \
--restart always ghcr.io/open-webui/open-webui:main
```

### 5.1.3 Python Client

```
pip3 install langchain
pip3 install ollama
pip3 install mistral
```

# EVALUATION

## 6.1 Einfache HTML Seite

### 6.1.1 ChatGPT 3.5

Entwurf

Entwurf



Entwurf

Entwurf

## DISKUSSION UND AUSBLICK

Entwurf

Entwurf

FAZIT

9

Entwurf

Entwurf

# LITERATUR

- [1] Erin Yepis. *Developers want more, more, more: the 2024 results from Stack Overflow's Annual Developer Survey*. 24. Juli 2024. URL: <https://stackoverflow.blog/2024/07/24/developers-want-more-more-more-the-2024-results-from-stack-overflow-s-annual-developer-survey/> (besucht am 09.08.2024).
- [2] Pekka Ala-Pietilä u. a. *Eine Definition der KI: Wichtigste Fähigkeiten und Wissenschaftsgebiete*. 5. März 2019. URL: [https://elektro.at/wp-content/uploads/2019/10/EU\\_Definition-KI.pdf](https://elektro.at/wp-content/uploads/2019/10/EU_Definition-KI.pdf) (besucht am 10.09.2024).
- [3] Xavier Amatriain. *Prompt Design and Engineering: Introduction and Advanced Methods*. 24. Jan. 2024. URL: <https://arxiv.org/abs/2401.14423v3> (besucht am 12.10.2024).
- [4] Jason Wei u. a. *Finetuned language models are Zero-Shot learners*. 3. Sep. 2021. URL: <https://arxiv.org/abs/2109.01652> (besucht am 12.10.2024).
- [5] Tom B. Brown u. a. *Language Models are Few-Shot Learners*. 28. Mai 2020. URL: <https://arxiv.org/abs/2005.14165> (besucht am 12.10.2024).
- [6] Sewon Min u. a. *Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?* 25. Feb. 2022. URL: <https://arxiv.org/abs/2202.12837> (besucht am 12.10.2024).
- [7] Jason Wei u. a. *Chain-of-Thought prompting elicits reasoning in large language models*. 28. Jan. 2022. URL: <https://arxiv.org/abs/2201.11903> (besucht am 12.10.2024).
- [8] Yifan Zhang, Yang Yuan und Andrew Chi-Chih Yao. *Meta Prompting for AI Systems*. 20. Nov. 2023. URL: <https://arxiv.org/abs/2311.11482> (besucht am 12.10.2024).
- [9] Jieyi Long. *Large language model guided Tree-of-Thought*. 15. Mai 2023. URL: <https://arxiv.org/abs/2305.08291> (besucht am 14.10.2024).

- [10] Shunyu Yao u. a. *Tree of Thoughts: Deliberate Problem Solving with Large Language Models*. 17. Mai 2023. URL: <https://arxiv.org/abs/2305.10601> (besucht am 14. 10. 2024).

Entwurf



ANHANG

Entwurf