

# Generative KI-Systeme entwickeln

KI-Engineering für die Praxis – vom Prompt  
Engineering bis zu RAG und Agenten

# DAS INHALTS- VERZEICHNIS

» Hier geht's  
direkt  
zum Buch

<b>Vorwort</b> .....	<b>13</b>
<b>1 Einführung in das Bauen von KI-Anwendungen mit Foundation Models</b> .....	<b>25</b>
Der Aufstieg des AI Engineering .....	26
Von Sprachmodellen zu Large Language Models .....	26
Von Large Language Models zu Foundation Models .....	32
Von Foundation Models zum AI Engineering .....	36
Anwendungsfälle für Foundation Models .....	40
Coding .....	43
Bild- und Videoproduktion .....	45
Texten .....	46
Lernen .....	48
Dialog-Bots .....	49
Informationsaggregation .....	50
Datenorganisation .....	51
Workflow-Automation .....	51
KI-Anwendungen planen .....	52
Evaluation des Anwendungsfalls .....	52
Erwartungen setzen .....	56
Meilensteinplanung .....	57
Wartung .....	57
Der AI-Engineering-Stack .....	59
Drei Schichten des KI-Stacks .....	61
AI Engineering versus ML Engineering .....	63
AI Engineering versus Full-Stack Engineering .....	70
Zusammenfassung .....	71
<b>2 Foundation Models verstehen</b> .....	<b>73</b>
Trainingsdaten .....	74
Mehrsprachige Modelle .....	76
Domänenspezifische Modelle .....	79

Modellieren	81
Modellarchitektur	82
Modellgröße	90
Post-Training	100
Supervised Finetuning	103
Preference Finetuning	106
Sampling	111
Grundlagen des Samplings	111
Sampling-Strategien	113
Test Time Compute	118
Strukturierte Ausgaben	121
Die statistische Natur der KI	126
Zusammenfassung	133
<b>3 Evaluierungsmethoden</b>	<b>135</b>
Herausforderungen beim Evaluieren von Foundation Models	136
Die Metriken von Sprachmodellen verstehen	140
Entropie	141
Kreuzentropie	142
Bits-per-Character und Bits-per-Byte	143
Perplexität	143
Interpretation und Anwendungsfälle für die Perplexität	144
Exakte Evaluation	147
Funktionale Korrektheit	147
Ähnlichkeitsmessung gegen Referenzdaten	149
Einführung in Embeddings	155
AI-as-a-Judge	157
Warum AI-as-a-Judge?	158
Wie Sie AI-as-a-Judge verwenden	159
Grenzen eines AI-as-a-Judge	162
Welche Modelle können als AI Judges agieren?	167
Modelle durch vergleichende Evaluation einstufen	170
Herausforderungen der vergleichenden Evaluation	173
Die Zukunft der vergleichenden Evaluierung	177
Zusammenfassung	177
<b>4 KI-Systeme evaluieren</b>	<b>179</b>
Evaluierungskriterien	180
Domänenspezifische Fähigkeiten	181
Generierungsfähigkeiten	184
Fähigkeit zum Befolgen von Anweisungen	192
Kosten und Latenz	197

Modellauswahl	199
Modellauswahl-Workflow	199
Modell bauen oder kaufen?	201
Öffentliche Benchmarks nutzen	212
Ihre Evaluierungs-Pipeline entwerfen	221
Schritt 1: Alle Komponenten in einem System evaluieren	221
Schritt 2: Eine Evaluierungsrichtlinie erstellen	223
Schritt 3: Methoden und Daten zur Evaluation definieren	225
Zusammenfassung	230
<b>5 Prompt Engineering</b>	<b>233</b>
Einführung in das Prompting	234
In-Context Learning: Zero-Shot und Few-Shot	235
System-Prompt und User-Prompt	237
Kontextlänge und Kontexteffizienz	240
Best Practices beim Prompt Engineering	242
Klare und explizite Anweisungen schreiben	242
Ausreichend Kontext bereitstellen	245
Komplexe Aufgaben in einfachere Unteraufgaben aufteilen	246
Dem Modell Zeit zum Denken geben	249
Über Ihre Prompts iterieren	250
Evaluieren Sie Prompt-Engineering-Tools	251
Prompts organisieren und versionieren	254
Defensive Prompt Engineering	256
Proprietäre Prompts und Reverse Prompt Engineering	257
Jailbreaking und Prompt Injection	259
Informationsextraktion	264
Verteidigung gegen Prompt-Angriffe	268
Zusammenfassung	272
<b>6 RAG und Agenten</b>	<b>275</b>
RAG	275
RAG-Architektur	278
Retrieval-Algorithmen	279
Retrieval-Optimierung	290
RAG für mehr als Texte	295
Agenten	298
Überblick über Agenten	298
Tools	301
Planung	304
Fehlerzustände von Agenten und deren Evaluation	320
Memory	323
Zusammenfassung	327

<b>7</b>	<b>Optimieren</b>	<b>329</b>
	Überblick über das Optimieren	330
	Wann man optimiert	333
	Gründe für das Optimieren	333
	Gründe gegen das Optimieren	335
	Optimieren und RAG	338
	Speicherengpässe	342
	Backpropagation und trainierbare Parameter	343
	Speichermathematik	345
	Numerische Repräsentationen	347
	Quantisierung	350
	Optimierungstechniken	354
	Parametereffizientes Optimieren	355
	Model Merging und Multi-Task Finetuning	368
	Optimierungstaktiken	378
	Zusammenfassung	382
<b>8</b>	<b>Dataset Engineering</b>	<b>385</b>
	Kuratieren von Daten	387
	Datenqualität	390
	Datenabdeckung	392
	Datenquantität	395
	Datenbeschaffung und Datenannotation	399
	Datenaugmentation und Datensynthese	402
	Warum Datensynthese?	403
	Klassische Techniken der Datensynthese	404
	KI-gestützte Datensynthese	408
	Modelldestillation	417
	Datenverarbeitung	419
	Daten inspizieren	419
	Daten deduplizieren	421
	Daten bereinigen und filtern	423
	Daten formatieren	424
	Zusammenfassung	425
<b>9</b>	<b>Inferenzoptimierung</b>	<b>427</b>
	Inferenzoptimierung verstehen	428
	Überblick über die Inferenz	428
	Metriken zur Inferenzperformance	434
	KI-Beschleuniger	440
	Inferenzoptimierung	447
	Modelloptimierung	448
	Optimieren des Inferenzservice	461
	Zusammenfassung	468

<b>10 Architektur beim AI Engineering und User-Feedback</b> .....	<b>471</b>
Architektur beim AI Engineering .....	471
Schritt 1: Kontext erweitern .....	472
Schritt 2: Leitplanken installieren .....	473
Schritt 3: Router und Gateways für das Modell hinzufügen .....	478
Schritt 4: Latenz durch Caches verringern .....	483
Schritt 5: Agenten-Patterns hinzufügen .....	485
Monitoring und Observability .....	487
KI-Pipeline-Orchestrierung .....	493
User-Feedback .....	495
Dialog-Feedback extrahieren .....	496
Feedback-Design .....	502
Grenzen des Feedbacks .....	510
Zusammenfassung .....	513
<b>Epilog</b> .....	<b>515</b>
<b>Index</b> .....	<b>516</b>