

A

- accelerators, 419-425
 - computational capabilities, 422
 - defined, 420-421
 - memory size and bandwidth, 422-424
 - power consumption, 424-425
- active injection, 243
- adapter-based methods, 336
- adapters
 - finetuning, 358
 - LoRA, 338-347
 - merging with concatenation, 356
 - PEFT techniques, 336-338
- agents, 275-300
 - agent failure modes and evaluation, 298-300
 - efficiency, 300
 - planning failures, 298
 - tool failures, 299
 - overview, 276-278
 - planning agents, 281-298
 - foundation models as planners, 284-286
 - overview, 282-284
 - plan generation, 286-292
 - reflection and error correction, 292-294
 - tool selection, 295-298
 - tools, 278-281
 - capability extension, 279
 - knowledge augmentation, 279
 - write actions, 280
- AI accelerators (see accelerators)
- AI application building (see application building)
- AI application planning (see application planning)
- AI engineering (AIE)
 - defined, 12
 - ML engineering versus, 39-46
 - rise of AI engineering, 2-14
- AI engineering architecture (see engineering architecture)
- AI engineering stack (see engineering stack)
- AI judge, 136
 - (see also AI-as-a-judge)
- AI pipeline orchestration (see pipeline orchestration)
- AI systems evaluation (see systems evaluation)
- AI-as-a-judge, 136-148
 - limitations, 141-145
 - biases, 144
 - criteria ambiguity, 142-144
 - inconsistency, 142
 - increased costs and latency, 144
 - models, 145-148
 - reasons, 137
 - reference-based, 147
 - uses, 138-141
- AI-powered data synthesis (see data synthesis, AI-powered)
- AMP (automatic mixed precision), 332
- ANN (approximate nearest neighbor), 262
- Annoy (approximate nearest neighbors oh yeah), 263
- anomaly detection, 129
- Anthropic
 - contextual retrieval, 271
 - inverse scaling and alignment training, 71
 - prompt caching, 444
 - RAG and, 256

- APIs (see open source models, model APIs ver-
- sus)
- application building, 1-48
 - application planning, 28-35
 - maintenance, 34
 - milestone planning, 33
 - set expectations, 32
 - use case evaluation, 29-32
 - engineering stack, 35-47
 - AI engineering versus ML engineering, 39-46
 - application development, 44-46
 - full-stack engineering versus, 46
 - three layers of AI stack, 37-39
 - foundation model use cases, 16-28
 - coding, 20-22
 - conversational bots, 26
 - data organization, 27
 - education, 24
 - image and video production, 22
 - information aggregation, 26
 - workflow automation, 28
 - writing, 22-24
 - rise of AI engineering, 2-14
 - foundation models to AI engineering, 12-14
- application development, 37, 44-46
 - AI interface, 45
 - evaluation, 44
 - prompt engineering and context construc-
 - tion, 45
- application planning, 28-35
 - maintenance, 34
 - milestone planning, 33
 - set expectations, 32
 - use case evaluation, 29-32
- approximate nearest neighbor (ANN), 262
- approximate string matching, 130
- ARC-C, 192
- attention mechanisms, 60-62
 - attention modules, 62
 - MLP modules, 62
 - optimization, 433-436
 - attention mechanism redesign, 435
 - wiring kernels for attention computa-
 - tion, 436
 - redesign, 435
- attention modules, 62
- augmentation of data
 - defined, 380
 - automated attacks, 240
 - automatic mixed precision (AMP), 332
 - autoregressive decoding bottleneck, 428-433
 - inference with reference, 430
 - parallel decoding, 432
 - speculative decoding, 428-430
 - autoregressive language model, 4

B

- backpropagation, 320-322
- batch inference APIs, 410-412
- batch size, 360
- batching
 - batch inference APIs, 410-412
 - batch size, 360
 - continuous, 441
 - dynamic, 441
 - static, 440
- benchmarks
 - for comparative evaluation, 155
 - data contamination detection, 124
 - domain distribution and, 56
 - domain-specific, 161-163
 - instruction-following criteria, 173-175
 - model-centric versus data-centric, 364
 - navigating public benchmarks, 191-197
- biases, 144, 490
- bits-per-byte (BPB), 121
- bits-per-character (BPC), 121
- bottlenecks
 - autoregressive decoding, 428-433
 - computational, 407-410
 - compute-bound, 407
 - memory, 319-332, 407
 - scaling, 75-77, 152
- BPB (bits-per-byte), 121
- BPC (bits-per-character), 121
- build time, 266

C

- canonical responses, 127
- capability extension, 279
- chain-of-thought (CoT), 227-229, 365
- chaining, 473
- change failure rate (CFR), 466
- CharacterEval, 176
- ChatGPT
 - comparative evaluation, 149

- data privacy issues, 184
 - effect on AI investment, 13
 - Gemini versus, 44
 - hallucinations, 107
 - and human writing quality, 23
 - introduction of, xi
 - and languages other than English, 55
 - query rewriting, 270
 - reverse prompt engineering attacks, 237
 - in schools, 24
 - Chinchilla scaling law, 72
 - chunking, 257, 268-269
 - Claude, RAG and, 256
 - CLIP, 10, 56, 135
 - clustering, 129
 - Common Crawl dataset, 50-55
 - comparative evaluation, 148-156
 - comparison data, 85
 - compilers, 438
 - components definition, 472
 - computational bottlenecks, 407-410
 - computational capabilities, of AI accelerators, 422
 - compute-bound bottlenecks, 407
 - compute-optimal models, 72-74
 - compute-optimal training, 72
 - concatenation, 356
 - constrained sampling, 103
 - context construction, 45, 224, 451
 - context efficiency, 218-220
 - context length, 218-220
 - context parallelism, 447
 - context precision, 264
 - context recall, 264
 - contextual retrieval, 271-272
 - continuous batching, 441
 - control flow, 291
 - conversational bots, 26
 - conversational feedback
 - conversation length, 480
 - conversation organization, 479
 - extracting, 475-480
 - language diversity, 480
 - natural language feedback, 476-479
 - complaints, 478
 - early termination, 476
 - error correction, 477
 - sentiment, 478
 - regeneration, 479
 - copyright regurgitation, 246
 - copyright, model training and, 185
 - CoT (chain-of-thought), 227-229
 - CPU memory (DRAM), 423
 - criteria ambiguity, 142-144
 - cross entropy, 120
 - cross-layer attention, 435
- ## D
- data annotation, 377-380
 - and data curation, 365-380
 - and data inspection, 398
 - dataset engineering and, 42
 - data augmentation, 380-396
 - defined, 380
 - data cleaning/filtering, 401
 - data contamination, 197-200
 - data coverage, 369-371
 - data curation, 365-380
 - data deduplication, 129, 399-400
 - data flywheels, 377
 - data formatting, 401-403
 - data inspection, 397-399
 - data lineage, 185
 - data organization, 27
 - data privacy, 184
 - data processing, 396-403
 - data cleaning/filtering, 401
 - data formatting, 401-403
 - deduplicating data, 399-400
 - inspecting data, 397-399
 - data synthesis, 380-396
 - AI-powered, 386-395
 - data verification, 391-393
 - instruction data synthesis, 388-391
 - limitations, 393-395
 - obscure data lineage problems, 395
 - potential model collapse, 394
 - quality control problems, 393
 - reasons for synthesizing data, 381-382
 - superficial imitation problems, 393
 - model distillation, 395
 - traditional techniques, 383-386
 - rule-based, 383-385
 - simulation, 385
 - data verification, 391-393
 - dataset engineering, 42, 363-404
 - data augmentation/synthesis, 380-396
 - data curation, 365-380

- data acquisition/annotation, 377-380
- data coverage, 369-371
- data quality, 368-369
- data quantity, 372-377
- data processing, 396-403
 - data cleaning and filtering, 401
 - data formatting, 401-403
 - deduplicating data, 399-400
 - inspecting data, 397-399
- data-centric view of AI, 364
- DDR SDRAM (doubled data rate synchronous dynamic random-access memory), 423
- debugging, 226
- decoding
 - autoregressive decoding bottleneck, 428-433
 - decoupling from prefilling, 442
 - in transformer architecture, 58
- defensive prompt engineering
 - jailbreaking and prompt injection, 238-243
 - automated attacks, 240
 - direct manual prompt hacking, 239-240
 - indirect prompt injection, 242-243
 - prompt attack defense, 248-251
 - model-level defense, 248
 - prompt-level defense, 249
 - system-level defense, 250
- degenerate feedback loops, 491
- demonstration data, 81
- dense retrievers, 258
- dimensionality reduction, 400
- direct manual prompt hacking, 239-240
- Direct Preference Optimization (DPO), 84
- distillation, 312
 - base, 358
 - model distillation, 182, 395, 427
 - synthetic data and, 382
- domain-specific capability, 161-163
- domain-specific task finetuning, 314
- domain-specific training data models, 56-57
- dot products, 61
- doubled data rate synchronous dynamic random-access memory (DDR SDRAM), 423
- DPO (Direct Preference Optimization), 84
- DRAM (CPU memory), 423
- drift detection, 471
- dynamic batching, 441
- dynamic features, 30

E

- edit distance, 130
- Elo, 151, 152, 346
- embedding, 134-136
- embedding algorithm, 133, 135
- embedding model, 10
 - embedding-based retrieval, 260-263
 - multimodal RAG and, 273
- embedding models, 134
- engineering architecture, 449-474
 - AI pipeline orchestration, 472-474
 - monitoring and observability, 465-472
 - drift detection, 471
 - logs and traces, 469-470
 - metrics, 467-469
 - monitoring versus observability, 466
 - step 1: enhancing context, 450
 - step 2: putting in guardrails, 451-455
 - guardrail implementation, 455
 - input guardrails, 451-452
 - output guardrails, 453-454
 - step 3: adding model router and gateway, 456-460
 - gateway, 458-460
 - router, 456-457
 - step 4: reducing latency with caches, 460-463
 - exact caching, 461
 - semantic caching, 461
 - step 5: adding agent patterns, 463
- engineering stack, 37-39
 - application development, 37
 - AI interface, 45
 - evaluation, 44
 - prompt engineering and context construction, 45
 - infrastructure, 37
 - ML engineering versus, 40-44
 - model development, 37
- entropy, 119
- epochs, 360
- error correction, 292-294
- evaluation, 44
- evaluation harnesses, 191
- evaluation methodology, 113-157
 - AI as a judge, 136-148
 - AI systems evaluation (see systems evaluation)
 - challenges, 152-155

- challenges of foundation model evaluation, 114-117
 - comparative performance to absolute performance, 154
 - lack of standardization and quality control, 153-154
 - scalability bottlenecks, 152
 - exact evaluation, 125-136
 - future, 155
 - language model for computing text perplexity, 125
 - language modeling metrics, 118-124
 - rank models with comparative evaluation, 148-156
 - evaluation pipeline design, 200-208
 - step 1: creating an evaluation guideline, 202-203
 - step 2: evaluating all components in a system, 200-201
 - creating scoring rubrics with examples, 202
 - defining evaluation criteria, 202
 - tying evaluation metrics to business metrics, 203
 - step 3: defining evaluation methods and data, 204-208
 - annotating evaluation data, 205-207
 - evaluating evaluation pipeline, 207
 - iteration, 208
 - selecting evaluation methods, 204
 - evaluation-driven development, 160-161
 - eviction policies, 461
 - exact caching, 461
 - exact evaluation, 125-136
 - functional correctness, 126-127
 - similarity measurements against reference data, 127-133
 - exact matches, 129
 - expectation setting, 32
 - explicit feedback, 475-480
- ## F
- factual consistency, 165-169, 202
 - faithfulness, 164
 - feature-based transfers, 104, 309
 - feature-free transfers, 104
 - federated learning, 348
 - feedback design
 - how to collect feedback, 485-489
 - when to collect feedback
 - in the beginning, 481
 - when something bad happens, 481
 - when the model has low confidence, 483-485
 - feedforward computation, 447
 - feedforward layer, 62, 343
 - few-shot learning, 213-215
 - finetuning, 307-362
 - defined, 42
 - domain-specific tasks, 314
 - finetuning and RAG, 316-319
 - hyperparameters, 359-361
 - batch size, 360
 - learning rate, 359
 - number of epochs, 360
 - prompt loss rate, 361
 - memory bottlenecks, 319-332
 - backpropagation and trainable parameters, 320-322
 - memory math, 322-324
 - numerical representations, 325-328
 - quantization, 328-332
 - overview, 308-311
 - structured outputs, 104
 - tactics, 357-361
 - techniques, 332-361
 - LoRA, 338-347
 - model merging and multi-task finetuning, 347-357
 - parameter-efficient finetuning, 332-347
 - PEFT techniques, 336-338
 - when to finetune, 311-319
 - reasons not to finetune, 312-315
 - reasons to finetune, 311
 - FLOP (floating point operation), 70
 - foundation models, 12, 49-112
 - evaluation challenges, 114-117
 - comparative performance to absolute performance, 154
 - lack of standardization and quality control, 153-154
 - scalability bottlenecks, 152
 - inverse scaling, 71
 - modeling, 58-77
 - model architecture, 58-66
 - model size, 67-77
 - parameter versus hyperparameter, 74
 - post-training, 78-88

- preference finetuning, 83-88
 - supervised finetuning, 80-83
- sampling, 88-111
 - probabilistic nature of AI, 105-111
 - sampling fundamentals, 88-90
 - sampling strategies, 90-95
 - structured outputs, 99-104
 - test time compute, 96-99
- training data, 50-57
 - domain-specific models, 56-57
 - multilingual models, 51-55
- use cases, 16-28
 - coding, 20-22
 - conversational bots, 26
 - data organization, 27
 - education, 24
 - image and video production, 22
 - workflow automation, 28
 - writing, 22-24
- full finetuning, 332-347
- function calling, 288-290
- fuzzy matching, 130

G

- gateways, 458-460
- Gemini, 44, 99, 444, 483
- generation capability, 163-172
- global factual consistency, 165
- goodput, 414-415
- GPU on-chip SRAM, 423
- ground truths, 127
- grouped-query attention, 436
- guardrail implementation, 455
- guardrails, 189, 251, 451-455

H

- H3 architecture, 66
- hallucinations
 - causes of, 107-111
 - defined, 105
 - and finetuning, 317
 - measurement, 166
 - metrics for, 467
 - superficial imitation and, 393
- hard attributes, 179
- hashing, 400
- HellaSwag, 192
- hierarchical navigable small world (HNSW), 263

- high-bandwidth memory (HBM), 423
- hyperparameters, 74, 359-361

I

- IDF (inverse document frequency), 259
- IFEval, 174
- implicit feedback, 475
- in-context learning, 213-215
- inconsistency, 106-107, 142
- indexing
 - chunking strategy and, 268-269
 - defined, 256
 - with embedding-based retrieval, 261
 - retrieval systems and, 266
- indirect prompt injection, 242-243
- inference APIs, 410-412
- inference optimization, 43, 405-448
 - AI accelerators
 - computational capabilities, 422
 - defined, 420-421
 - memory size and bandwidth, 422-424
 - power consumption, 424-425
 - case study from PyTorch, 439
 - inference overview
 - computational bottlenecks, 407-410
 - online and batch inference APIs, 410-412
 - inference performance metrics, 412-419
 - latency, TTFT, and TPOT, 412-414
 - throughput/goodput, 414-415
 - utilization, MFU, and MBU, 416-419
 - inference service optimization, 440-447
 - batching, 440
 - decoupling prefill and decode, 442
 - parallelism, 444-447
 - prompt caching, 443-444
 - KV cache size calculation, 435
 - memory-bound versus bandwidth-bound interference, 408
 - at model/hardware/service levels, 426
 - model optimization, 426-439
 - attention mechanism optimization, 433-436
 - autoregressive decoding bottleneck, 428-433
 - kernels and compilers, 437-440
 - model compression, 427
 - understanding, 406-425
 - AI accelerators, 419-425

- inference overview, 406-412
- inference performance metrics, 412-419
- inference performance metrics, 412-419
 - latency, TTFT, and TPOT, 412-414
 - throughput/goodput, 414-415
 - utilization, MFU, and MBU, 416-419
- inference quantization, 329-331
- inference service
 - defined, 183
 - and inference optimization, 406
 - throughput/goodput, 414-415
- inference service optimization, 440-447
 - decoupling prefill and decode, 442
 - parallelism, 444-447
 - prompt caching, 443-444
- inference with reference, 430
- INFOBench, 174
- information aggregation, 26
- information extraction, 243-247
- information retrieval optimization, 267-272
 - chunking strategy, 268-269
 - contextual retrieval, 271-272
 - query rewriting, 270
 - reranking, 269
- instruction data synthesis, 388-391
- instruction-following capability, 172-177
- instruction-following criteria, 173-175
- intent classifiers, 457
- inter-token latency (ITL), 413
- interface, AI, 45
- internal knowledge, 301
- inverse document frequency (IDF), 259
- inverted file index (IVF), 263
- iteration, 208

J

- jailbreaking, 238-243
 - automated attacks, 240
 - direct manual prompt hacking, 239-240
 - indirect prompt injection, 242-243
- Jamba architecture, 66
- judges (see AI judges)

K

- k-nearest neighbors (k-NN), 262
- kernels, 436, 437-440
- key vector (K), 60
- key-value (KV) cache, 433-436
- key-value vectors, 323

- knowledge augmentation, 279
- knowledge-augmented verification, 167
- KV cache (see key-value cache)

L

- LangChain, 232, 250, 303
- language modeling metrics, 118-124
 - bits-per-byte, 121
 - bits-per-character, 121
 - cross entropy, 120
 - entropy, 119
 - perplexity, 121
 - perplexity interpretation and use cases, 122-124
- language models, 2-6, 125
- large language models, 8-12
 - AI product defensibility, 31
 - role of AI and humans in the application, 30-31
 - set expectations, 31
- large multimodal model (LMM), 9
- latency
 - AI judges and, 144
 - inference performance and, 412-414
 - metrics, 33
 - reliability versus, 455
- layer stacking, 354-355
- leaderboards, 152-154, 191-197
- learning rate, 359
- leniency bias, 490
- lexical similarity, 130-131
- linear combination summing, 350-352
- Llama
 - attention function, 62
 - data coverage, 370
 - data quality, 368
 - data quantity, 372
 - data synthesis, 387, 390
 - finetuning, 310
 - inference optimization, 439
 - inference quantization, 330
 - model distillation, 395
 - open source models, 182
 - prefer, 84
 - preference finetuning, 78
 - prompt template, 215
 - scaling law and, 73
- LLM-as-a-judge, 136
 - (see also AI-as-a-judge)

- LMM (large multimodal model), 9
- local factual consistency, 165
- locality-sensitive hashing (LSH), 263
- logit vectors, 89
- logprobs, 93, 204
- logs, 469-470
- long-term memory, 301
- loop tiling, 438
- LoRA (low-rank adaptation), 338-347
 - configurations, 341-343
 - LoRA adapters service, 343-345
 - mechanism of operation, 340
 - quantized LoRA (QLoRA), 345-347
- low-rank factorization, 340
- LSH (locality-sensitive hashing), 263

M

- Mamba architecture, 66
- manual generation, 383-386
- masked language models, 4
- Massive Multitask Language Understanding (MMLU), 34, 192
- matches, 150
- MBU (model bandwidth utilization), 416-419
- MCQs (multiple-choice questions), 163
- mean time to detection (MTTD), 466
- mean time to response (MTTR), 466
- memory, 300-304
 - internal knowledge, 301
 - long-term memory, 301
 - short-term memory, 301
- memory bottlenecks, 319-332
 - bandwidth-bound, 407
 - memory math, 322-324
 - memory needed for inference, 323
 - memory needed for training, 323-324
 - quantization, 328-332
 - inference quantization, 329-331
 - training quantization, 331-332
 - size and bandwidth, 422-424
- memory math, 322-324
- metrics, 467-469
 - correlations between, 208
 - for AI as a judge, 142-144
 - for generation capability, 163
 - for hallucination measurement, 166
 - inference performance metrics, 412-419
 - language modeling (see language modeling metrics)

- observability metrics, 466
- reference-based versus reference-free, 127
- tying evaluation metrics to business metrics, 203
- usefulness thresholds, 33
- MFU (model FLOPs utilization), 416-419
- milestone planning, 33
- mixture-of-experts (MoE) models, 68, 354
- ML engineering, AI engineering versus, 39-46
- MLP modules, 62
- MMLU (Massive Multitask Language Understanding), 34, 192
- model APIs, open source models versus (see open source models, model APIs versus)
- model architecture, 58-66
 - (see also specific architectures, e.g.: transformer architecture)
- model bandwidth utilization (MBU), 416-419
- model compression, 427
- model development, 37, 40-44
 - dataset engineering, 42
 - inference optimization, 43-44
 - modeling and training, 41-42
- model distillation, 395
- model FLOPs utilization (MFU), 416-419
- model inference, 34
- model merging, 347-357
 - concatenation, 356
 - layer stacking, 354-355
 - summing, 350-354
- model optimization, 426-439
 - attention mechanism optimization, 433-436
 - attention mechanism redesign, 435
 - KV cache size optimization, 436
 - write kernels for attention computation, 436
 - autoregressive decoding bottleneck, 428-433
 - inference with reference, 430
 - parallel decoding, 432
 - speculative decoding, 428-430
 - kernels and compilers, 437-440
 - model compression, 427
- model ranking, 148-156
- model router, 456-460
- model selection, 179-200
 - model build versus buy, 181-191
 - open source models versus model APIs, 183-191

- open source, open weight, and model licenses, 181-183
- model selection workflow, 179-181
- navigating public benchmarks, 191-197
 - benchmark selection and aggregation, 191
 - public leaderboards, 192
- model size, 67-77
 - scaling bottlenecks, 75-77
 - scaling extrapolation, 74
 - scaling law: building compute-optimal models, 72-74
- model-centric AI, 364
- model-level defense, 248
- modeling, 58-77
 - model architecture, 58-66
 - model size, 67-77
- MoE (mixture-of-experts) models, 354
- monitoring, 226, 465-472
- MTTD (mean time to detection), 466
- MTTR (mean time to response), 466
- multi-query attention, 435
- multi-task finetuning, 347
- multilingual training data models, 51-55
- multimodal models, 9
- multiple-choice questions (MCQs), 163

N

- n-gram similarity, 131
- natural language feedback, 476-479
 - complaints, 478
 - early termination, 476
 - error correction, 477
 - sentiment, 478
- natural language generation (NLG), 163-172
- natural language processing (NLP), 163-172
- needle in a haystack (NIAH) test, 218

O

- obscure data lineage, 395
- observability, 465-472
- on-device deployment, 190
- online inference APIs, 410-412
- Open CLIP, 56
- open source licenses, 181-183
- open source models, model APIs versus, 183-191
 - API cost versus engineering cost, 188
 - control, access, and transparency, 189

- data lineage and copyright, 185
- data privacy, 184
- functionality, 187
- on-device deployment, 190
- performance, 186
- open weight models, 182
- OpenAI
 - batch APIs, 410
 - evaluation harnesses, 191
 - first GPT model, 8
 - instruction hierarchy for model-level defense, 248
 - model as a service, 14
 - natural language supervision, 10
 - open source APIs, 183
 - progression/distillation paths, 357
 - quality of updated models, 196
 - test time compute, 97
- operator fusion, 438
- optimization
 - inference optimization (see inference optimization)
 - of retrieval systems, 267-272

P

- pairwise comparison, 400
- parallel decoding, 432
- parallelism, 444-447
- parallelization, 226, 438
- parameter-efficient finetuning, 332-347
 - adapter-based/soft-prompt techniques, 336-338
 - LoRA, 338-347
 - configurations, 341-343
 - how it works, 340
 - LoRA adapters service, 343-345
 - quantized LoRA, 345-347
- Pareto optimization, 177
- partial finetuning, 333
- passive phishing, 242
- PEFT (see parameter-efficient finetuning)
- perplexity, 121-124
- perturbation, 385
- pipeline orchestration, 472-474
 - monitoring and observability, 465-472
 - drift detection, 471
 - logs and traces, 469-470
 - metrics, 467-469
- planning

- plan generation, 286-292
 - complex plans, 291
 - function calling, 288-290
 - granularity, 290
 - reflection and error correction, 292-294
- pointwise evaluation, 84, 148
- position bias, 491
- post-processing, 102
- post-training, 42, 78-88
 - preference finetuning, 83-88
 - supervised finetuning, 80-83
- potential model collapse, 394
- power consumption, 424-425
- PPO (proximal policy optimization), 87
- pre-training, 41
- precision bits, 326
- preference bias, 491
- preference finetuning, 83-88, 309
- preference models, 147
- prefilling, 60
- prefilling, decoupling from decoding, 442
- proactive features, 30
- probabilistic nature of AI, 105-111
 - hallucination, 107-111
 - inconsistency, 106-107
 - probabilistic definition, 105-111
- procedural generation, 383-386
- product quantization, 263
- prompt attacks, 235, 238-243
 - automated attacks, 240
 - defense against, 248-251
 - direct manual prompt hacking, 239-240
 - indirect prompt injection, 242-243
- prompt caching, 443-444
- prompt catalogs, 235
- prompt engineering, 211-252
 - basics, 212-220
 - context length and context efficiency, 218-220
 - in-context learning: zero-shot and few-shot, 213-215
- best practices, 220-235
 - break complex tasks into simpler sub-tasks, 224-227
 - evaluating prompt engineering tools, 230-233
 - give the model time to think, 227-229
 - iterating on your prompts, 229
 - organize and version prompts, 233-235

- provide sufficient context, 223
- write clear and explicit instructions, 220
- defensive engineering, 235-251
 - information extraction, 243-247
 - jailbreaking and prompt injection, 238-243
 - prompt attacks defense, 248-251
 - proprietary prompts and reverse prompt engineering, 236-238
- defined, 45
- restricting model knowledge to its context, 224
- terminology ambiguity: prompt versus context, 214
- prompt loss rate, 361
- prompt optimization, 230
- prompt versioning, 233-235
- prompt-level defense, 249
- proprietary prompts, 236-238
- proximal policy optimization (PPO), 87
- public leaderboards, 192

Q

- QAT (quantization-aware training), 331
- QLoRA (quantized LoRA), 345-347
- QPS (queries per second), 266
- quality control, 393
- quantization, 328-332
 - inference quantization, 329-331
 - training quantization, 331-332
- quantization-aware training (QAT), 331
- quantized LoRA (QLoRA), 345-347
- queries per second (QPS), 266
- query rewriting, 270
- query vector (Q), 60

R

- RAG (retrieval-augmented generation), 253-275
 - finetuning and, 316-319
- RAG architecture, 256
- RAG beyond texts, 273-275
 - multimodal RAG, 273
 - RAG with tabular data, 274-275
- retrieval algorithms, 257-267
 - combining, 266
 - comparing, 264-266
 - embedding-based retrieval, 260-263
 - term-based retrieval, 258-260

- retrieval optimization, 267-272
 - chunking strategy, 268-269
 - contextual retrieval, 271-272
 - query rewriting, 270
 - reranking, 269
- random feedback, 491
- range bits, 326
- ranking, 129
- rating algorithms, 151
- reactive features, 30
- recall, 266
- recurrent neural networks (RNNs), 58
- reference-based judges, 147
- reference-based metrics, 127
- reference-free metrics, 127
- reflection, 292-294
- regeneration, 479
- reinforcement learning from human feedback (RLHF), 83-88
- relevance, 164
- reliability, latency versus, 455
- replica parallelism, 445
- reranking, 269
- restricted weight, 183
- retrieval algorithms, 257-267
 - combining, 266
 - comparing, 264-266
 - embedding-based retrieval, 260-263
 - term-based retrieval, 258-260
- retrieval optimization
 - chunking strategy, 268-269
 - contextual retrieval, 271-272
 - query rewriting, 270
 - reranking, 269
- retrieval-augmented generation (see RAG)
- retrievers
 - combining retrieval algorithms, 266
 - main functions, 256
 - multimodal RAG and, 273
 - quality evaluation, 264
 - sparse versus dense, 258
- reverse prompt engineering, 236-238
- reward models, 84-87, 147
- RLHF (reinforcement learning from human feedback), 83-88
- RNNs (recurrent neural networks), 58
- RoleLLM, 176
- roleplaying, 175-177
- routers, 456-457
- rule-based data synthesis, 383-385

S

- S4 architecture, 66
- safety, 170-172
- safety, as evaluation criteria, 170-172
- sampling, 88-111
 - probabilistic nature of AI, 105-111
 - sampling fundamentals, 88-90
 - sampling strategies, 90-95
 - strategies, 90-95
 - stopping condition, 95
 - temperature, 90-93
 - top-k, 94
 - top-p, 94
 - structured outputs, 99-104
 - test time compute, 96-99
- scaling bottlenecks, 75-77, 152
- scaling extrapolation, 74
- scaling law, 72-74
- scoring rubrics, 202
- self-evaluation, 146
- self-supervision language models, 6-8
- self-verification, 167
- semantic caching, 461
- semantic similarity, 132-133
- sequence parallelism, 447
- sequential finetuning, 348
- SFT (supervised finetuning), 78, 80-83, 309
- short-term memory, 301
- simulation, 385
- simultaneous finetuning, 347
- SLERP (spherical linear interpolation), 352
- slicing, 205
- soft attributes, 179
- soft prompt-based PEFT methods, 336-338
- sparse models, 68, 427
- sparse retrievers, 258
- speculative decoding, 428-430
- spherical linear interpolation (SLERP), 352
- SQL queries, 277
- static batching, 440
- static features, 30
- stopping condition, 95
- structured data, 123, 303
- structured outputs, 99-104
 - constrained sampling, 103
 - finetuning, 104
 - post-processing, 102

- summing, 350-354
 - linear combination, 350-352
 - pruning redundant task-specific parameters, 353
 - spherical linear interpolation (SLERP), 352
- superficial imitation, 393
- supervised finetuning (SFT), 78, 80-83, 309
- supervision, 6
- synthesis of data (see data synthesis)
- system components evaluation, 200-201
 - creating scoring rubrics with examples, 202
 - defining evaluation criteria, 202
 - tying evaluation metrics to business metrics, 203
- system prompts, 215-217
- system-level defense, 250
- systems evaluation, 159-209
 - evaluation criteria, 160-179
 - cost and latency, 177-179
 - domain-specific capability, 161-163
 - evaluation-driven development, 160-161
 - generation capability, 163-172
 - instruction-following capability, 172-177
 - evaluation pipeline design, 200-208
 - step 1: creating an evaluation guideline, 202-203
 - step 2: evaluating all components in a system, 200-201
 - step 3: defining evaluation methods and data, 204-208
 - evaluation-driven development, 160-161
 - model selection, 179-200
 - data contamination with public benchmarks, 197-200
 - model build versus buy, 181-191
 - model selection workflow, 179-181
 - navigating public benchmarks, 191-197
- OpenAI model quality, 196

T

- task-based evaluation, 201
- temperature, 90-93
- term frequency (TF), 259
- text-to-SQL, 99, 126, 274
- throughput, 414-415
- time between tokens (TBT), 413
- time per output token (TPOT), 33, 412-414
- time to first token (TTFT), 33, 412-414
- tokenization, 55, 69, 121, 260, 268

- defined, 3
- tokenizer, 268
- tokens, 3, 68
- tool use, 296
- top-k, 94
- top-p, 94
- TPOT (time per output token), 33, 412-414
- traces, 470
- trainable parameters, 320-322
- training, 41-42
- training data, 50-57
 - domain-specific models, 56-57
 - multilingual models, 51-55
- training quantization, 331-332
- transfer learning, 308
- transformer architecture, 58-64
 - attention mechanism, 60-62
 - attention modules, 62
 - MLP modules, 62
- transformer blocks, 62-64
 - attention modules, 62
 - embedding modules, 63
 - MLP modules, 62
 - output layers, 63
- TruthfulQA, 192
- TTFT (time to first token), 33, 412-414
- turn-based evaluation, 201

U

- unstructured data, 27, 303
- use case evaluation, 29-32
- usefulness threshold, 33
- user feedback, 474-492
 - extracting conversational feedback, 475-480
 - natural language feedback, 476-479
 - other conversational feedback, 479-480
 - feedback design, 480-489
 - when to collect feedback, 481
 - feedback limitations, 490-492
 - biases, 490
 - degenerate feedback loops, 491

V

- value vector (V), 61
- vector database, 261-263
- vectorization, 438
- vocabulary, 123
 - defined, 3

W

WinoGrande, 192

workflow automation, 28

write actions, 280

Z

zero-shot learning, 213-215