Sumário

Prefácio xxi

1

Entendendo o conceito e a arquitetura do LLM Twin 1

Compreendendo o conceito de LLM Twin 2

O que é um LLM Twin? 2

Por que construir um LLM Twin é importante 3

Por que não usar o ChatGPT (ou outro chatbot semelhante)? 4

Planejamento do MVP do produto LLM Twin 6

O que é um MVP? 6

Definindo o MVP do LLM Twin 7

Construindo sistemas de ML com pipelines de atributos/treinamento/inferência 8

O problema com o desenvolvimento de sistemas de ML 8

O problema com as soluções anteriores 10

A solução – pipelines de ML para sistemas de aprendizado de máquina 12

O pipeline de características 13

O pipeline de treinamento 14

O pipeline de inferência 14

Benefícios da arquitetura FTI 14

Projetando a arquitetura do sistema do LLM Twin 16
Listando os detalhes técnicos da arquitetura do LLM Twin 16
Como projetar a arquitetura do LLM Twin usando o design do pipeline de FTI 17
Pipeline de coleta de dados 18
Pipeline de características 19
Pipeline de treinamento 20
Pipeline de inferência 21
Considerações finais sobre o design do FTI e a arquitetura do LLM Twin 22
Resumo 23
Referências 23

2
Ferramentas e instalação 25

Ecossistema do Python e instalação do projeto 26
Poetry: gerenciamento de dependências e de ambientes virtuais 27
Poe the Poet: ferramenta de execução de tarefas 29
Ferramentas de MLOps e LLMOps 30
Hugging Face: registro de modelos 31
ZenML: orquestrador, artefatos e metadados 32
Orquestrador 33
Artefatos e metadados 37
Como executar e configurar um pipeline do ZenML 41
Comet ML: rastreador de experimentos 43
Opik: monitoramento de prompts 44
Bancos de dados para armazenamento de dados não estruturados e vetoriais 45
MongoDB: banco de dados NoSQL 45
Qdrant: banco de dados vetorial 45
Preparação para a AWS 46
Configurando uma conta da AWS, uma chave de acesso e a CLI 46
Amazon SageMaker: computação para treinamento e inferência 48
Por que optar pelo AWS SageMaker? 49
Resumo 50
Referências 50

3
Engenharia de dados 53

Projetando o pipeline de coleta de dados do LLM Twin 54

Implementando o pipeline de coleta de dados do LLM Twin 58
Pipeline e etapas do ZenML 58
O despachante: como instanciar o rastreador correto? 63
Os rastreadores 65
Classes base 66
Classe GitHubCrawler 69
Classe CustomArticleCrawler 71
Classe MediumCrawler 73
Os documentos do data warehouse NoSQL 76
Os padrões de software ORM e ODM 76
Implementar a classe ODM 78
Categorias de dados e classes de documentos do usuário 83
Coletando dados brutos no data warehouse 85
Solução de problemas 89
Problemas do Selenium 89
Importe nossos dados de backup 90
Resumo 90
Referências 91

4

Pipeline de atributos de RAG 93

Entendendo a RAG 94
Por que usar RAG? 94
Alucinações 95
Informação antiga 95
O framework RAG básico 95
Pipeline de ingestão de dados 97
Pipeline de recuperação 98
Pipeline de geração 99
O que são embeddings? 100
Por que embeddings são tão poderosos 102
Como os embeddings são criados? 104
Aplicações de embeddings 107
Mais sobre bancos de dados vetoriais 107
Como funciona um banco de dados vetorial? 108
Algoritmos para a criação do índice vetorial 109
Operações de banco de dados 109

Uma visão geral de RAG avançada 110
Pré-recuperação 111
Recuperação 114
Pós-recuperação 116
Explorar a arquitetura do pipeline da funcionalidade de RAG do LLM Twin 118
O problema que estamos resolvendo 119
O feature store 119
De onde vêm os dados brutos? 120
Projetando a arquitetura do pipeline do atributo RAG 120
Pipelines em lote 121
Pipelines em lote versus pipelines de streaming 122
Etapas principais 125
Captura de alterações de dados: sincronizando o data warehouse e a feature store 126
Por que os dados são armazenados em dois snapshots? 129
Orquestração 129
Implementando o pipeline do atributo RAG do LLM Twin 130
Configurações 130
Pipeline e etapas do ZenML 131
Consultando o data warehouse 133
Limpeza de dados dos documentos 136
Fragmentar e gerar embeddings dos documentos limpos 137
Carregando os documentos no banco de dados vetorial 139
Entidades de domínio do Pydantic 140
OVM 143
A camada despachante 149
Os handlers 151
Os handlers de limpeza 152
Os handlers de fragmentação 153
Os handlers de embedding 157
Resumo 161
Referências 161

5

Ajuste fino supervisionado 163

Criação de um conjunto de dados de instruções 164
Arcabouço geral 164

Quantidade de dados 166
Curadoria de dados 167
Filtragem baseada em regras 168
Deduplicação de dados 169
Descontaminação de dados 171
Avaliação da qualidade dos dados 171
Exploração de dados 174
Geração de dados 176
Aumento de dados 177
Criação do nosso próprio conjunto de dados de instruções 179
Explorando o SFT e suas técnicas 188
Quando realizar o ajuste fino 188
Formatos de conjuntos de dados de instruções 189
Templates de chat 190
Técnicas de ajuste fino com eficiência de parâmetros 192
Ajuste fino completo 192
LoRA 194
QLoRA 196
Parâmetros de treinamento 197
Taxa de aprendizado e agendador 197
Tamanho do lote 198
Comprimento máximo e empacotamento 199
Número de epochs 199
Otimizadores 200
Decaimento de pesos 200
Checkpointing de gradiente 200
Ajuste fino na prática 201
Resumo 208
Referências 208

6

Ajuste fino com alinhamento às preferências humanas 211

Compreendendo conjuntos de dados de preferências 212
Dados de preferências 212
Quantidade de dados 214
Geração de dados e avaliação 215
Geração de preferências 215
Dicas para geração de dados 216

Avaliação de preferências 217
Criação do nosso próprio conjunto de dados de preferências **219**
Alinhamento às preferências humanas **226**
Aprendizado por reforço a partir de feedback humano 227
Otimização Direta de Preferências (DPO) 228
Implementando DPO **230**
Resumo **237**
Referências **238**

7

Avaliação de LLMs **239**

Avaliação de modelos **240**
Comparando a avaliação de ML e de LLM 240
Avaliações de LLMs de propósito geral 241
Avaliações de LLMs específicas de domínio 243
Avaliações de LLMs específicas à tarefa 245
Avaliação de RAG **248**
Ragas 249
ARES 251
Avaliando o TwinLlama-3.1-8B **252**
Geração de respostas 253
Avaliação de respostas 254
Analisando resultados 259
Resumo **262**
Referências **263**

8

Otimização de inferência **265**

Estratégias de otimização de modelos **266**
Cache KV 267
Processamento contínuo em lotes 269
Decodificação especulativa 270
Mecanismos de atenção otimizados 272
Paralelismo de modelo **273**
Paralelismo de dados 274

Paralelismo de pipeline 275
Paralelismo de tensores 276
Combinando abordagens 277
Quantização de modelos 278
Introdução à quantização 278
Quantização com GGUF e llama.cpp 282
Quantização com GPTQ e EXL2 284
Outras técnicas de quantização 286
Resumo 287
Referências 288

9

Pipeline de inferência de RAG 289

Compreendendo o pipeline de inferência de RAG do LLM Twin 290
Explorando as técnicas avançadas de RAG do LLM Twin 292
Otimizações avançadas de pré-recuperação para RAG: expansão de consulta e autoextração de metadados da consulta 295
Expansão de consulta 295
Autoextração de metadados da consulta 299
Otimização avançada da recuperação em RAG: busca vetorial filtrada 303
Otimização pós-recuperação avançada em RAG: rerranqueamento 304
Implementando o pipeline de inferência RAG do LLM Twin 309
Implementando o módulo de recuperação 309
Reunindo tudo no pipeline de inferência de RAG 316
Resumo 321
Referências 321

10

Implantação do pipeline de inferência 323

Critérios para escolher tipos de implantação 324
Vazão e latência 324
Dados 325
Infraestrutura 325
Entendendo os tipos de implantação de inferência 326

Inferência online em tempo real 328
Inferência assíncrona 328
Transformação em lote offline 329

Arquitetura monolítica versus arquitetura de microsserviços no serviço de modelos 330

Arquitetura monolítica 331
Arquitetura de microsserviços 332
Escolhendo entre arquiteturas monolíticas e de microsserviços 333

Explorando a estratégia de implantação do pipeline de inferência do LLM Twin 334

O treinamento versus o pipeline de inferência 337

Implantação do serviço LLM Twin 338

Implementando o microsserviço de LLM usando AWS SageMaker 339

O que são os DLCs da Hugging Face? 339
Configurando funções do Amazon SageMaker 340
Implantação do modelo LLM Twin no AWS SageMaker 341
Chamando o endpoint de inferência do AWS SageMaker 351

Construindo o microserviço de negócios usando FastAPI 355

Capacidades de escalonamento automático para lidar com picos de uso 358

Registrando um alvo escalável 360
Criação de uma política escalável 361
Limites mínimos e máximos de normalização 362

Período de cooldown 362

Resumo 363
Referências 363

11

MLOps e LLMOps 365

O caminho para o LLMOps: compreendendo suas raízes em DevOps e MLOps 366

DevOps 366

O ciclo de vida do DevOps 367
Os conceitos fundamentais de DevOps 368

MLOps 369

Componentes centrais de MLOps 370
Princípios de MLOps 371

ML vs. engenharia de MLOps 372
LLMOps 373
Feedback humano 374
Salvaguardas 374
Monitoramento de prompts 376
Implantação dos pipelines do LLM Twin na nuvem 378
Compreendendo a infraestrutura 378
Configurar o MongoDB 381
Configurando o Qdrant 382
Configurando a nuvem do ZenML 383
Containerizar o código usando Docker 386
Execute os pipelines na AWS 391
Solução de problemas do erro ResourceLimitExceeded após a execução de um pipeline do ZenML no Amazon SageMaker 393
Adicionando LLMOps ao LLM Twin 394
Fluxo do pipeline de CI/CD do LLM Twin 395
Mais sobre erros de formatação 397
Mais sobre erros de linting 397
Visão geral rápida do GitHub Actions 397
O pipeline de CI 399
Arquivo YAML de CI do GitHub Actions 399
O pipeline de entrega contínua 402
Teste o pipeline de integração contínua e entrega contínua 405
O pipeline de CT 406
Gatilhos iniciais 407
Acionar pipelines a jusante 408
Monitoramento de prompts 411
Alertas 416
Resumo 417
Referências 417

Apêndice

Princípios de MLOps 419

1. Automação ou operacionalização 419
2. Versionamento 421
3. Rastreamento de experimentos 422

4. Testes **422**
Tipos de teste 422
O que testamos? 423
Exemplos de teste 423
5. Monitoramento **426**
Logs 426
Métricas 426
Métricas do sistema 427
Métricas do modelo 427
Derivas 427
Deriva de dados 428
Deriva do alvo 428
Deriva de conceito 429
Como detectar e medir derivas 430
Monitoramento vs. observabilidade **430**
Alertas 431
6. Reprodutibilidade **431**
Índice remissivo **433**