# 目录结构说明

大模型技术及开发-课程项目代码/

│

├─ finetune-bi-encoder/ # Bi-Encoder 微调与测试相关代码

│ ├─ Bi-encoder微调模块说明文档.docx

│ ├─ gen\_hard\_negatives\_by\_api.py

│ ├─ prepare\_data.py

│ ├─ test\_api\_embedding.py

│ ├─ test\_bi\_encoder\_with\_rerank.py

│ ├─ test\_bi\_encoder.py

│ ├─ test\_bm25\_bi\_encoder\_hybrid\_with\_rerank.py

│ ├─ test\_bm25\_bi\_encoder\_hybrid\_with\_rewrite.py

│ ├─ test\_bm25\_bi\_encoder\_hybrid.py

│ ├─ test\_bm25.py

│ └─ train\_bi\_encoder.py

│

├─ finetune-llm/ # 大语言模型（LLM）微调与推理相关代码

│ ├─app.py

│ └─ LLaMA-Factory/ # LLaMA-Factory 主体代码

│ ├─ data/ # 数据集与数据处理

│ ├─ evaluation/ # 评测脚本与数据

│ ├─ examples/ # 配置文件与训练/推理示例

│ ├─ scripts/ # 工具脚本

│ ├─ src/ # 主程序源码

│ ├─ tests/ # 测试用例

│ └─ 其他配置与文档

│

├─ rag/ # 检索增强生成（RAG）相关代码

│ ├─ static/ # 前端静态资源

│ ├─ templates/ # 网页模板

│ ├─ pdfs/ # 知识库文件

│ ├─ models/ # 模型文件

│ ├─ app.py # 主程序

│ ├─ app\_save.py

│ ├─ bm25.py

│ ├─ config.py

│ ├─ config\_web.py

│ ├─ conflict\_detection.py

│ ├─ creat.py

│ ├─ download.py

│ ├─ file\_processor.py

│ ├─ history.db

│ ├─ llm\_api.py

│ ├─ models.py

│ ├─ qa\_history.db

│ ├─ retrieval.py

│ ├─ test\_cli.py

│ ├─ test\_cli\_with\_llm.py

│ ├─ test\_llm\_api.py

│ ├─ TextCleaner.py

│ ├─ vector\_store.py

│ ├─ Query\_Rewriting.py

│ ├─ Split\_Text.py

│ ├─ web\_scraper.py

│ └─ web\_search.py

│

├─ web clawer/ # 网络爬虫与数据清洗相关代码

│ ├─ output/ # 处理过程文件存储

│ ├─ Web Clawer说明文件.docx

│ ├─ complete.py

│ ├─ config.py

│ ├─ Deepclean.py

│ ├─ JsontoTXT.py

│ ├─ solver.py

│ └─ TXTtoPDF.py

│

└─ ...

**各模块功能说明**

**1.finetune-bi-encoder**

* **功能**：实现Bi-Encoder模型的微调、测试与评估，支持负样本生成、BM25混合检索等。
* **主要文件说明**：
* train\_bi\_encoder.py：Bi-Encoder模型训练主脚本。
* test\_bi\_encoder.py：模型测试脚本。
* gen\_hard\_negatives\_by\_api.py：通过API生成困难负样本。
* test\_bm25.py、test\_bm25\_bi\_encoder\_hybrid.py等：BM25与Bi-Encoder混合检索相关测试。
* 其他脚本：支持数据准备、API测试、重排序等功能。

**2.finetune-llm/LLaMA-Factory**

* **功能**：运用了[hiyouga/LLaMA-Factory: Unified Efficient Fine-Tuning of 100+ LLMs & VLMs (ACL 2024)](https://github.com/hiyouga/LLaMA-Factory)中的微调框架，支持LLaMA等主流大模型的微调、推理、评测，包含丰富的数据处理、训练、评测、推理脚本。
* **主要目录与文件说明**：
* app.py:用于生成模型的fastapi部署
* saves/:存储模型评估时的日志文件
* checkpoint-4782/:存储训练时的配置以及日志文件
* data/：多种数据集及其处理脚本。
* evaluation/：评测脚本，支持CEVAL、MMLU等主流评测。
* examples/：训练、推理、合并等多种配置文件和示例。
* scripts/：模型转换、推理、统计等工具脚本。
* src/：核心源码，包括API、模型、训练、WebUI等。
* tests/：单元测试与集成测试。

**3.rag**

* **功能**：实现RAG（检索增强生成）流程，包括文档检索、向量存储、LLM问答、Web UI等。
* **主要文件说明**：
* app.py：RAG主程序，集成检索与生成。
* vector\_store.py：向量存储与检索。
* llm\_api.py：大模型API调用。
* file\_processor.py、web\_scraper.py：文件与网页数据处理。
* bm25.py、retrieval.py：检索算法实现。
* static/、templates/：Web前端资源与页面模板。
* Query\_Rewriting.py、Split\_Text.py：数据预处理与重写相关脚本。
* 其他脚本：包括配置、测试、数据清洗等辅助功能。

**4.web clawer**

* **功能**：实现网页爬取、数据清洗、格式转换等功能。
* **主要文件说明**：
* complete.py：完整爬虫运行流程
* Deepclean.py：深度清洗脚本，处理原始数据。
* JsontoTXT.py、TXTtoPDF.py：数据格式转换工具。
* solver.py：爬取网页与简单清晰、分块。
* config.py：爬虫相关配置。