

大作业：（多模态）大语言模型应用开发

目标：开发一个解决实际需求的大语言模型应用，需体现对 LLM 或者 VLM 核心能力的理解与扩展

一、项目要求

1. 基础功能层（必须完成）

- 集成至少 1 个大语言模型 API（如 GPT-3.5/4、qwen、deepseek 等），展示不同参数（如 temperature=0.7 vs 1.2）对输出的影响对比
- 实现核心功能闭环，需定义明确的输入输出规范（如 JSON Schema）并实现校验机制
- 包含基础交互界面（命令行/CUI/GUI 任选）
- 处理一些真实场景测试数据作为展示
- 包含下方“技术实现规范”中的要点。

2. 进阶功能层（选择性完成）

- 实现多模态扩展（图像/语音输入输出）
- 实现高质量结构化输出控制，能 95% 的概率生成符合要求的格式
- 构建领域知识库增强，如 RAG 架构需说明 embedding 模型选型（如 text2vec）和检索策略（如 FAISS 索引）
- 开发记忆机制（对话历史/用户画像）
- 集成外部工具（包括但不限于计算器、数据库、API、Python 代码解释器）
- 实现一个自主“Agent”，需实现至少 3 种工具调用决策逻辑
- 其他可能的额外功能

3. 创新维度（选择性完成）

- 解决未被主流产品覆盖的需求痛点

- 提出新颖的 **prompt engineering** 方案
- 设计独特的输出呈现形式
- 其他可能的创新形式

注：以上括号内的，均是举例，而非必须和仅仅。比如集成的外部工具可以从括号中选几个，也可以自己再额外设计。

二、技术实现规范

1. 模型调用

- 必须展示 API 调用参数调优过程（**temperature/top_p** 等）
- 需处理流式响应，实现逐字/分块输出效果（如 ChatGPT 式打字机效果），禁用单次完整响应
- 实现输入预处理（敏感词过滤/指令注入防护）

2. 系统架构

- 需提供架构设计图（数据流图/模块关系图）
- 要求模块化开发（至少 3 个独立功能模块）
- 必须包含异常处理机制（API 失败重试、结构化输出失败等）

三、评分标准矩阵

评估维度	权重	评估要点
基础功能	35%	核心功能实现度、异常处理覆盖率、单元测试通过率
进阶功能	20%	进阶功能实现度
创新价值	15%	需求新颖性（需附竞品分析）、解决方案独创性、技术组合创新性
工程规范	10%	git 管理、API 密钥安全管理、依赖管理（ requirements.txt ）
展示质量	20%	演示视频流畅度（3-5 分钟）、技术文档完整性（API 文档+用户手册等）

扣分项:

- 大规模代码抄袭: 零分处理
-

四、交付清单

1. 源代码 (以 `git` 仓库的形式提交) 或其他等价内容。
 2. 技术报告 (正文最多 8 页, 附录不限制页数, 会提供模板, 需包含: 需求分析→技术选型→实现细节→评估对比→反思)
 3. 演示视频 (MP4 格式, 含功能演示与技术解说)
 4. 可执行文件或部署方案 `README` 或部署链接 (需保证期末周可访问)
 5. 多人组队时, 需要提供小组分工表 (组队人数不超过 3 人)
-

五、补充说明

- 推荐使用 [CAMEL-AI](#), 或者 `Dify` 框架搭建应用。

注: `CAMEL-AI` 团队提供微信群为大家提供相关技术支持 (可添加微信: `qfscwl` 加入群聊)
 - 鼓励使用 `LLM` 生成代码; 如遇查重, 可提供原始代码生成网址、截图等信息免于查重。
 - 如工程规范不适用, 则对应的 10% 分配至功能完整度。
 - 技术报告虽然可添加不限制页数的附录, 但助教无法保证阅读所有附录, 并且给分主要依据为正文。
 - 如提出的项目在原理上无法完成/没有必要实现某一基础功能/要求, 可以在正文中加以说明并略过该功能/要求。
-

六、附录:

可免费使用以下硅基流动 `api key` 进行 `DeepSeek` API 调用:

`sk-xfcnogwtvodvdbfblkleaodorbactpyyqvjfiapanuiyhxy`

代码示例见 `test_api.py`, 其中包含两个调用样例