NLP Course Project Framework

1. Introduction

构建实用性 LLM 应用的研究具有前沿性和实践指导意义，尤其在资源受限环境下如何平衡模型精度与推理延时成为热点问题。本项目提出的研究计划立足于 LangChain 框架，融合两大关键技术

模型量化：探讨如何利用量化技术（如 PTQ、QAT、低秩自适应（LoRA）等）降低大语言模型在边缘设备上的资源占用，确保在内存和延时压力下依然能确保较高的效能。

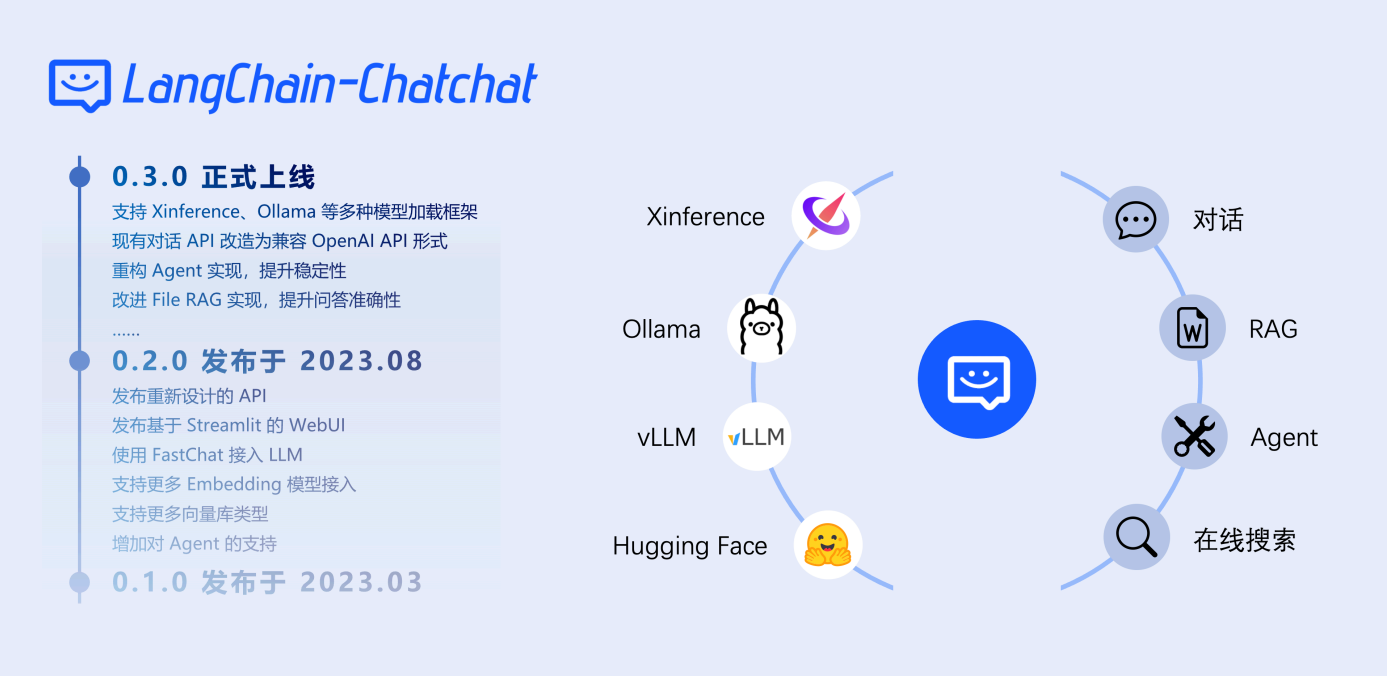
人类反馈机制：结合 RLHF 或 Retrieval-Augmented Generation（RAG）的思想，为系统引入人类反馈，进一步优化响应质量，尤其在金融、算法设计等高敏感领域可发挥巨大作用。

2. 系统场景与架构设计

2.1 文档问答系统 (Document QA System)

参考开源链接：

- Langchain Chatchat 基于 ChatGLM 等大语言模型与 Langchain 等应用框架实现，开源、可离线部署的 RAG 与 Agent 应用项目



系统典型工作流程如下：

1. 用户提出问题；
2. 系统通过访问控制验证并从向量数据库中检索相关文档内容；

3. 检索内容与问题一起输入量化后的 LLM 模型；
4. 生成答案后同步触发人类反馈回路，用于后续微调和效果优化（难度高，选做）
5. 优化策略建议使用RAG(参考project 4 guideline)

2.2 会话代理开发 (Conversational Agent)

会话代理（Chatbot）的重点在于对话上下文的维护与短期记忆管理，通过LangChain内置的内存模块可实现对话历史存储与上下文更新。

2.3 自主代理 (Autonomous Agent)

自主代理（Autonomous Agent）侧重于任务规划与API集成，能够根据用户需求执行简单任务，如查询数据库、调用外部服务等。
这个任务可以做的很简单（天气查询,简单prompt和爬虫就行，最简单的，直接用Data Engineering的作业1或者project就行，能爬电影数据和做推荐任务），也可以做的非常难(Trade Agent)
~~注：这里的trade agent适合采用RLHF，但是时间太紧张了，估计做不了，以后感兴趣可以等这篇论文复现了再做一下。~~

3. 模型对比与量化部署

- Awesome Chinese LLM
- Awesome LLM
- API(待补充)

常见量化策略

- FP32（全精度）：可以作为baseline
- PTQ (Post-Training Quantization)
- QAT (Quantization-Aware Training)
- LORA + FP16&FP8&FP4
- Others Finetune Method

4. 构建评估体系

可以引入统计显著性检验，增强结论可信度

- 平均响应延时
- 模型准确率 （测试不同上下文长
  度下的问答准确率及响应延时）
- 内存使用
- 显存使用

- 上下文理解能力：测量模型在多轮对话中保持上下文一致性的能力
- 回答相关性：使用ROUGE、BLEU或自定义相关性得分衡量回答与问题的关联程度
- 真实性评分：评估模型生成内容的事实准确性

5. 消融实验

5.1 量化策略消融（必做）：

逐一测试不同量化方法(PTQ、QAT、LoRA等)
在不同精度下(FP16、INT8、INT4等)评估性能-资源权衡
混合量化方法的对比(如某些层使用INT8，某些层保持FP16)

5.2 RAG组件消融：

检索机制对比(语义检索 vs. 关键词检索)
不同向量数据库的性能比较
Chunk大小影响(较大vs较小文档块对理解和响应质量的影响)

5.3 上下文长度消融（简单，必做）：

测试不同上下文窗口大小对性能的影响
分析上下文长度与延迟、内存使用的关系曲线

6. 预期结论与未来工作方向

预期结论
混合量化方式A在延时和资源占用上展现出较好的平衡，实验中较 FP32 模型显著降低延时与内存占用。
一些消融实验的结论，某些模型适用于特定任务的结论

未来工作方向
研究金融领域专用的反馈权重自适应算法，~~提升 RLHF 系统的鲁棒性能；~~
进一步集成多模态数据（如图像，类似于vision课程的image instance search一样）以扩展文档问答系统的应用场景；
考虑数据安全与隐私问题，在边缘部署过程中应用联邦学习框架，实现跨终端反馈聚合和模型更新。