RustSBI Agent 项目计划

工作方法：用 RAG，必须用知识库，否则无法学习专业知识

项目规模：考虑 pdf，暂不考虑网页（html 解析、爬虫），html RAG 可能出问题。pdf 形式也有难度
- pdf 不可编辑，需要提取内容
- 数据存储的格式
思路有两方面。一、自己写 pdf 解析还是用现成工具。二、数据提取之后，如何保存？以 chunk 吗？涉及表格的时候，是否不能将表格截断（可能影响大模型的理解，出现幻觉）？
表格等特殊类型的数据有很多。
方案 2：使用现成的轮子，需要广泛测试。需要 AnythingLLM 或 Ollama 或 LLMStudio。测试这些软件拥有的预置模型，如果预置没有，需要用扩展库。如果这些模型的效果不好，我们需要去 HuggingFace 上找合适的 RAG 相关模型，并测试。哪个效果好用哪个。
第二步 1.做和大模型结合（知识库如何接入大模型），2.用哪个开源大模型？Qwen、llama3、字节的，开源的都可以试试。如果两步都实现了，L1 方案已经实现了 LLM Agent。
还存在的问题：1.文档会不断更新，会加入新知识，不可能再次重头构建知识库。L2 阶段要实现知识库的动态更新。
聊天机器人变成广义的 Agent。往 Kimi 的方向发展。Kimi 的优势：如果它的知识库找不到用户的输入信息，它会调搜索引擎（如百度）的 API。如果知识检索的相似性匹配没有超过阈值，知道目前的知识不能解决，就会去搜索引擎上找。此时是完全体方案。

疑问点：1. 不能自己造大模型，我们的能力有限。2. 微调是双刃剑；如果微调效果好，它可能比不微调的流畅度、专业程度高，但如果微调效果不好，甚至不如不微调的大模型。微调是可选的，优先级不高，有额外精力时能考虑。
提示词（任潇）。1.RAG 2.微调 3.prompt。接入知识库之后 prompt 可以继续优化，不冲突。
知识库存储：

指令集手册难度比传统的 RAG 输入难度要大。如果有 markdown 源格式，千万别用 pdf。
分配一个开发强的人做！纯工作量。
目标：优先以保证原始信息为目标。原始信息不要丢掉，追求 100%解析。

同步做的事情：查 anythingllm 或 ollama 是否能解析冷门格式的文档（如 adoc）。解析完可以拿到可视化的分词结果，可判断解析的时候是否出现问题。如果效果不好，就得靠自己了。

如何保存：专业性强（如需要调整模型参数），必须有一定 AI 经验。

测试：专业性不是特别强。一两个小时的学习后可以上手，有现成轮子。
不同大模型的结合：需要一定的专业性。每个大模型的侧重不同，如 llama3 的风格比较简洁，chatglm 容易说无关的废话。属于模型本身的特性，需要 AI 经验才能选择哪个效果好。

以上是 L1 方案的点，L1 结束后，看 L1 的效果，接入哪个搜索 API、实时更新的功能是 L2 考虑的问题。

人手：

1. 解析约 2 个人（纯开发）
2. 大模型测试 2 个人起步（看 LLM document，调 API，不太需要大模型知识，需要工程化能力和调试能力）
3. 调参 1~2 个人（需要对 LLM 本身感兴趣）
4. AnythingLLM、Ollama 最多 2 个人（使用工具，去官网上读文档，用、测，偏测试）

整个团队约 7 个人。