github 仓库对话机器人技术调研

1 现状分析

github 做为一款开源的软件托管平台，上面存放了大量优秀的开源项目供用户学习参考。但是一个完整的开源项目往往包含许多复杂的模块，阅读及理解难度较大，这给许多用户造成了学习上的困难。而以 Chatgpt 为代表的大语言模型，表现出了不俗的代码理解能力，使得让机器人帮助用户理解学习代码成为了可能。但是大语言模型在回答具体项目的问题时，因为缺少相应的背景知识，常常出错甚至会存在编内容的情况。同时由于 token 数量的限制问题，将一整个代码仓库发送给大模型也是不实际的。

用户除了能在 github 上学习其他优秀开源项目外，还能通过 PR 等形式参与到开源项目中。这需要用户通过项目的 issues 了解到当前项目需要完善的内容，并以此为贡献点。但是大多活跃的开源项目 issues 数量都达到了 40+以上，阅读起来需要消耗大量的时间。且 issues 的内容大多分散，在其中找到自己能贡献的方向并不容易。

2 项目目标

基于上述两点问题，本项目希望能构建一款 github 仓库对话机器人。用户可以通过文字对话的方式向机器人提问关于特定 GitHub 仓库的问题。机器人会将特定的 GitHub 仓库做为本地知识库，并以此为基础与大模型进行交互，以提供有关该仓库的相关信息和回答用户的问题。从而帮助用户快速的学习相关内容，并了解到当前项目的最新进展。

3 行业实践

3.1 专业领域机器人实现流程

正如现状分析中提到的那样，大语言模型在回答特定问题时表现的十分糟糕。为此业内的解决方案主要为 fine-tune model 和 context injection 两种。本文主要对 context injection 技术进行了研究，并从中选出两款具有代表性的开源项目。下文将从文档加载，数据嵌入及向量存储，相似度匹配，3 个方面对这两款开源项目进行技术分析。

3.1.1 supabase Clippy

介绍: supabase Clippy 是一款文档查询机器人，用于查询有关 supabase 的相关文档内容。github 仓库：https://github.com/supabase/supabase/tree/master

文档加载

supabase Clippy 需要加载的文档为 supabase 的帮助文档。大多为 markdown 格式的文件，内容格式较为单一，supabase Clippy 通过预先编写的脚本对这些文档进行读取。在读取过程中，按照标题(Heading)对文档进行了拆分。

数据嵌入及向量存储

使用 OpenAI 的 embeddings API 为每个拆分出来的部分创建了 embeddings，嵌入模型为 ada-002。并将它们存储在 Postgres 中，存储过程采用 sql 语句编写。

相似度匹配

通过 pgvector 提供的运算符，在 Postgres 中计算两个向量的余弦相似度，并选出其中前 10 个最相似比配向量。

3.1.2 使用 LangChain 的 GPT4 分析 Twitter 算法源代码

文档加载

该项目需要加载 twitter 的开源代码做为背景知识。首先通过 git clone 拉取到代码后，采用 langchian 文档加载器提供的 LanguageParser 对代码进行加载。它将执行以下操作：
将顶级函数和类组合在一起（构成一个单一的文档),将剩余的代码放入单独的文档,保留每个拆分位置的元数据。之后按照 chunk 的大小进行拆分。

数据嵌入及向量存储

通过 langchain 的相关 api 接口，将相关文档存入其中并采用 openai 提供的 embedding 模型(默认)。

相似度匹配

同样通过 langchain 的相关 api 接口即可为完成。

3.1.3 实现流程总结

项目类型	文档切分策略	嵌入模型	向量数据库调用	相似度匹配算法
supabase Clippy	按标题进行切分	text-ada-020	编写 sql 语言	余弦相似度匹配
langchain	按代码逻辑	text-ada-020	利用系统集成好的 api	余弦相似度匹配

表 1: 现存方案对比

首先在文档切分策略上，对于文本内容 supabase Clippy 是按标题进行切分。而对 twitter 的源码分析项目中，是通过顶级函数和类将相应代码组合再一起的形式对代码文件进行切分。对于本项目而言，需要读取代码仓库，里面即包含了文本文件，也包含了代码文件，因此可以尝试将两种策略结合在一起。而嵌入模型方面，原先一直对自然语言与代码的匹配能否成功存在疑问。但是通过对 twitter 项目的研究发现，text-ada-020 模型能够完成对代码的检索。openai 的相关文档(https://platform.openai.com/docs/guides/embeddings/what-are-embeddings)中的描述也证明了这一想法。对于向量数据库的调用，当前 matrix one 向量数据库并未集成到 langchain 中，所以没有办法直接通过调用 langchain 的 api 为此。但是我们可以通过学习 supabase clippy 中 sql 的编写逻辑，设计出 langchain 与 matrix one 之间的 api，这也是项目未来的努力方向。