基于大模型的语料库问答详细方案

1. 所用开发工具和技术：

**使用大模型ChatGLM3-6B**

ChatGLM3是一个基于Transformer的预训练语言模型，由清华大学KEG实验室和智谱AI公司于2023年共同训练发布。

基本原理：将大量无标签文本数据进行预训练，然后将其用于各种下游任务，例如文本分类、命名实体识别、情感分析等。

ChatGLM3-6B是ChatGLM3系列中的开源模型，保留前两代模型对话流畅、部署门槛低等众多优秀特性。

在开发过程中，我们也尝试过用llama，mixtral，gemma大模型，通过langchain与llamaindex集成的方式构建RAG，通过streamlit搭建。

我们对llama-alpaca模型进行微调，尝试了huggingface格式与gguf格式部署，对模型进行量化，但是由于模型回复时间，模型回复质量等各种问题还是选用了chatglm3-6B。

相对而言chatglm3-6B对硬件设施的要求更低，在不量化的情况下也可以使用，保存了精度。这也是我们选用这个模型的原因。

**python编程语言**

选用python是因其易用性和简洁性，并且有丰富的库支持以及强大的平台和集成性，非常适合快速实验和迭代。

**Streamlit**

**前端设计**：

使用Streamlit框架集成HTML和CSS，我们实现了一个直观且响应式的网页前端。这允许自定义页面布局和样式，提供了一个清晰、美观的用户交互界面。

Streamlit的互动性质使得无需编写复杂的前端代码就能快速部署原型，这对于迭代开发和快速测试新功能特别有利。

**后端集成**：

与大型语言模型的问答系统对接，利用其强大的自然语言处理能力来驱动后端功能。用户的查询通过前端输入，后端模型处理后返回相应的答案或信息。

通过这种方式，我们能够提供丰富的交互式查询服务，从简单的数据检索到复杂的问题回答，大大增强了应用的功能性。

**页面流和数据库连接**：

利用Streamlit的会话状态管理和控件，我们实现了页面间的跳转，如从主页跳转到备忘录页面，再到问答页面等，为用户提供流畅的导航体验。

接入SQLite数据库，构建了备忘录功能。用户可以添加、查询、修改和删除备忘录条目，系统会自动保存在数据库中，实现数据持久化。

二．选定目标城市并收集文旅数据

1.使用python爬虫，从苏州园林、携程网、去哪儿旅行、马蜂窝、百度、知乎、贴吧等网站上获取有关苏州著名景点的看点简介、具体位置、开放时间、参观要求，及其附近的美食、酒店、交通等具体信息，并获得相应的大众评价作为参考。

2.同时，利用huggingface平台，搜索其他用户上传的相关文旅数据集，以丰富数据集内容。

三．整理与清洗数据，构建语料库

1.数据清洗：包括数据Loader，提取PDF、word、markdown以及数据库和API等；

2.数据处理：包括数据格式处理，不可识别内容的剔除，压缩和格式化等；

3.分块（Chunking）:分块的主要目的是减少嵌入（embedding）内容的噪声，并保持语义相关性1。在语义搜索等应用中，通过有效的分块策略，可以确保搜索结果准确捕获用户查询的本质4。此外，分块还有助于在构建基于知识库的会话代理时，构建上下文，使代理能够基于受信任的信息进行回答。

4.向量化（embedding）:这是将文本、图像、音频和视频等转化为向量矩阵的过程，也就是变成计算机可以理解的格式，embedding模型的好坏会直接影响到后面检索的质量，特别是相关度。text2vec-base-chinese模型在中文文本表示、语义匹配任务、易用性、可访问性和社区支持方面展现出显著的优势，因此我们选择这个模型。

四．大模型处理

1.Models模型

在langchain中加载ChatGLM3模型，需要进行如下配置：

继承langchain.llms.base.LLM类，新建自定义的类

通过@property装饰器将\_llm\_type方法转为【只读属性】，即可以被类访问，而不需要实例的方法来访问（比如model.\_llm\_type==ChatGLM3的结果为true）

重写\_call方法：加载自己的模型，并限制只输出结果（ChatGLM3原输出部署直接str，langchain中要求模型返回必须是str的结果）

2.Indexs索引

由于大语言模型（LLMs）存在数据实时性的问题，它们只知道通过它们的训练数据所呈现的世界。因此我们可以将最新的一些知识文档添加到LLMs中，来补充LLMs模型的知识。

在langChain中进行外部数据检索，一般按照如下流程实现：

基于langchain.document\_loaders下的加载类进行文件加载

基于langchain.text\_splitter对加载后的文件进行分割，以便传入大语言模型

将分割后的文档转成嵌入向量，然后存储到向量数据库

基于向量的余弦相似度来进行检索

3.Prompts提示词

提示模板是生成语言模型提示的预定义配方。模板可能包括指令、少量示例以及适用于特定任务的特定上下文和问题。

4.Chains链

LLMChain（格式化链）：负责对用户输入进行格式化，并将格式化的输入传递给下一个链。

RetrievalQAChain（检索型问答链）：通过嵌入向量索引进行查询，可以用来构建文档系统，并回答用户提出的问题。

5.RAG技术结合了检索与生成的能力，使得模型在生成回答时能够参考从FAISS数据库中检索到的相关信息。利用这种技术来提高了模型的回答准确性，使模型的回答更具深度和上下文关联性。

**外部知：**RAG模型可以有效地利用外部知识库，可以引用大量的信息，以提供更深入、准确且有**识的利用**价值的答案，提高生成文本的可靠性。

**数据更新及时性：**利用RAG检索库的更新机制，实现知识的即时更新，无需重新训练模型。

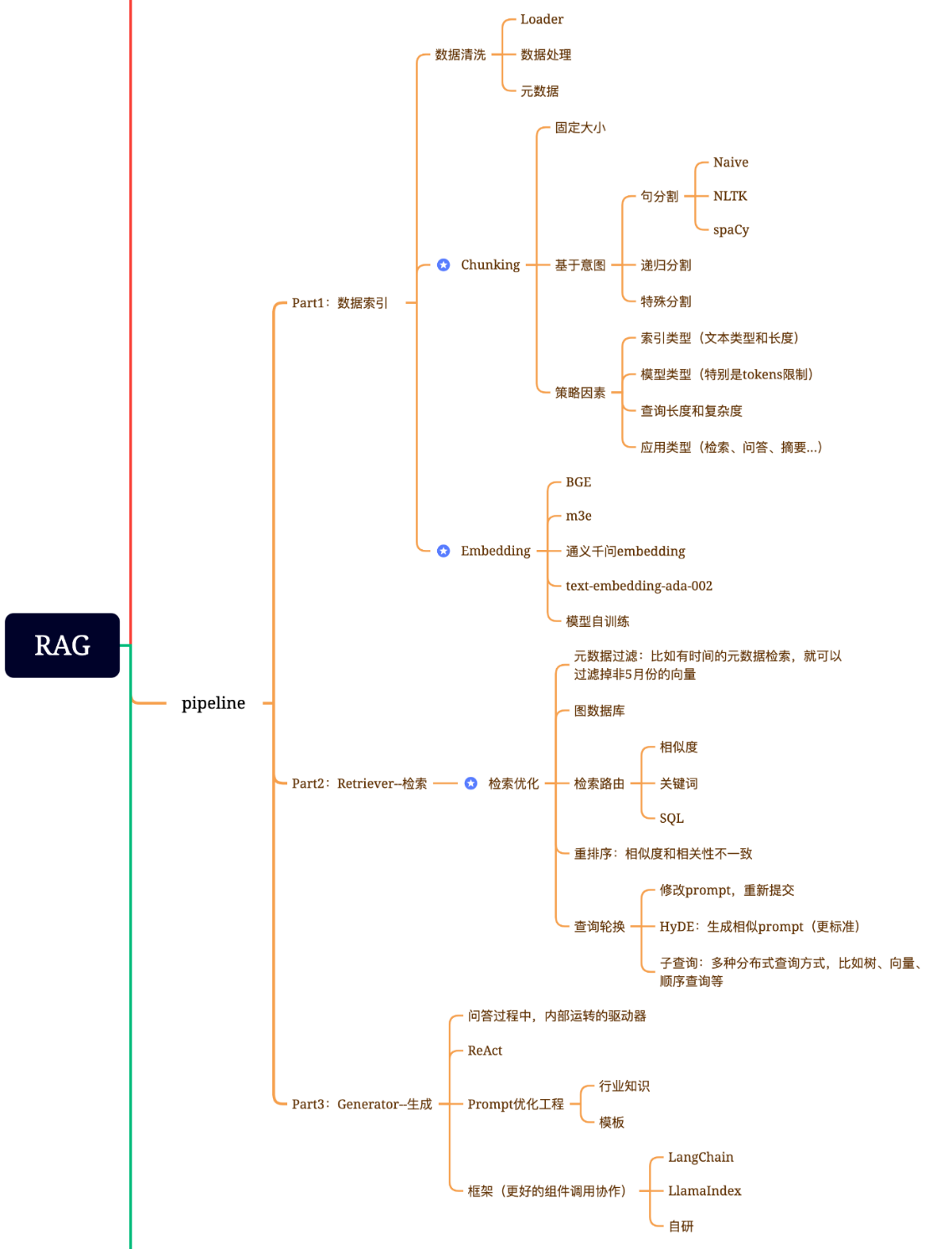
**回复具有解释性：**RAG模型的答案直接来自检索库，回复具有很强的可解释性，减少大模型的幻觉。可以核实答案的准确性，从信息来源中获取支持。

**高度定制能力：**RAG模型可以根据特定领域的知识库和prompt进行定制，使其快速具备该领域的能力。

**安全和隐私管理：**RAG模型可以通过限制知识库的权限来实现安全控制，确保敏感信息不被泄露，提高了数据安全性。

**减少训练成本：**RAG模型在数据上具有很强的可拓展性，可以将大量数据直接更新到知识库，以实现模型的知识更新。

**以下是RAG实现流程：**



五．设计并实现问答系统

1.综合比对了faiss和chroma在向量检索的区别，在检索精度上差异不大，如果需要处理大规模的向量数据，并且想利用GPU加速，那么FAISS可能更适合。如果需要快速搭建一个带有嵌入向量的AI应用程序，并且不想自己管理索引，那么Chroma可能更适合。为了大模型问答的即时性，选用faiss向量数据库。

使用Faiss的高效搜索和数据库构建功能实现对语料库的高效管理、快速检索和精确匹配

**降维和嵌入**：FAISS通常与降维技术（如t-SNE、PCA、UMAP等）和嵌入方法（如Word2Vec、BERT等）结合使用，将高维数据投影到低维空间，以便于进行快速的相似性搜索。

**索引和检索**：FAISS允许用户创建数据的索引，加速搜索过程。索引可以根据数据的特性和搜索需求进行优化，以提高检索的准确性和速度。

2.使用Faiss实现向量检索（RAG），利用similarity\_search\_with\_score来体现文档相似度，利于后续对文档的处理。

但是在检索后，把检索所得文档与模型回答整合时遇到困难，以下时思考过程以及最终方案：

1.直接从页面内容中选择与查询文本相似度最高的部分，并将这部分内容加入到输出中。如果相似度过低，就删去。

缺点：这种方法可能导致输出的连贯性不强，并且在文档内容截断的情况下，回答可能不流畅。

优点：即使遇到与检索文档不相关的问题，也能够提供相关的默认输出。

2.把根据query检索到的内容作为prompt中一个模块，要求模型对其进行修改，并根据query给出回复，在提示词中加入了如果文档没有相应内容就使用默认回答。

缺点：因为我在提示词中加入了如果文档没有相应内容就使用默认回答，导致问一些常识性问题也会回答不了，所以提示词需要修改。

优点：可以对文本二次处理，连贯性强。

3.将检索到的文档内容进行筛选，并将筛选后的文档同query作为prompt的一部分传给模型，，然后要求模型基于文档针对query进行回复。

缺点：默认的prompt是专门针对旅游方面修改的，如果遇到与旅游不相关的问题，prompt没有良好的支持度。可能会因为信息量过多而影响提问的质量。导致大模型忽略query中的细节。

优点：允许对检索到的文本进行二次处理，从而提高回答的连贯性。相对大模型参数微调，没有遗忘先前知识库的风险，可以最大程度利用好现有知识库。

我们采用的是第三种方式，能够有效地处理和整合检索到的文档与用户问题，从而得到质量较高的回复。

3.模型回复生成

Langchain的优势在于其全面性和灵活性。它不仅提供与LLMs交互的标准化方法，而且还通过chains机制结合了其他服务和数据源，提供了一种更为复杂和强大的数据处理和任务执行策略。这使得Langchain不只是一个数据增强工具，而是一个完整的开发框架，适用于构建复杂的AI驱动应用。

Langchain的另一个优势是其内置的memory功能，这对于构建需要维护状态的对话系统和连续任务处理系统特别有用。此外，Langchain的evaluation组件也为模型性能提供了持续的监测和提升的机会。

3.使用Streamlit构建一个Web界面，用于展示和交互问答系统。

**快速原型开发**：Streamlit提供了快速创建和迭代数据应用的能力，能够迅速搭建出大模型的演示或原型系统。

**交互式用户界面**：Streamlit能够轻松构建出具有交互性的用户界面，可以通过这些界面与大模型进行交流，例如输入问题、调整生成参数、查看模型输出等。

**参数调整和优化**：通过Streamlit的滑动条、输入框等控件，可以实时调整大模型的参数。

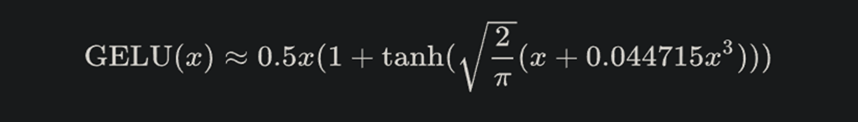
1. 测试与优化

对比实际答案和系统生成的答案，找出问题和不足，并进行相应的优化。

七．理论基础

1．激活函数

使用的激活函数为GELU，其可以近似实现为：



2．GLU层

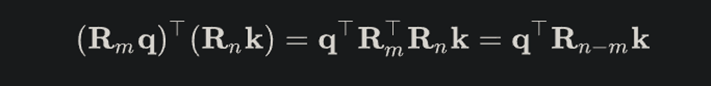
虽然在实现代码中命名为GLU，但这里实现的还是MLP层



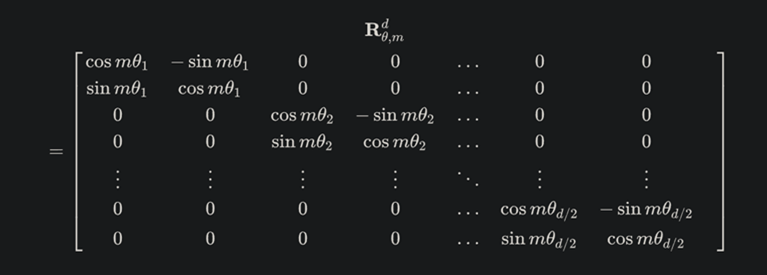
3.位置编码：RoPE

（1）原理

RoPE的目标是构建一个位置相关的投影矩阵，使得

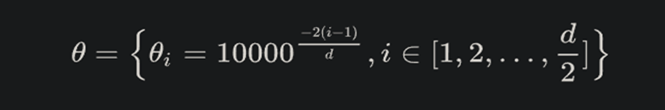


其中，q和k分别对应注意力机制中的query和key向量，m和n代表两个位置，Ri表示位置i处的投影矩阵。下面是作者建议R的形式：

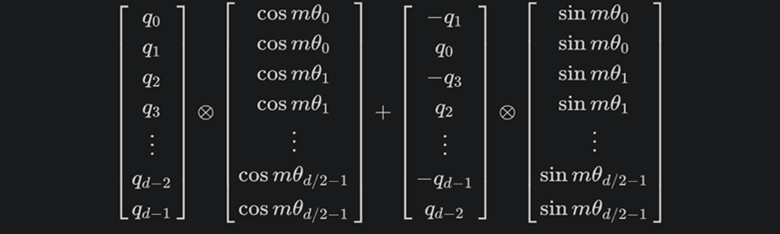


其中，d是query和key的维度，θ是一个超参数。

通常，θ会设置为

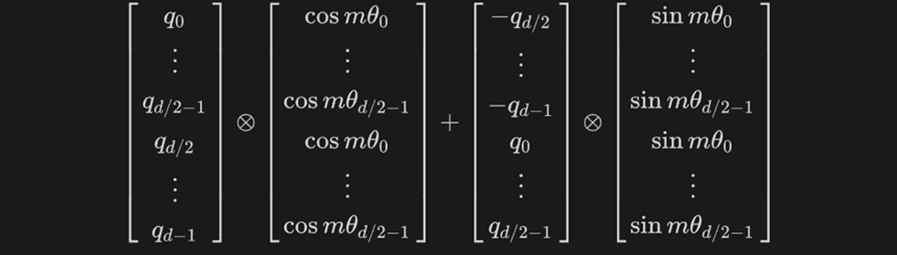


由于矩阵R非常稀疏，为了提供运算速度，作者也给出了实现方式，以query向量q为例：



（2）实现

采用了PaLM的实现方式，不同于上面的公式：



方便验证，该位置编码仍然满足对称性

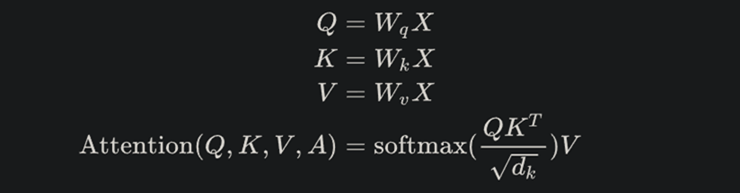


4、注意力层

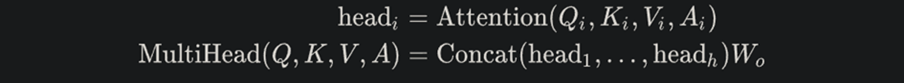
（1）原理

自注意力机制。标准的自注意力机制为：





其中，X是输入，Wq，Wk，Wv，分别是query、key、value的投影矩阵。相比于标准的注意力机制，ChatGLM-6B在Q和K中注意力了RoPE位置信息。多头注意力就是将多个单头注意力的结果拼接起来。



（2）Faiss相似性搜索

给定一组维数为d的xi向量，Faiss用它在RAM中构建一个数据结构。在构造结构之后，当给定一个新的矢量维数时，它会有效地执行以下操作：

