北京邮电大学人工智能学院

自然语言处理实验中期报告（概要设计文档）

课程名称：自然语言处理

实验名称：大模型检索增强生成

实验完成人：

姓名：孟祥超　　学号：2021522081

姓名：巩羽飞　　学号：2021522077

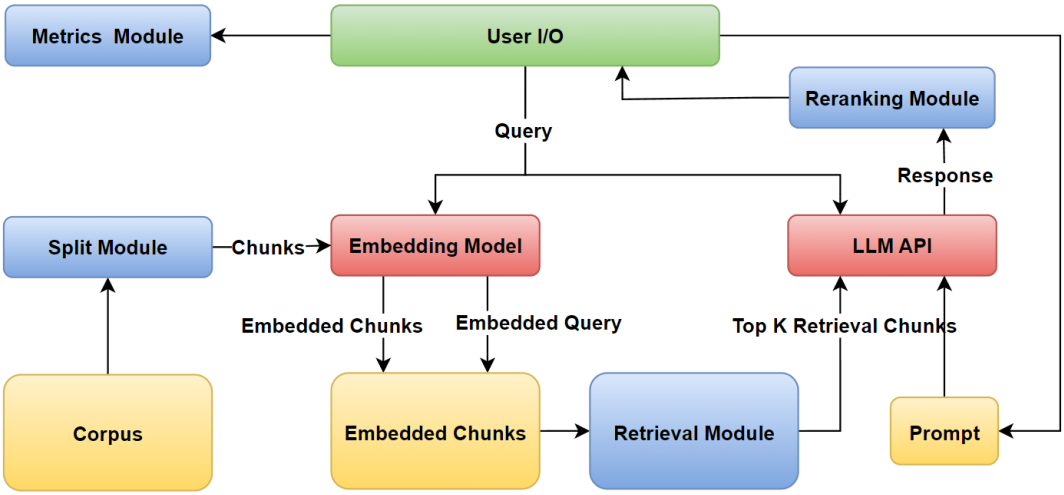
姓名：廉　涟　　学号：2021522082

指导教师：袁彩霞

日　　期：　2024　年　06　月 03 日

一、 系统架构

系统整体考虑采用 pipeline 架构，首先用户可以通过交互界面输入 query，此 query 会先经过 Embedding 得到向量，然后在已有的对 corpus 进行 chunks split 后的 Embeddings 中进行 Retrieval 并融合结果，最后使用优化后的生成模型进行文本生成，得到最后的 anwser，上述过程可简单用以下示意图来表示：



具体到每个模块的工作方式，请参见第四部分——技术方案。

二、 模块划分

我们将模块划分为 Split Module, Retrieval Module, Reranking Module, Metrics Module, LLM API and User I/O Module, Embedding Model Module 这六个模块，这种模块化设计可以提高我们 RAG 系统的灵活性、可扩展性和可维护性。

- **Split Module:**

  将输入文本拆分为合适的段落或句子，以便后续的信息检索和文本生成。

- **Retrieval Module:**

  基于拆分后的输入，从知识库中检索相关的信息片段。

- **Reranking Module:**

  对检索到的候选信息片段进行重新排序，确定最相关的内容。

- **Metrics Module:**

评估检索和生成输出的质量，为优化提供反馈。

- **LLM API and User I/O Module:**

  处理用户输入，调用 LLM 生成输出，并呈现给用户。

- **Embedding Model Module:**

  为检索和生成任务提供文本表示能力，如词嵌入、句嵌入等。

三、 接口设计

本模块负责处理用户输入，调用大语言模型（LLM）生成相应的输出，并将结果呈现给用户。

具体功能包括：

- 用户输入处理：接收用户输入的问题，并将其转换为适合模型处理的格式。这包括对输入文本进行分词和编码处理，以确保模型可以正确理解和处理输入内容。
- 调用 LLM 生成输出：利用零一万物开发的 Yi-34B 大语言模型生成回答。该模型经过训练，支持 4K 序列长度，在推理期间可扩展到 32K，能够提供高质量的双语回答。在多项评测中，Yi-34B 取得了全球领先的表现，达到了多项国际最佳性能指标。
- 输出结果呈现：将模型生成的回答返回给用户。为了确保用户体验，模块会对输出结果进行适当的格式化和处理，使其易于阅读和理解。

API 参数说明：

prompt_tokens (int)：表示用户输入的问题中包含的 tokens 数量。

completion_tokens (int)：表示模型生成的回答中包含的 tokens 数量。

total_tokens (int)：表示整个交互过程中涉及的 tokens 总数量，包括输入和输出的 tokens

四、 技术方案

4.1 总体方案

本技术方案旨在设计一个模块化的 RAG（Retrieval-Augmented Generation）系统，以提高系统的灵活性、可扩展性和可维护性。系统将通过六个主要模块协同工作，实现文本的检索和生成任务。具体模块包括 Split Module, Retrieval Module, Reranking Module, Metrics Module, LLM API and User I/O Module，和 Embedding Model Module。整个系统采用 pipeline 架构，流程如下：

1. **用户输入**：用户通过交互界面输入 query。
2. **文本拆分**：Split Module 将输入文本拆分为合适的段落或句子。
3. **文本嵌入**：Embedding Model Module 对拆分后的文本进行嵌入，得到语义向量。
4. **信息检索**：Retrieval Module 基于语义向量从知识库中检索相关信息片段。
5. **重排序**：Reranking Module 对检索到的候选信息片段进行重新排序，确定最相关的内容。
6. **文本生成**：LLM API 调用生成模型，基于排序后的信息片段生成答案。
7. **输出评估**：Metrics Module 评估生成输出的质量，并反馈给系统进行优化。

4.2 各模块方案

1. Split Module

作用：将输入文本拆分为合适的段落或句子，以便后续的信息检索和文本生成。

设计：

- 使用预处理算法将输入 query 分割成多个子句或段落。
- 考虑中文文本的特殊性，使用特定的中文分词工具，如 jieba 分词。

优势：

- 任务拆分为更小的子任务，提高系统灵活性和可扩展性。
- 合理的拆分有助于提高检索和生成的精度。

2. Retrieval Module

作用：基于拆分后的输入，从知识库中检索相关的信息片段。

设计：

- 采用向量化检索方法。
- 使用 Embedding Model Module 对文本进行向量化表示，利用 BGE-base 模型。
- 基于余弦相似度进行向量检索，筛选出最相关的信息片段。

优势：

- 引入外部知识增强模型的推理和生成能力。
- 模块化设计便于尝试不同的检索算法。

3. Reranking Module

作用：对检索到的候选信息片段进行重新排序，确定最相关的内容。

设计：

- 使用先进的排序算法，如 BERT-based Reranker，对候选信息片段进行重新排序。
- 结合上下文和 query 的语义相似度，确定最相关的信息片段。

优势：

- 通过复杂的排序算法，更准确地选择最有价值的信息，提高下游生成任务的性能。

4. Metrics Module

作用：评估检索和生成输出的质量，为优化提供反馈。

设计：

- 对 QA 任务，评估指标包括答案准确率、BLEU 分数。
- 对摘要生成任务，评估指标包括 ROUGE-1/2/L 分数。
- 对整体生成结果，评估生成文本的流畅度和贴合度。

优势：

- 引入客观指标指导系统持续改进，提高整体性能。
- 模块化设计便于尝试不同的评估方法。

5. LLM API and User I/O Module

作用：处理用户输入，调用 LLM 生成输出，并呈现给用户。

设计：

- 接收用户输入 query，调用 Embedding Model Module 进行文本嵌入。
- 调用生成模型生成答案，并返回给用户。
- 用户界面友好，支持多轮交互。

优势：

- 用户交互和 LLM 调用逻辑与其他模块解耦，提高系统可维护性和扩展性。

6. Embedding Model Module

作用：为检索和生成任务提供文本表示能力，如词嵌入、句嵌入等。

设计：

- 使用 BGE-base 模型进行文本嵌入。
- 根据任务需要，可切换和尝试不同的嵌入模型。

优势：

- 文本表示功能独立，灵活使用不同的嵌入模型。
- 优化整体性能，提高检索和生成的准确性。

4.3 实验设计

1. 模型选择

- 使用 BGE-base 模型进行实验。
- 对比其他相似模型，如 SimCSE、ERNIE 等，进行对照试验，分析其效果。

2. Retrieval Augment Methods

- 采用向量化检索方法，利用 Embedding 模型对输入进行 Embedding 得到语义向量。
- 基于余弦相似度进行检索，提高检索精度。

3. Generation Optimize Methods

- 采用 Top-k 采样进行文本生成。
- 调整 temperature 参数控制生成文本的多样性。

4. 系统架构设计

- 整体采用 pipeline 架构。
- 用户通过界面输入 query，经过各模块处理后生成最终答案。

5. 评估指标

- QA 任务：答案准确率、BLEU 分数。

- 摘要生成任务：ROUGE-1/2/L 分数。
- 生成文本的流畅度和贴合度。

五、 已完成工作

- 完善了后续的任务分工。
- 指定了最终的系统架构
- 对实验进行了合理的模块划分与任务分配
- 完成了 Embedding Model 模块、Retrieval 模块和 Reranking 模块
- 跑通实验 demo

六、 初步结论

经过对实验进度的进一步推进，我们的**结论如下：**

通过对系统架构进一步完善，绘制了系统架构图，分出来了各个需要的模块，确定了实验的可行性，使得我们实验的目的性更加明确。此外我们对各个模块还都进行了更加细致的设计，以便于后续代码可以完成的更加有效率。已完成的模块也满足我们的预期要求。

在接下来的工作中，我们将进一步完善系统的细节，并进行更全面的实验评估。具体来说我们需要完成待完成的模块、可能会根据需求对已有的模块进行微调，在实验的过程中一步步完善我们的程序。

七、 遇到的问题和可能的解决办法

问题：

1、数据质量问题：构建一个全面而准确的知识库是任务的基础。在我们构建的过程中，我们遇到了一些数据缺失、错误标注等问题，需要进行对数据清洗和验证。此外，大型语言模型的训练通常需要大规模的数据集。确保训练数据的质量和多样性是一个挑战，需要处理数据偏置、噪声和不一致性等问题。

2、此外，我们还考虑到模块间的协调和接口设计的问题：在检索模块和生成模块之间的数据交互和接口设计时需要仔细考虑。确保数据的正确传递和一致性，以及接口的稳定性和灵活性。而且将检索模块和生成模块集成到一个统一的系统中我们可能会面临技术兼容性、

性能匹配等问题。我们需要确保模块之间的协调和整体系统的稳定性。

可能的解决办法：

1、数据质量问题：我们可以进行数据的清洗和验证，并引入人工审核和纠正机制。

2、模块间的协调和接口设计：我们可以定义更为清晰的数据传递格式和接口规范，从而进行模块间的充分测试和集成验证。

八、 后续计划

05.26	06.02	06.09	06.16	06.18
开题报告	概要设计文档	模块1，3，4	完整的系统	结题 PPT
	模块2，5，6		测试代码	
	Demo 跑通		详细设计文档	