北京邮电大学人工智能学院

自然语言处理实验开题报告

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 5 月 22 日

1. 任务描述与预期目标

1. 检索策略优化

目标：尝试更多的组块拆分策略，以及更好的检索算法和重排算法。

具体操作

- **组块拆分策略**：尝试不同的文本分块方式（固定长度、基于句子和语义分块等）。
- **检索算法**：实验不同的检索算法（如 BM25、DPR、ColBERT 等）来提高检索性能。
- **重排算法**：使用基于 Transformer 的重排模型（如 BERT、T5）进行结果重排，以提高检索精度。

2. 微调 BGE 的检索和重排模型

目标：微调 BGE 检索以及重排模型。

具体操作：

- **数据划分**：将 CRAG 全量训练集进行划分。
- **模型微调**：按照 FlagEmbedding 微调指南 进行操作。
- **正负样例构造**：
- **人工标注**：手动标注 query 对应的 chunk 正例。
- **自动判定**:使用 LLM 通过 prompt 判定 query 和 chunk 的相关性，自动生成正负例。

3. 对 LLM 进行进一步预训练或微调

（完成内容根据任务 1、2 的完成情况以及算力资源进行动态调整）

目标：根据 CRAG 数据对 LLM 进行进一步的预训练或有监督微调。

具体操作：

- **数据准备**：准备 CRAG 数据集进行 LLM 的预训练或微调。
- **选择框架**：根据算力情况选择适合的 LLM 框架进行训练：
- **llama-factory**：用于构建和训练大规模语言模型。
- **megatron-lm**：NVIDIA 提供的高效分布式语言模型训练框架。
- **训练配置**：根据数据和资源配置训练参数，进行 LLM 的预训练或有监督微调。

2. 相关工作调研

首先是对检索增强生成（Retrieval Augmented Generation，RAG）有一个细致的了解。它是一种强大的自然语言处理框架，结合了检索和生成技术，用于提高生成文本的质量与相关性。RAG 模型利用检索和生成模型的优势，可以生成更准确、更与上下文相关的文本输出。

RAG 包含两个主要组件：检索器和生成器。

1. 检索器： 检索器的功能是从大规模文本语料库中检索与查询或上下文相关的信息。它采用信息检索技术，如 TF-IDF（词频-逆文档频率）或语义相似度度量，对文档或片段进行排序和检索，以获取与输入上下文语义相关的信息。检索器的目标是找到最相关的信息，以用于增强生成过程。

2. 生成器： 生成器是一个语言模型，通常在大规模文本语料库上进行预训练，它使用检索器返回的信息和输入上下文作为输入，生成文本输出。生成器可以使用循环神经网络（RNN）或变换器（Transformer）等技术，以生成连贯、与上下文相关的文本。生成器利用检索到的信息来引导生成过程，从而产生更准确、与上下文相关的输出。

所以，RAG 的关键思想是利用检索组件来增强生成过程。通过整合从文本语料库中检索到的相关信息，生成器可以更好地理解上下文，并生成更准确、与上下文相关的文本。RAG 模型适用于各种 NLP 任务，如问答系统、摘要生成、对话系统和内容生成等。

有了以上了解，我们就明确了目前的主要任务，就是选择一种更契合我们任务要求的检索器与生成器。下面是一些我们可能会用到的、常见的检索器与生成器：

检索器：

1. 基于关键词的检索器：这种检索器使用关键词匹配的方法，在大规模文本语料库中检索与查询或上下文相关的文档或片段。可以根据关键词的匹配程度对文档进行排序，并返回最相关的信息。它虽然具有简单性和可解释性，但是其限制也较为明显，即无法处理词义的多义性或上下文的语义关系。
2. TF-IDF（词频-逆文档频率）检索器：TF-IDF 检索器使用 TF-IDF 算法衡量文档中词的重要性，并根据词的 TF-IDF 值对文档进行排序。可以从文本语料库中检索与查询或上下文语义相关的文档或片段。
3. BERT-based 检索器：这种检索器基于预训练的 BERT（Bidirectional Encoder Representations from Transformers）模型，将查询或上下文输入编码为向量表示，并计

算查询与文档之间的语义相似度。可以通过计算余弦相似度或其他相似度度量方法，检索与查询或上下文最相关的文档或片段。

4. Dense Vector 检索器：这种检索器使用密集向量表示文档或片段，例如使用 Doc2Vec 或 Sentence-BERT 等技术。它将查询或上下文与文档的向量进行比较，以找到最相关的信息。

5. 知识图谱检索器：这种检索器利用知识图谱中的结构化数据，如实体关系和属性，来检索与查询或上下文相关的信息。它可以通过查询图谱中的实体或关系，获取相关的知识和信息。

6. 向量化检索（Vectorized Retrieval）：向量化检索是使用向量空间模型将文档和查询转换为向量表示，并通过计算它们之间的相似度来进行检索的方法。

生成器：

1. 基于循环神经网络（RNN）的生成器：这种生成器使用循环神经网络（如 LSTM 或 GRU）作为基础模型，以序列方式生成文本。它可以接受检索器返回的信息和输入上下文作为输入，并生成与上下文相关的连贯文本输出。

2. 变换器（Transformer）生成器：这种生成器基于 Transformer 架构，利用自注意力机制以及多头注意力机制，能够并行处理输入序列。它可以接受来自检索器的信息和输入上下文，并生成与上下文相关的输出。

3. 语言模型生成器：这种生成器基于大规模预训练的语言模型，如 GPT（Generative Pre-trained Transformer）系列模型。它可以利用检索器返回的信息和输入上下文，以自回归方式生成连贯、与上下文相关的文本。

4. 条件生成器：这种生成器可以根据给定的条件或控制指令来生成文本。它可以利用检索器返回的信息和上下文，结合额外的条件信息，生成与指定条件相匹配的文本。

5. Top-k 采样生成（Top-k Sampling Generation）：Top-k 采样是一种用于生成文本的策略，它允许从预训练的语言模型中根据概率分布选择最高的 k 个概率进行采样。

我们最后将根据实验所需，选择合适的检索器与生成器，如果时间等条件允许，我们还可以通过实践来测试不同检索器与生成器的差异。

3. 技术选型与开发环境

3.1 技术选型

A. 模型选择

经过初步的横向对比，考虑使用 BGE-base 模型进行实验，因为其在同规模的模型中对中文自然语言的效果较好。此外，我们还考虑使用其他的一些相似模型进行对照试验，以得到更加多样化的实验结果，也便于进行更详细的分析。

B. Retrieval Augment Methods

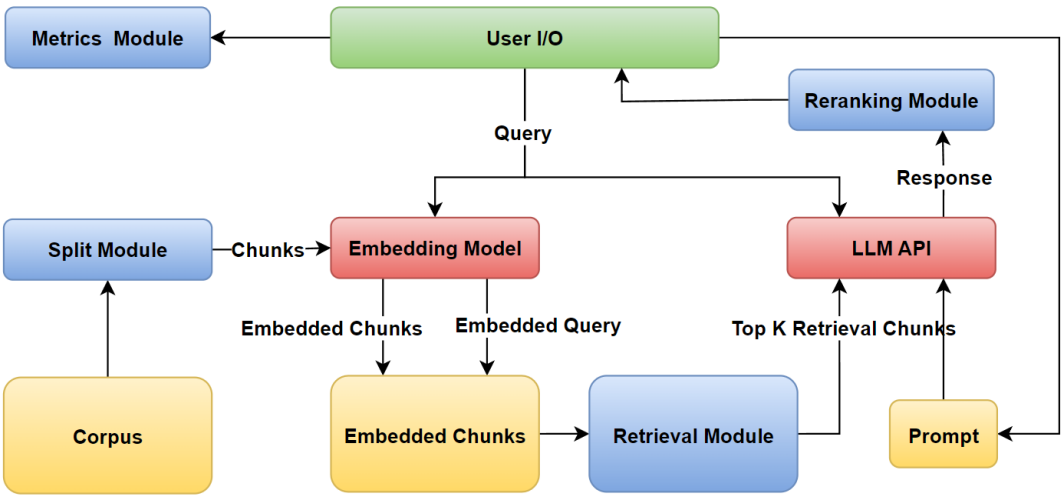
考虑采用向量化检索的方法，利用 Embedding 模型对输入进行 Embedding 得到语义向量，并基于向量相似度（这里使用余弦相似度，是因为相较于其他的而言易于实现且效果较好）进行检索，向量化检索的方法可以较好地捕获语义相关性，提高检索精度。

C. Generation Optimize Methods

采用 Top-k 采样的方式进行文本生成，可以生成更流畅、凝练的文本内容。同时，可以调整 temperature 参数以控制生成文本的多样性。

D. 系统架构设计

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：



E. 实验设计

i. 对于 QA 任务，评估指标为：答案准确率、BLEU 分数等。

ii. 对于摘要生成任务，评估指标为：ROUGE-1/2/L 等。

iii. 对于整体来说，需要评估生成文本的流畅度和贴合度。

3.2 开发环境

- Python 3.10
- Windows / Mac / Linux
- CUDA 12.2

更多第三方库细节将在开发过程中不断测试和更新。

4. 实验分工与检查点

4.1 实验分工：

孟祥超：

1. 开题报告 2（相关工作调研）
2. 跑通 3 个 Demo
3. 概要设计文档 5+6+7+8（已完成工作，初步结论，问题与办法，后续计划）
4. 详细设计文档 4+5（实验结果与分析，总结）
5. 演示视频录制

廉　涟：

1. 开题报告 1（任务描述与预期目标）
2. 协助跑通 Demo
3. 概要设计文档 3（接口设计）
4. 原型系统开发，模块 1+2+6（Corpus Seg, User I/O, Metrics Calc）
5. 详细设计文档 1+2+3（数据，环境，模型与方法）
6. 模块整合与测试

巩羽飞：

1. 项目整体进度管理与监督
2. 开题报告 3+4

3. 概要设计文档 1+2+4（系统架构，模块划分，技术方案）
4. 原型系统开发，模块 3+4+5（Embedding, Retrieval, Reranking）
5. 协助模块整合与测试
6. 结题 PPT 制作

4.2 检查点表：

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**	