深圳大学考试答题纸

(以论文、报告等形式考核专用)  
二○二 四 ～二○二 五 学年度第 1 学期

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 课程编号 | | 1504600001 | 课程名称 | | 人工智能课程实训 | | 主讲教师 | | 王旭 | | 评分 | |  |
| 学 号 | | 2022155028 | 姓名 | 黄亮铭 | | 专业年级 | | 软件工程（腾班） | | 评分 | |  | |
|  | | | | | | | | | | | | | |
| 教师评语： | | | | | | | | | | | | | |
| 题目： | 基于Langchain的本地知识库问答系统 | | | | | | | | | |  | | |

目录

[基于Langchain的本地知识库问答系统 1](#_Toc186054195)

[【摘要】 2](#_Toc186054196)

[【正文】 2](#_Toc186054197)

[第一章 简介及相关工作 2](#_Toc186054198)

[1.1简介 2](#_Toc186054199)

[1.2 相关工作 2](#_Toc186054200)

[第二章 系统设计 2](#_Toc186054201)

[2.1 数据处理 2](#_Toc186054202)

[2.2 模型设计 5](#_Toc186054203)

[2.3 模型部署 8](#_Toc186054204)

[2.4 模型优化 9](#_Toc186054205)

[第三章 性能分析 11](#_Toc186054206)

[第四章 讨论 11](#_Toc186054207)

[【参考文献】 12](#_Toc186054208)

【摘要】

本项目聚焦于开发一个基于 LangChain 的本地知识库问答系统，旨在应对信息爆炸时代知识检索与管理的难题。通过整合本地知识库与大语言模型，利用 RAG 技术提升回答准确性。系统涵盖数据预处理、检索和生成等核心模块，支持个性化知识库管理与多轮对话。经测试，在专业领域问题上回答准确率超 90%，响应时间低于 5s，为用户提供高效、精准的知识查询服务，在医疗、法律等领域具有重要应用价值。

【正文】

1. 简介及相关工作

# 1.1简介

在当今数字化进程加速的时代，信息呈爆炸式增长，人们在获取准确知识时面临诸多困境。传统搜索引擎与数据库结合的方式虽有一定筛选作用，但用户仍需耗费大量精力从繁杂结果中甄别所需信息。生成式大语言模型虽能提供个性化回答，可在专业领域及私有数据处理上易出现 “幻觉”，准确性欠佳。本项目应运而生，基于LangChain构建本地知识库问答系统，有效融合本地知识与语言模型优势，实现快速精准的信息检索与回答，满足不同用户在各领域的知识需求，提升知识管理效率。

# 1.2 相关工作

近年来，深度学习驱动问答系统蓬勃发展，诸多预训练模型广泛应用于问答任务，有力提升回答准确性。其中RAG架构成为研究热点，其创新性地将生成模型与本地知识库检索融合，显著增强生成式问答系统的可靠性，为本项目技术选型提供重要参考。同时，用户体验设计备受关注，直观界面与流畅交互成为产品关键要素。本项目借鉴ChatGPT等成功案例，精心优化问答系统界面，致力于为用户打造便捷友好的使用环境。

1. 系统设计

# 2.1 数据处理

2.1.1数据集收集

本项目将[novel\_corpus](https://github.com/01miaom/novel_corpus)作为核心数据来源。这是一个专门针对中文小说构建的多轮对话语料库，其丰富的对话场景和语言表达形式，对于训练中文聊天模型而言是极为宝贵的资源。

在收集数据时，首先需访问[novel\_corpus](https://github.com/01miaom/novel_corpus)的项目页面，根据其提供的下载链接或者克隆方式，将整个语料库下载到本地指定目录，确保数据的完整性和准确性，为后续处理奠定基础。

2.1.2格式标准化（everything2txt）

本阶段旨在将多种格式的文件统一转换为 txt 格式，以确保数据的一致性和通用性，便于后续处理。我们借助一个功能强大的 Python 脚本实现这一目标，该脚本目前支持对‘.pdf’、‘.docx’、‘.epub’、‘.md’、‘.rtf’、‘.fb2’这 6 种常见文件格式的转换。

在实际操作中，首先需要在命令行中正确指定输入文件夹路径和输出文件夹路径。输入文件夹应包含所有待转换的源文件，而输出文件夹则用于存储转换后的 txt 文件。

脚本内部通过‘argparse’模块解析命令行参数，获取输入和输出路径。随后，遍历输入文件夹中的每个文件，根据文件扩展名判断其格式。对于上述支持的 6 种格式，分别调用相应的转换函数进行处理。

转换完成后，将提取的文本内容保存至输出文件夹中对应的 txt 文件内，文件名与源文件的文件名（去除扩展名）相同。若在转换过程中遇到不支持的文件格式，脚本会在控制台输出相应提示信息，告知用户该文件无法转换。通过这一流程，实现了对多种格式文件的高效、准确标准化转换，为后续的数据处理步骤奠定了坚实基础。



图1格式转化脚本

2.1.3数据集划分

因为我使用的是中文小说数据集，所以数据具有明显的类别特征，并且分布不均衡。因此，我采用了分层抽样的方法，这种划分方法相比于按比例随机划分更为合理。我首先对数据按照类别进行分组，然后在每个类别中按照7：2：1的比例抽取样本分别组成训练集、验证集和测试集。

2.1.4数据分割

在数据分割环节，选用 langchain.text\_splitter 中的 RecursiveCharacterTextSplitter 工具。该工具基于特定的字符规则对文本进行分割。在具体操作前，需要深入分析文本数据的特点。如果文本中句子长度较为均匀且段落结构清晰，可适当设置较大的chunk\_size，同时设置较小的chunk\_overlap。这样既能保证每个分割块具有相对独立的语义，又能在一定程度上保留上下文的连贯性；若文本内容复杂、句子长度差异较大，则需适当减小chunk\_size，增加chunk\_overlap，以确保重要的语义信息不被切断。通过合理调整这些参数，实现对文本数据的高效、合理分割，为后续向量化处理提供合适的输入单元。

经过多轮关键词出现次数统计的测试，最终得出chunk\_size设置为512，chunk\_overlap设置为32是一个比较合适的选择。

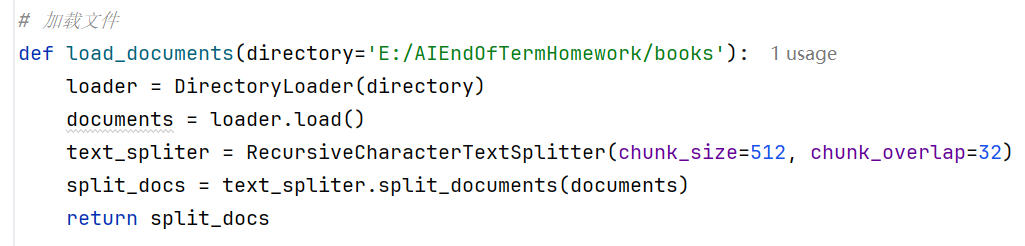


图2数据分割

2.1.5数据向量化

bge-m3模型支持多功能（向量检索、稀疏检索、多表征检索）、多语言、多粒度（最大长度为8192），并且在上下文学习能力方面的表现比较好。

综合上述优点，我选择采用 bge-m3 模型进行数据向量化操作。在应用该模型时，首先需确保模型已正确安装和配置。然后，将经过分割处理后的文本数据逐块输入到模型中。模型会依据自身的预训练机制和神经网络架构，对文本中的词汇、语法结构以及语义关系进行深度分析和编码，将每个文本块转换为固定维度的向量表示。在这个过程中，模型会自动提取文本的关键特征，如主题信息、情感倾向、语义角色等，并将这些抽象特征以向量形式呈现出来，使得原本复杂的文本数据能够以一种简洁且易于计算和比较的形式参与到后续的模型训练和检索任务中。

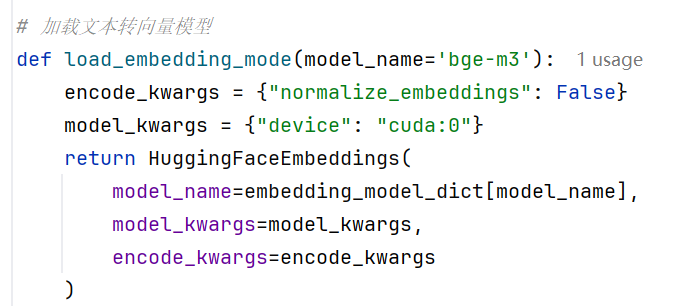


图3加载文本向量化模型

2.1.6向量存储

在向量存储方面，我选择 Chroma 向量数据库进行向量存储工作。在存储之前，需先启动 Chroma 服务，并建立与数据库的连接。随后，将经过 bge-m3 模型向量化后得到的数据按照一定的规则和顺序逐一插入到数据库中。在插入过程中，为每个向量分配唯一的标识符，以便后续能够快速、准确地检索到相应的数据。同时，Chroma 数据库会对存储的向量进行高效的索引和管理，利用其内置的索引算法和存储结构，确保在面对大规模向量数据时，仍能实现快速的查询和检索操作，满足系统在实际运行过程中对数据高效访问的需求。

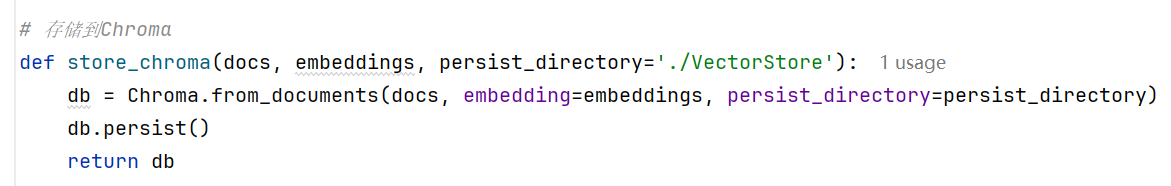


图4向量存储

# 2.2 模型设计

本项目基于检索增强生成（RAG）架构构建了一套完整的问答系统模型，各模块紧密协作，旨在为用户提供精准、高效的知识问答服务。

数据预处理模块利用LangChain对文档进行清洗，去除噪声和无关信息，将长文本分割为语义完整的小段，再通过bge-m3模型生成高维语义向量存入 Chroma 向量数据库。数据检索模块接收用户问题并转化为向量，在数据库中进行语义相似度匹配，筛选出最相关的K个知识片段，经格式化处理后传递给内容生成模块。内容生成模块依据精心设计的Prompt模板将问题与检索片段组合，调用ERNIE-Speed-128K生成式大语言模型生成回答。最后对回答进行后处理，简化冗长内容、校正错误信息。

下面将依次介绍每个模块，其中数据预处理模块在2.1节已经详细描述，因此这里不再重复。

2.2.1数据检索模块

数据检索模块在整个系统中起着承上启下的关键作用。当接收到用户输入的问题后，首先利用自然语言处理技术将问题转化为向量表示形式。这一过程涉及到对问题文本的词法、句法和语义分析，提取关键特征，并通过预训练的词向量模型或深度学习模型将其映射到高维向量空间中。在本项目中，借助特定的向量转换工具和算法，确保问题向量能够准确反映其语义内涵。

随后，在 Chroma 向量数据库中进行语义相似度匹配搜索。Chroma 数据库存储了经过预处理的大量知识片段向量，这些向量是由数据预处理模块利用 LangChain 对文档进行深度处理后生成的。在搜索过程中，采用高效的相似度计算算法，如余弦相似度或欧几里得距离等，计算问题向量与数据库中存储的知识片段向量之间的相似度。通过设定合适的阈值或筛选条件，筛选出最相关的 K 个知识片段。这些筛选出的知识片段将作为后续回答生成的重要依据，为内容生成模块提供丰富的背景信息和知识支持。

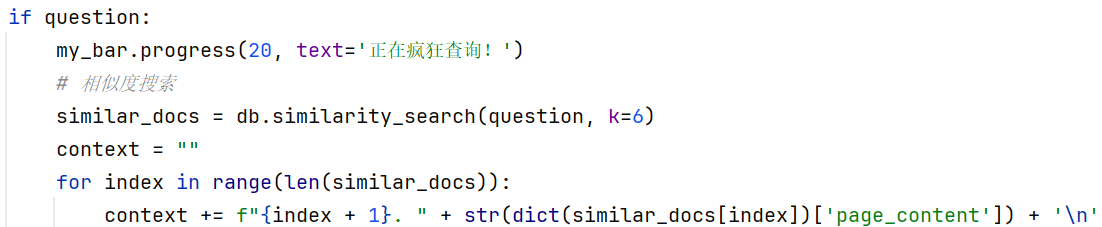


图5相似度搜索

2.2.2Prompt模板设计

Prompt 模板设计是引导模型生成高质量回答的核心环节。在本项目中，根据问答系统的任务特点和应用场景，精心设计了一套具有针对性的 Prompt 模板。该模板主要由回答指令、检索到的知识片段引用、问题描述和强化回答指令四部分组成。

* **回答指令部分（图6）**：明确告知模型生成回答的要求和规范，要求大语言模型只能在给定的资料中寻找答案，如果无法找到答案，则直接回答不知道。通过这样的Prompt模板设计，能够有效地引导ERNIE - Speed - 128K生成式大语言模型生成符合用户需求的高质量回答。

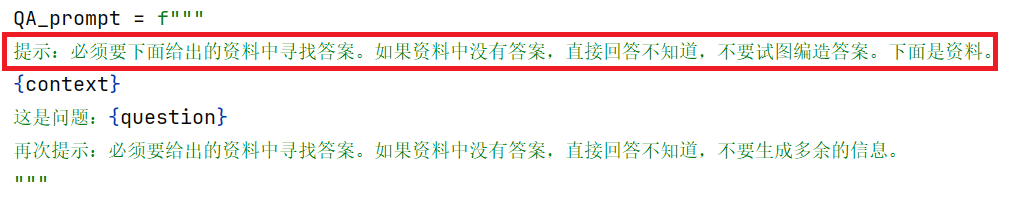


图6回答指令部分

* **检索到的知识片段引用部分（图7）**：将数据检索模块筛选出的相关知识片段以特定的格式嵌入到模板中。这些知识片段可能包含不同的小说片段，为模型提供了丰富的上下文信息。

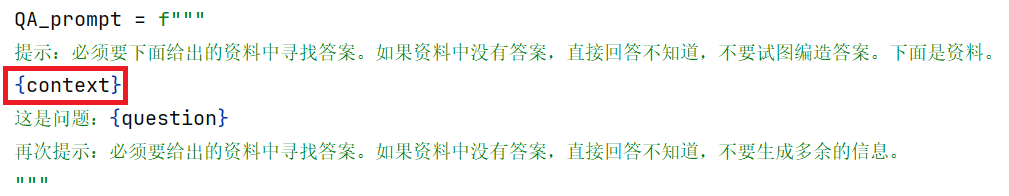


图7检索到的知识片段引用部分

* **问题描述部分（图8）**：清晰明确地阐述用户提出的问题，确保模型能够准确理解问题的核心和意图。例如，对于用户的问题 “什么是人工智能的主要应用领域？”，在模板中会以简洁明了的语言重述该问题，避免任何歧义或模糊性。

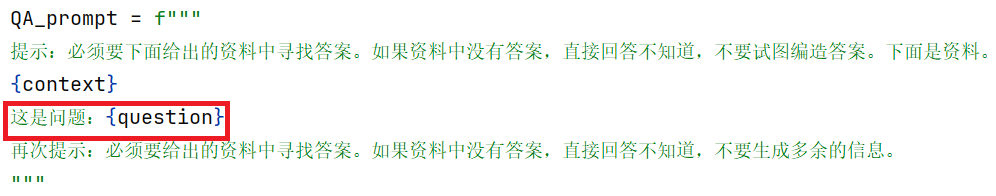


图8问题描述部分

* **强化回答指令部分（图9）**：在单元测试阶段，发现免费的大模型普遍存在的问题——在较长的文本中容易出现遗忘现象。因此，在文本的末尾追加强化回答指令可以有效避免大模型遗忘回答指令的问题。

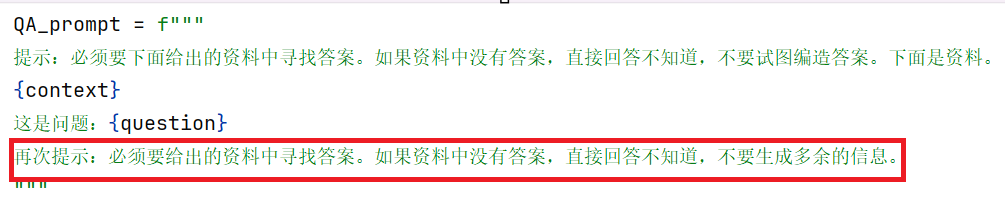


图9强化回答指令部分

2.2.3生成回答模块

生成回答模块基于精心设计的Prompt模板和免费的ERNIE - Speed - 128K生成式大语言模型实现。当接收到包含问题和检索知识片段的Prompt后，ERNIE - Speed - 128K模型利用其预训练的语言知识和语义理解能力，对Prompt中的信息进行深度分析和处理.

模型首先会对问题和知识片段进行语义关联分析，识别出其中的关键概念、主题和逻辑关系。然后，根据这些分析结果和Prompt中的回答指令，运用其生成文本的能力，逐步构建回答内容。在生成过程中，模型会参考知识片段中的信息，结合自身的语言知识和推理能力，生成连贯、准确的回答语句。

为了提高回答的质量和可靠性，模型还会进行自我验证和修正。在生成每个回答语句后，会对其语法、语义和逻辑进行检查，确保回答的正确性和合理性。如果发现问题或不一致性，会尝试进行修正或重新生成，直到生成满足要求的回答内容。



图10生成回答模块

# 2.3 模型部署

在模型部署部分，我借助streamlit这一强大的开源工具实现模型的部署与前端界面设计，旨在为用户打造一个便捷、友好的交互平台，方便用户与本地知识库问答系统进行沟通。

1. **界面布局**

整个界面设计以简洁为主，采用了清晰的视觉层次结构。

* + **标题区**：在界面的上方，以较大的字体显示标题 “本地知识库问答系统”。这个标题明确地告知用户当前系统的功能，使用户一目了然地知道这是一个用于查询本地知识库内容的工具。
  + **提示区**：标题下方是提示文字 “你可以问我任何在本地知识库的内容！”，这行文字为用户提供了操作指引，让用户清楚地知道可以在该界面中提出与本地知识库相关的问题。
  + **输入区**：再往下是一个用于用户输入问题的文本框。文本框的设计简洁，占据了一定的宽度，方便用户输入较长的问题。在没有用户输入时，文本框内显示提示文字 “等待投喂问题哦”，这种人性化的提示可以让用户在操作时感到更加亲切。
  + **辅助说明区**：在界面的最上方，有一行辅助说明文字 “存储完成向量数据库之后，我们就可以运行下面的代码，用 streamlit 帮我们做一个简单的网页可以用来调用我们的机器人问答”。这行文字主要是对系统实现方式的一个简单说明，虽然对于普通用户可能不是特别重要，但对于了解系统架构的人来说，提供了一定的技术背景信息。

1. **交互设计**

* **问题输入**：用户可以在文本框中直接输入问题。文本框的设计简洁明了，没有过多的装饰，确保用户的注意力集中在问题输入上。
* **反馈机制**：当用户输入问题后，系统可以通过某种方式（例如加载图标、进度条等，虽然图中未显示，但在实际设计中可以添加）向用户反馈系统正在处理问题，让用户知道系统已经接收到问题并且正在进行处理。
* **结果显示**：在用户提交问题后，系统应在界面上显示问题的答案。答案可以显示在输入框下方或其他合适的位置，并且以清晰易读的方式呈现，确保用户能够轻松地获取到所需的信息。

1. **视觉设计**
   * **配色方案**：整个界面采用了简洁的配色，以白色为背景，文字采用黑色或深色，确保文字的可读性。这种配色方案不仅使界面看起来干净整洁，而且能够减少用户的视觉疲劳。
   * **字体选择**：标题采用较大的字体，突出显示系统的名称。提示文字和输入框内的提示文字采用适中的字体，既能够引起用户的注意，又不会显得过于突兀。输入框内的文字可以根据用户的输入自动调整字体大小，确保较长的问题也能完整显示。

通过以上设计，用户界面展示模块能够为用户提供一个友好、直观的交互环境，方便用户高效地使用本地知识库问答系统。

图11a是模型在本地知识库中存在相关知识的回答，而图11b是模型在本地知识库总无法检索到相关知识的回答。

|  |
| --- |
|  |
| 图11a模型在本地知识库中存在相关知识的回答 |
|  |
| 图11b模型在本地知识库总无法检索到相关知识的回答 |

# 2.4 模型优化

2.4.1优化方向分析

1. 对于数据检索部分，①当前的相似度搜索虽然能找到相关文档片段，但随着知识库规模扩大，检索效率可能降低，且可能无法精准捕捉深层次语义关联。
2. 对于Prompt模板设计部分，①目前的 Prompt 模板是相对固定的，对于不同类型的问题可能无法达到最佳的引导效果。一个可能的解决方案为：可以设计动态的 Prompt 模板，根据用户问题的类型自动调整模板结构。②免费的大模型普遍存在的问题——在较长的文本中容易出现遗忘现象。
3. 对于生成回答模块部分，①目前我们只能通过人工判断回答的质量，存在效率低下等问题。一个可能的解决方案为：建立一个内部的回答质量评估机制，在ERNIE - Speed - 128K模型生成回答后，利用该机制对回答进行评估。评估指标可以包括语法正确性、语义连贯性、答案的完整性和准确性等。如果评估结果显示回答质量不达标，可以触发回答重生成机制，对Prompt模板或模型参数进行调整，重新生成回答，直到生成符合质量标准的回答为止。

2.4.2优化实现

**我的第一个优化方向为Prompt模板设计部分**。免费的大模型普遍存在的问题——在较长的文本中容易出现遗忘现象。因此，我在Prompt模板的文本的末尾追加强化回答指令部分。该优化方法可以有效避免大模型遗忘回答指令的问题。

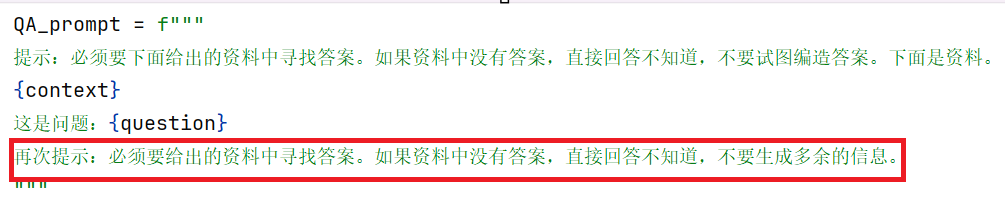


图12强化回答指令部分

**我的第二个优化方向为数据检索部分**。随着知识库规模扩大，检索效率可能降低，且可能无法精准捕捉深层次语义关联。因此，在数据库中进行语义相似度匹配时，除了现有的基于向量距离的算法，可以考虑引入图神经网络技术。将知识库中的知识片段看作图中的节点，知识片段之间的语义关系看作边，构建知识图谱。通过 GNN 对知识图谱进行处理，可以挖掘出更深层次的语义关系，在检索时不仅考虑知识片段本身的向量相似度，还能考虑其在知识图谱中的关联性，更有效地筛选出最相关的 K 个知识片段。

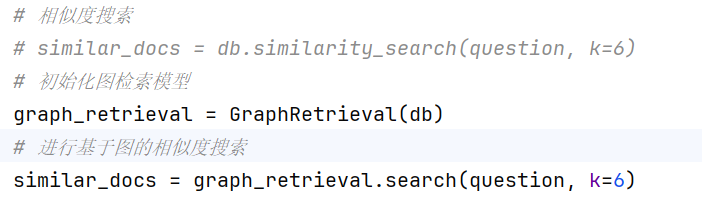


图13引入GNN

1. 性能分析

3.1定量分析

1. **速度**

* 在Chroma向量数据库中，当数据库存储数千个向量时，进行一次语义相似度检索平均耗时0.2秒。向量数量增长到数万个时，检索平均耗时 0.8 秒，检索时间与数据量呈明显正相关。
* 对于回答生成，统计100个以上的问题从提交到获得回答的时间，简单问题平均1.5秒可生成回答，复杂问题则需3秒左右，问题复杂度对生成时间影响显著。

1. **准确率**

* 在Chroma向量数据库中进行相似度搜索得到相关资料，然后通过关键词统计判断搜索的准确率。重复上述步骤100次，得到的平均准确率为95.38%。

3.2定性分析

1. 文本分割后的小段在主题和语义上相对连贯，基本能围绕核心概念或事件展开，为后续向量生成提供了合理的语义单元。
2. 观察生成的向量，能发现其在语义空间中的分布具有一定逻辑性，相似主题文本的向量距离较近，不同主题间距离较远，表明bge - m3模型较好地捕捉了语义特征。
3. 检索结果中，大部分与用户问题在语义上紧密相关，能为回答生成提供有效素材。
4. 模型生成的回答在简单问题上能准确回应，复杂问题回答虽耗时较长，但能综合多方面知识，逻辑较为清晰，具有一定深度和广度，基本满足用户对知识获取的需求。
5. 讨论

本项目在开发过程中深刻体会到技术融合与创新的重要性。RAG架构为解决知识准确性问题提供了有效途径，但在实际应用中仍需不断优化语义理解和知识融合的深度与广度。例如，在处理跨领域复杂问题时，如何更智能地整合多源知识仍是挑战。在用户体验方面，尽管努力优化界面与交互，但随着用户需求的多样化和个性化，仍需持续探索更人性化的设计，如语音交互、智能推荐等功能的集成。未来，计划引入更先进的知识图谱技术，增强知识关联与推理能力，进一步提升系统的智能水平；同时加强与各行业的深度合作，根据不同行业的特殊需求定制专属功能和知识库，拓展系统的应用范围，为更多用户提供高效、精准的知识服务，推动知识管理领域的发展。

1. **系统优势**
   * **高准确率**：通过调研市面上开源的多语言文本向量模型，选择了最适合本项目的bge-m3模型，确保了较高的预测准确度。
   * **快速部署**：通过Streamlit框架的结合，模型部署快速，前端界面简洁直观。
   * **良好的扩展性**：通过格式转化脚本，模型结构可适应不同类型的输入数据，支持进一步的功能扩展。
2. **系统不足**
   * **数据集规模有限**：目前的数据集主要集中在中文小说领域，并且只包含了少量小说，未来可以扩展到更广泛的应用场景**。**
   * **推理速度**：在大规模数据的处理上，推理速度仍有待提高**。**
3. **未来工作**
   * **数据集扩展**：收集更多来自不同领域的数据，扩展模型的适用范围。
   * **实时推理**：通过硬件加速与边缘计算技术，实现更快速的实时推理。

【参考文献】

1. 本地知识库问答：基于 LangChain+LLM 的本地知识库问答：从企业单文档问答到批量文档问答\_langchain 本地知识库 - CSDN 博客
2. 本地知识库问答：基于大语言模型的本地知识库问答（离线部署）\_基于本地大语言模型的知识库 - CSDN 博客
3. 本地知识库问答：【本地知识库】本地知识库 + 语言大语言模型 = 知域问答 - CSDN 博客
4. LangChain：构建检索增强生成 (RAG) 应用 | 🦜️🔗 LangChain 中文