### 项目报告书模板

**奖励与认证：**

参加者可以根据自己的能力选择在规定时间完成任意一个类型的RAG智能对话机器人：

* 基础版：完成文字形式的RAG智能对话机器人将获得NVIDIA AI-Agent训练营结业证书。
* 进阶版：如果在功能整合部分加入了进阶的功能(语音/Agent/多模态等)，除了结业证书外，还将获得一份精美礼品。

**提交说明：**

1. 请按照本模板撰写项目报告书。
2. 请将完成的报告书发布到您的社交平台，包括但不限于知乎、CSDN、Github等，并确保内容公开可访问。
3. 填写并将文章链接提交到此处：(<https://jsj.top/f/i4QM5Z>) ，以便我们审核您的项目报告并寄送相应的奖励与认证。

**模板：**

## NVIDIA AI-AGENT夏季训练营

项目名称：AI-AGENT夏季训练营 — RAG智能对话机器人

报告日期：2024年8月18日

项目负责人：邓杰夫

#### 项目概述（必写）：

本项目旨在利用NVIDIA NIM微服务架构构建一个基于检索增强生成（RAG）技术的问答机器人，能够根据百度弱智吧提取的问答进行智能回答。该系统不仅可以为用户提供实时的问答服务，还能根据用户的提问上下文，智能匹配最相关的回答。

### 应用场景：

**社交媒体互动**：该系统可以集成到社交媒体平台中，与用户进行互动，提供有趣的回答和建议。

### 项目亮点

**数据隐私保护**：NVIDIA NIM的微服务架构确保所有数据处理都在本地进行，用户的数据隐私得到了有效保护，特别适合处理敏感信息的场景。

**高效的模型部署**：NIM支持快速部署和扩展AI模型，开发者可以利用预构建的容器，减少配置时间，快速上线问答机器人。

#### 技术方案与实施步骤

* 模型选择（必写）： 详细描述项目采用的技术方案，包括大模型的选择理由、RAG模型的优势分析。

本项目采用了**Meta Llama 3.1 405B Instruct**作为大模型，并使用**NV-Embed-QA**作为嵌入模型，结合**Faiss**生成本地向量数据库，并利用**LangChain**进行检索增强生成（RAG）问答系统的构建。以下是对所选技术的详细分析。

### 大模型选择理由

* **强大的性能**：Llama 3.1 405B是Meta推出的最先进的开源AI模型，拥有4050亿个参数，能够处理复杂的任务，包括多语言翻译、数学问题、工具使用等。其强大的推理能力使其在多种应用场景中表现出色。
* **扩展的上下文长度**：该模型支持高达128K的上下文长度，适合处理长文本和复杂问题，从而提高了对用户查询的理解和响应能力。
* **多语言支持**：Llama 3.1支持多种语言，使其在全球化应用中具备更高的适用性，能够满足不同用户的需求。

### RAG模型的优势分析

* **增强的知识获取**：RAG通过结合检索机制与生成能力，使得模型能够在生成回答时访问外部知识库。这种方法弥补了大模型在特定领域知识和最新信息方面的不足，确保回答的准确性和时效性。
* **提升生成质量**：RAG模型通过将外部检索到的信息与用户的查询结合，能够生成更具上下文相关性的回答，提升了用户交互的体验和满意度。
* **高效的相似性搜索**：通过使用Faiss向量数据库，RAG系统能够快速高效地进行相似性搜索，确保在海量数据中迅速找到最相关的信息。这种高效性在处理大规模数据时尤为重要。
* **可扩展性与灵活性**：RAG架构的设计允许开发者根据需求进行扩展，支持多种数据源和模型的集成，使系统能够适应不同的业务场景和应用需求。

通过以上技术方案的选择与应用，本项目旨在构建一个高效、智能、灵活的问答机器人，能够在多种场景下提供优质的用户体验。

* 数据的构建（必写）： 说明数据构建过程、向量化处理方法及其优势。

数据构建与向量化处理过程

本项目的数据构建过程主要包括从百度弱智吧经典问答中提取数据、保存为本地文本文件，并利用NV-Embed-QA模型进行向量化处理。以下是详细的步骤说明及其优势分析。

数据构建过程

1. **数据提取**：首先，从百度弱智吧经典问答中提取相关问答数据。这一过程涉及到数据的筛选与清洗，确保提取的内容具有代表性和高质量。
2. **数据保存**：提取后的数据被保存为本地的TXT文件，便于后续的处理和访问。保存为文本文件的格式简洁明了，易于读取和操作。
3. **文本分割**：在向量化之前，使用CharacterTextSplitter对提取的文本进行分割。每个文本块的大小设定为512个字符，这样可以确保每个块在处理时不会过于庞大，同时保留足够的上下文信息。
4. **元数据处理**：在分割文本的同时，为每个文本块生成相应的元数据（如源信息），以便后续检索时能够追溯信息来源。

向量化处理方法在向量化过程中，采用了以下步骤：

1. **加载或创建向量数据库**：首先检查指定的向量存储目录是否存在。如果存在，则加载之前保存的FAISS向量数据库；如果不存在，则进行新数据库的创建。
2. **文本向量化**：使用FAISS.from\_texts(docs, embedder, metadatas=metadatas)方法，将分割后的文本块通过NV-Embed-QA模型进行向量化处理。这一过程将每个文本块转换为高维向量，便于后续的相似性检索。
3. **保存向量数据库**：向量化完成后，使用store.save\_local(embed\_dir)将生成的向量数据库保存到本地，以便后续快速加载和使用。

向量化处理的优势

1. **高效的相似性检索**：FAISS是一个高效的相似性搜索库，能够快速处理大规模向量数据，支持高效的最近邻搜索。这对于大规模问答数据的实时检索尤为重要。
2. **上下文保持**：通过将文本分割为适当大小的块，向量化处理能够保留文本的上下文信息，从而在生成回答时提供更准确的内容。
3. **灵活性和可扩展性**：向量化处理方法支持多种文本输入格式和模型，使得系统能够灵活适应不同的数据源和需求。
4. **高质量的嵌入表示**：NV-Embed-QA模型能够生成高质量的文本嵌入，使得相似性检索的准确性和相关性大幅提升，确保用户能够获得更为精准的回答。

通过以上数据构建与向量化处理的方法，本项目能够有效地实现问答系统的高效检索与响应，提升用户体验。

* **功能整合**（进阶版RAG必填）：  介绍进阶的语音功能、Agent功能、多模态等功能的整合策略与实现方法。

#### 实施步骤：

* 环境搭建（必写）： 描述开发环境的搭建过程，包括必要的软件、库的安装与配置。

conda create --name ai\_endpoint python=3.8

conda activate ai\_endpoint

pip install langchain-nvidia-ai-endpoints

pip install jupyterlab

pip install langchain\_core langchain matplotlib

pip install numpy openai

pip install faiss-cpu==1.7.2

* 代码实现（必写）： 列出关键代码的实现步骤，可附上关键代码截图或代码块。

A screen shot of a computer code

Description automatically generated

* 测试与调优： 描述测试过程，包括测试用例的设计、执行及性能调优。
* 集成与部署： 说明各模块集成方法及最终部署到实际运行环境的步骤。

#### 项目成果与展示：

* 应用场景展示(必写)： 描述对话机器人的具体应用场景，如客户服务、教育辅导等。

**社交媒体互动**：该系统可以集成到社交媒体平台中，与用户进行互动，提供有趣的回答和建议。

例如，在论坛或者视频留言区，自动以弱智吧风格回答评论。

* 功能演示（必写）： 列出并展示实现的主要功能，附上UI页面截图，直观展示项目成果。

A screenshot of a computer program

Description automatically generated

#### 问题与解决方案：

* 问题分析： 详细描述在项目实施过程中遇到的主要问题。
* 解决措施： 阐述针对每个问题采取的具体解决措施及心路历程，体现问题解决能力。

#### 项目总结与展望：

* 项目评估： 对项目的整体表现进行客观评估，总结成功点和存在的不足。
* 未来方向： 基于项目经验，提出未来可能的改进方向和发展规划。

#### 附件与参考资料

[列出项目报告中引用的所有附件和参考资料。]

<https://blog.csdn.net/kunhe0512/article/details/140910139>

<https://build.nvidia.com/>

<https://tieba.baidu.com/f?kw=%E5%BC%B1%E6%99%BA&ie=utf-8>

<https://news.sciencenet.cn/htmlnews/2024/4/521058.shtm>