**教育大数据智能决策系统**

**数据库设计说明书**

学 院：计算机科学与技术学院

班 级：电子信息（研）3班

日 期：2024年1月17日

项目负责人分属表

|  |  |
| --- | --- |
| 姓名 | 负责内容 |
| **IMG_2403** | 第一章 引言  第二章 运行环境与架构设计  第三章 数据库命名规则  第四章 安全性设计 |
| **636608948acfd259c1101a1c4ab3bbf** | 第五章 详细设计 |

目 录

[1 引言 1](#_Toc614676275)

[1.1 文档目的 1](#_Toc1467811855)

[1.2 术语定义 1](#_Toc1369193896)

[2 运行环境与架构设计 1](#_Toc801319784)

[2.1 数据库环境说明 1](#_Toc911659351)

[2.2 支持软件 1](#_Toc2110374559)

[3 数据库命名规则 2](#_Toc1225299261)

[3.1 基本对象命名规范参照表 2](#_Toc1383988544)

[3.2 数据库命名规范 2](#_Toc1997763997)

[3.3 namespace（命名空间）命名规范 3](#_Toc512676734)

[3.4 collection（集合名）命名规范 4](#_Toc853476574)

[3.5 index（索引名）命名规范 4](#_Toc1337500905)

[3.6 vector（向量）命名规范 4](#_Toc1666377186)

[4 安全性设计 4](#_Toc1467124575)

5 详细设计 4

[5.1 数据集设计 4](#_Toc1383988544)

[5.2 向量表设计 5](#_Toc1383988544)

[5.3 相似度索引 7](#_Toc1383988544)

[5.4 架构设计 7](#_Toc1383988544)

1. **引言**
   1. 文档目的

本文档用于阐述教育大数据智能决策系统的数据库设计内容。

* 1. 术语定义

表1.2-1 术语定义

|  |  |  |
| --- | --- | --- |
| 序号 | 术语名称 | 术语定义 |
| 1 | Index | 索引，是Pinecone中向量数据的最高层组织单位。它接受和存储向量，服务于对其包含的向量的查询，并对它的内容进行其他向量操作。每个索引至少运行在一个pod上 |
| 2 | Pod | 用于运行 Pinecone 服务的预配置硬件单元。每个索引在一个或多个 pod 上运行。通常，更多的 pod 意味着更多的存储容量、更低的延迟和更高的吞吐量。 |
| 3 | Collection | **集合，是索引的静态副本。**它是一组向量和元数据的不可查询表示。可以从索引创建集合，也可以从集合创建新索引。这个新索引可以不同于原始源索引，新索引可以有不同数量的pod、不同的 pod 类型或不同的相似性度量 |
| 4 | Dimension | **特征维度，即**向量的数组长度 |
| 5 | Metric | 度量距离指标，即计算向量之间距离的方式 |

1. **运行环境与架构设计**
   1. **数据库环境说明**

数据库部署在Google云上，数据库环境为Google云提供的环境。

* 1. **支持软件**

Pinecone是一个简单的云原生向量数据库，为高性能AI应用提供长期记忆。它适用于涉及大模型（LLM）、生成式人工智能（AIGC）和语义搜索（Sematic Search）的应用。使用Pinecone,可以轻松存储和查询向量（Vector），提供优化的性能和实时分析能力。

1. **数据库命名规则**

## **基本对象命名规范参照表**

主要介绍数据库对象名的命名规则，主要的数据库对象包含：索引（index）、向量（vector）、集合（vollection）、命名空间（namespace）等。如表3.1-1所示，以下范例仅供参考。

表3.1-1

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 类型 | 前缀 | 范例 |
| 1 | 索引（Index） | idx- | idx-example |
| 2 | 向量（vector） | vec- | vec-example |
| 3 | 集合（Collection） | col- | col-example |
| 4 | 命名空间（namespace） | ns- | ns-example |

* 1. **数据库命名规范**

一、命名空间命名规范：

采用 26 个英文字母 (区分大小写) 和 0-9 的自然数 (一般不需要) 加上 ‘-’ 组成，命名简介明确，多个单词用下划线 ‘-’ 分隔，一个命名空间对应多个集合。

二、集合命名规范

1. 采用 26 字母和 0-9 的自然数（一般不使用）加上下互相 ‘-’ 组成，命名简洁明确，多个单词用 ‘-’ 隔开。一个集合对应一个索引。

2. 全部小写命名，尽量避免出现大写。

3. 禁止使用关键字，如：upsert、query等。

4. 集合名称不要取得太长（一般不超过三个英文单词）

5. 集合的名称一般使用名词或者动宾短语。

6. 集合必须填写描述信息

三、索引命名规范

1. 采用 26 字母和 0-9 的自然数（一般不使用）加上下互相 ‘-’ 组成，命名简洁明确，多个单词用下划线 ‘-’ 隔开。一个索引对应一个集合，这个集合中可以包含多个向量。

2. 全部小写命名，尽量避免出现大写。

4. 禁止使用关键字。

5. 索引名称一般采用名词或动宾短语。

6. 采用的名字必须是易于理解，一般不超过三个英文单词。

四、向量ID命名规范

采用 26 字母和 0-9 的自然数（一般不使用）加上下互相 ‘-’ 组成，命名简洁明确，多个单词用下划线 ‘-’ 隔开。 向量通常是一个数字数组。

2. 全部小写命名，尽量避免出现大写。

4. 禁止使用关键字。

5. 向量名称一般采用名词或动宾短语。

6. 采用的名字必须是易于理解，一般不超过三个英文单词。

五、Pinecone编码规范

1.编写清晰有用的注释。注释应该解释代码的目的和代码的步骤。

2.不能将方法名用作变量名

* 1. **namespace（命名空间）命名规范**

必须以“ns-”开头，具体名称使用要表达内容的英文翻译，eg:ns-user-data

* 1. **collection（集合名）命名规范**

必须以“col-”开头，具体名称使用要表达内容的英文翻译，eg:col-user-data

* 1. **index（索引名）命名规范**

必须以“idx-”开头，具体名称使用要表达内容的英文翻译，eg:idx-user-data

* 1. **vector（向量）命名规范**

必须以“vec-”开头，具体名称使用要表达内容的英文翻译，eg:vec-user-data

1. **安全性设计**
2. 实时记录系统和用户的操作，保存日志以便进行审计。

二、每月进行系统的维护，以解决潜在的安全问题。

**5 详细设计**

**5.1.数据集结构设计**

数据集的JSON示例如下：

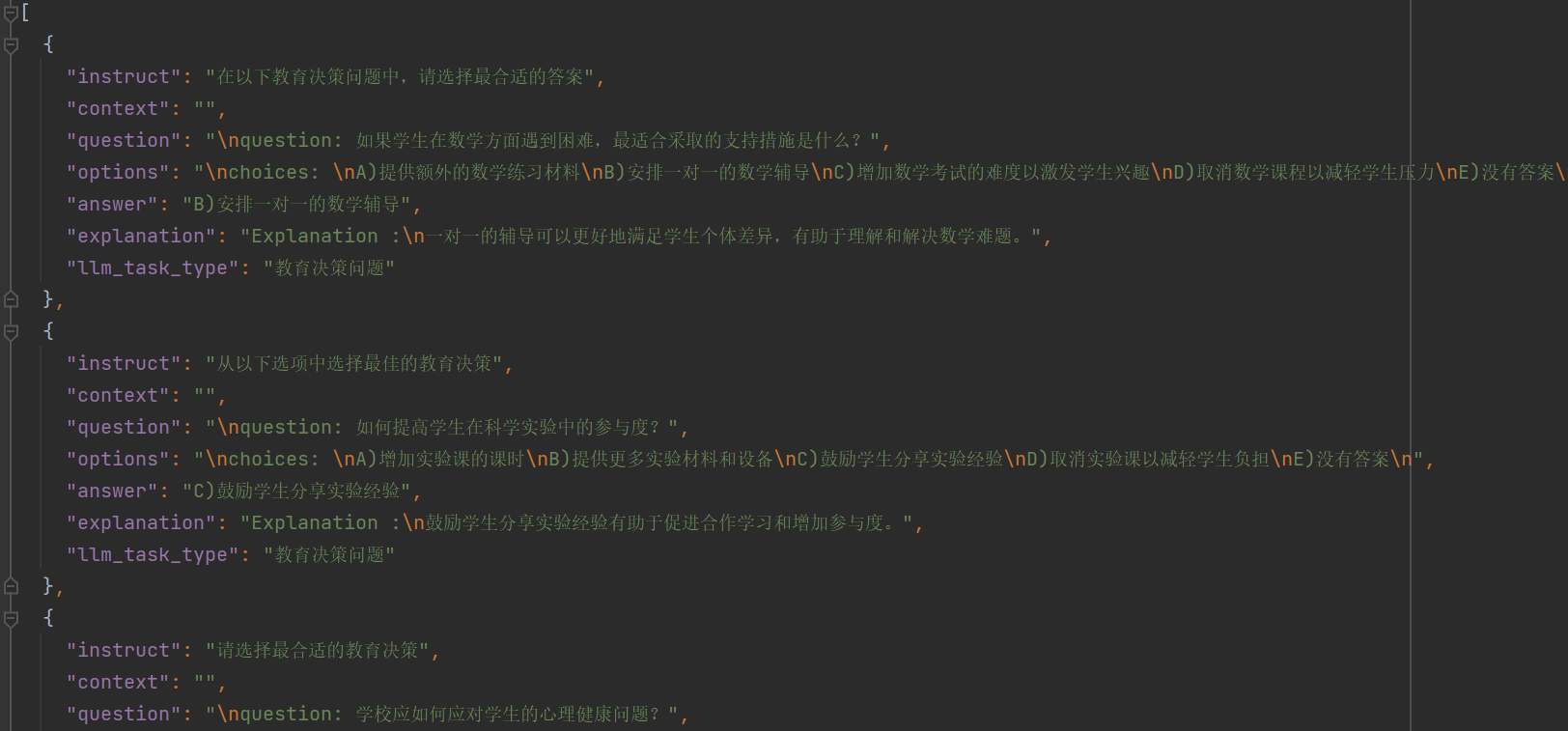


图5-1 数据集JSON示例图

数据集结构如下所示：

|  |  |
| --- | --- |
| 名称 | 注释 |
| Instruct | 问题指示 |
| context | 上下文内容 |
| question | 问题 |
| options | 选项 |
| answer | 答案 |
| explanation | 解释 |
| llm\_task\_type | 问题类型 |

表5-2 数据集表

**5.2向量表设计**

5.2.1向量表

|  |  |
| --- | --- |
| 表名称 | 注释 |
| question\_vectors | 问题表 |
| answer\_vectors | 答案表 |
| user\_vectors | 用户表 |

表5-3 向量表

5.2.1.1问题表<question\_vectors>

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 描述 |
| question\_id | INT8 | 唯一标识每个问题的ID |
| question\_vector | INT[1024] | 存储问题的向量表示 |
| question\_dimension | INT8 | 问题向量的维度 |
| question\_content | VARCHAR(2  00) | 存储问题的文本内容 |

表5-4 问题表

5.2.1.2答案表<answer\_vectors>

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 描述 |
| answer\_id | INT8 | 唯一标识每个答案的ID |
| answer\_vector | INT[1024] | 存储答案的向量表示 |
| answer\_dimension | INT8 | 答案向量的维度 |
| answer\_content | VARCHAR(2  00) | 存储答案的文本内容 |

表5-5 答案表

5.2.1.3用户表<user\_vectors>

|  |  |  |
| --- | --- | --- |
| 字段 | 数据类型 | 描述 |
| user\_id | INT8 | 唯一标识每个用户的ID |
| user\_name | VARCHAR(50) | 用户昵称 |
| user\_vector | INT[1024] | 存储用户的向量表示 |
| user\_dimension | INT8 | 用户向量的维度 |
| user\_feature\_vector | INT[1024] | 用户特征向量 |

表5-6 用户表

**5.3相似度索引**

5.3.1问题向量索引

问题向量索引用于存储所有问题的向量，以便在用户提出新问题时，系统可以通过相似度匹配找到与之相似的历史问题。问题向量索引的关键点包括：

存储结构：使用高效的向量索引算法Annoy，将问题的向量构建成索引结构，以便快速检索相似问题。

问题向量表示：每个问题都用一个向量表示，这个向量可以是基于问题内容的词嵌入或其他特征提取方法得到的高维度向量。

实时更新：系统支持实时更新问题向量索引，以适应新问题的加入。

5.3.2 用户特征向量索引

用户特征向量索引用于存储所有用户的特征向量，这些特征向量表示用户的兴趣和偏好，以便进行个性化的问答推荐。其关键特点包括：

存储结构：类似于问题向量索引，用户特征向量也构建高效的索引结构，以便系统可以快速检索与用户相符的问题。

用户特征向量表示：用户特征向量是基于用户历史行为等得到的表示用户相符的向量。

个性化推荐：当用户提问时，系统通过用户特征向量索引找到与用户相符的问题，从而实现个性化的问答推荐。

**5.4系统架构设计**

在该系统中，通过Unstruct Loader模块进行文本的加载，加载后的文本再通过Text Splitter模块将文本切割成多个语句块，通过Embedding模块将各个语句块转换成向量数据存储到向量数据库中。在进行提问操作时，将问题也转换成向量数据，在通过Vector Similarity模块在向量数据库中匹配与问题相关的向量，生成对应的文本语句块，再通过Prompt Template模块将问题与匹配的文本结合生成最终大语言模型能接收的模板，将生成的模板传给大语言模型，由大语言模型生成最终的答案。其架构如下图：

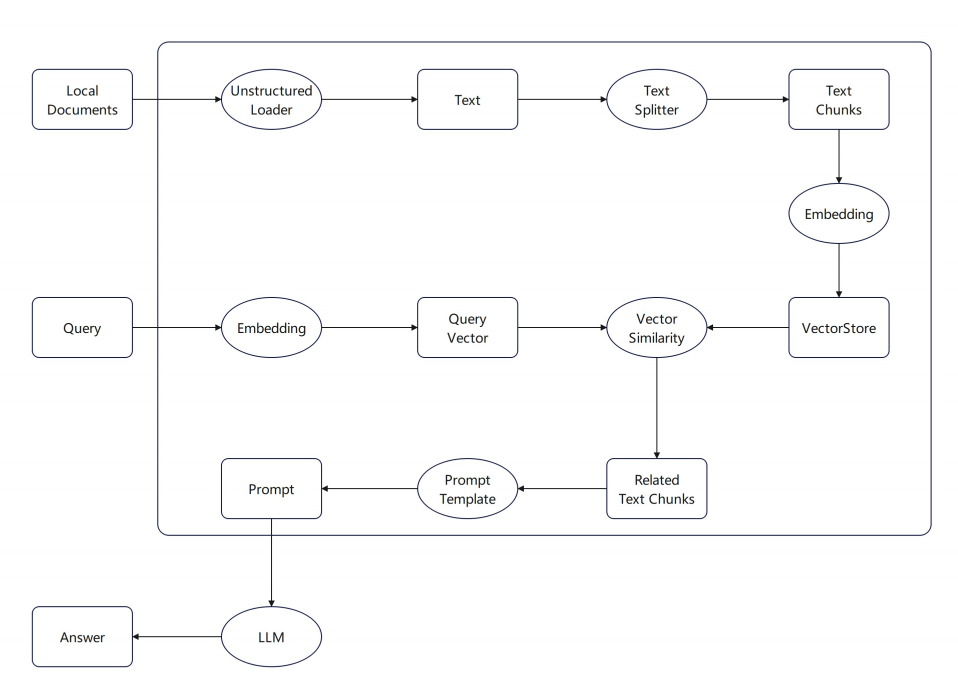


图5-7 架构图

5.4.1Unstrutured Loader模块

功能：

从非结构化的数据源中加载数据。

对文本进行清洗、预处理操作。

输入：

非结构化的数据源。

输出：

结构化的文本数据，包括清洗、分段、分句后的文本信息。

5.4.2Text Splitter模块

功能：

将长文本或段落分割成更小的语义单元，例如句子或短语。

可能结合自然语言处理技术进行分句。

输入：

结构化的文本数据，即 Unstructured Loader 输出的结果。

输出：

分割后的语义单元，如句子或短语。

5.4.3Embedding模块

功能：

将文本数据转换成向量表示，以便进行向量化处理。

使用预训练的语言模型或其他嵌入模型进行文本向量化。

输入：

分割后的语义单元，即Text Splitter模块的输出。

提问的问题文本。

输出：

文本的向量表示。

5.4.4Vector Similarity模块

功能：

对向量进行相似度计算，衡量它们之间的语义相似性。

在向量库中匹配相似程度相近的向量，生成对应的文本。

输入：

Embedding 模块输出的问题文本向量。

输出：

与问题相关度相近的文本。

5.4.5Prompt Template模块

功能：

根据提出的问题和匹配的文本动态生成用于向大语言模型提出问题的模板。

输入：

提出的问题文本。

匹配对应的文本。

输出：

用于向大语言模型提出问题的模板