Project Number: 14 项目编号：14

Project Title: AI Comment Moderation - RAG and Classification modelling

项目标题：人工智能评论修改--RAG 和分类建模

Project Clients: Nathaniel Lewis

项目客户： 纳撒尼尔-刘易斯

Project specializations: Software Development;Computer Science and Algorithms;Artificial Intelligence (Machine/Deep Learning, NLP);Web Application Development;

项目专长： 软件开发；计算机科学与算法；人工智能（机器/深度学习、NLP）；网络应用程序开发；

Number of groups: 4 groups

组数：4 组

Main contact: Nathaniel Lewis

主要联系人：Nathaniel Lewis

Background: 背景介绍

To support review and moderation of student comment data, staff currently run a manual checking of thousands of student comments. AI and LLM tools are potentially well suited to moderation/checking of qualitative comment data and making recommendations for reviewers actions to take, reducing manual tasks and standardising approaches.

为了支持对学生评论数据的审查和修改，工作人员目前要对数千条学生评论进行人工检查。人工智能和LLM工具可能非常适合定性评论数据的审核/检查，并为审核人员的行动提出建议，从而减少人工任务并使方法标准化。

Requirements and Scope: 要求和范围：

Develop a proof-of-concept (PoC) system that moderates student feedback comments using Retrieval-Augmented Generation (RAG) instead of custom training.

开发一个概念验证（PoC）系统，利用检索-增强生成（RAG）而不是定制培训来调节学生的反馈意见。

The system will: 该系统将
-Ingest and store student feedback comments in a structured format.

-以结构化的格式收集和存储学生的反馈意见。

-Retrieve past similar feedback examples from a vector database

-从矢量数据库中检索过去类似的反馈示例

-Use an LLM to classify new comments based on retrieved examples.

-使用 LLM 根据检索到的示例对新注释进行分类。

-Determine if human review is needed based on classification confidence.

-根据分类可信度确定是否需要人工审核。

-Provide an interface for manual review of flagged comments, and versioning, and saving of data.

-提供一个界面，以便对标记的评论进行人工审核，并对数据进行版本控制和保存。

-Provide recommendation for alternative approaches to supplement RAG approach with LLM pre-training

-通过 LLM 预先培训，为补充 RAG 方法的替代方法提供建议

-Bonus points for Agent integration and LLM tool use for structured outputs

-Agent集成和LLM工具的使用为结构化输出加分

Key Constraints: 关键制约因素：
-No custom model training-only RAG-based techniques.

-没有定制模型训练，只有基于 RAG 的技术。

-Uses dummy data with a well-defined schema.

-使用具有明确模式的虚拟数据。

Required Knowledge and skills:

所需知识和技能

Data Ingestion, Storage, Management: The system will accept student feedback comments (dummy dataset) from ‘Best Aspects’ and ‘Areas for Improvement’ fields, storing them in a structured format for processing. IT should also allow versioning and export of pre and post datasets to file (csv, tsv, excel).

数据输入、存储和管理：系统将接受 "最佳方面 "和 "有待改进的方面 "字段中的学 生反馈意见（虚拟数据集），并将其存储为结构化格式以便处理。信息技术还应允许版本控制和将前后数据集导出到文件（csv、tsv、excel）中。

Schema Definition: A standardised input/output JSON schema will be used to ensure consistency across all components, making it easier to pass data between modules.

模式定义：将使用标准化的输入/输出 JSON 模式，以确保所有组件的一致性，使模块之间的数据传递更容易。

Vector Database for Retrieval: Feedback comments will be embedded using a pre-trained model and stored in a vector database (FAISS, Weaviate, or Pinecone) to enable similaritybased retrieval.

用于检索的矢量数据库：反馈意见将使用预先训练好的模型进行嵌入，并存储在矢量数据库（FAISS、Weaviate 或 Pinecone）中，以便进行基于相似性的检索。

Retrieval Engine: When a new comment is received, the system will retrieve top- N most similar past comments to provide context for classification and moderation.

检索引擎：当收到一条新评论时，系统将检索过去最相似的前 N 条评论，为分类和审核提供背景信息。

LLM-Based Classification: A pre-trained LLM (GPT-4, Claude, or Hugging Face model) will analyze the new comment in relation to retrieved examples and assign it a moderation category (e.g., Safe, Needs Review, Violent, Swearing, etc.).

基于LLM的分类：预先训练好的LLM（GPT-4、克劳德或拥抱脸模型）将根据检索到的示例对新评论进行分析，并为其指定一个审核类别（例如，安全、需要审核、暴力、脏话等）。

Moderation Confidence Scoring: The system will generate a confidence score for each classification decision. If confidence is high, the comment will be automatically categorized; if confidence is low, it will be flagged for human review.

审核置信度评分：系统将为每个分类决定生成置信度评分。如果置信度高，评论将被自动分类；如果置信度低，评论将被标记为需要人工审核。

Human Review Interface: A simple web-based UI will allow moderators to view flagged comments, check retrieved examples, and confirm, override, or edit AI-generated classifications.

人工审核界面：一个简单的网络用户界面将允许版主查看标记的评论，检查检索到的示例，并确认、覆盖或编辑人工智能生成的分类。

API Integration & Automation: The entire pipeline will be modular and API-driven, allowing seamless integration between data ingestion, retrieval, classification, and human review.

应用程序接口（API）集成与自动化：整个管道将采用模块化和应用程序接口（API）驱动，实现数据摄取、检索、分类和人工审核之间的无缝集成。

Expected outcomes/deliverables:

预期成果/交付成果：

-Source Code (GitHub repo with documentation)

-源代码（包含文档的 GitHub 代码库）

-Structured Dummy Dataset (JSON/CSV format)

-结构化虚拟数据集（JSON/CSV 格式）

-Vector Database with Sample Comments (FAISS/Weaviate setup)

-带样本注释的矢量数据库（FAISS/Weaviate 设置）

-Basic API for Retrieval & Classification

-用于检索和分类的基本应用程序接口

-Simple UI for Moderation Review

-审核的简单用户界面

-Project Documentation (setup guide, architecture)

-项目文档（设置指南、架构）

-Final Report & Presentation (PoC demonstration)

-最终报告和演示（PoC 演示）