Project Number: 14

Project Title: AI Comment Moderation - RAG and Classification modelling

Project Clients: Nathaniel Lewis

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

Number of groups: 4 groups

Main contact: Nathaniel Lewis

Background:

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.

Requirements and Scope:

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

The system will:

-Ingest and store student feedback comments in a structured format.

-Retrieve past similar feedback examples from a vector database

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

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

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

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

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

Key Constraints:

-No custom model training—only RAG-based techniques.

-Uses dummy data with a well-defined schema.

Required Knowledge and skills:

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).

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.

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 similarity-based retrieval.

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.

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.).

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.

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.

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

Expected outcomes/deliverables:

-Source Code (GitHub repo with documentation)

-Structured Dummy Dataset (JSON/CSV format)

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

-Basic API for Retrieval & Classification

-Simple UI for Moderation Review

-Project Documentation (setup guide, architecture)

-Final Report & Presentation (PoC demonstration)