**Artificial Assessment Intelligence – For Educators (AAIE) Report**

1. **Team Information**

* **Team**: Solution Architecture Team in Model Training
* **Lead**: VAN HIEU NGUYEN
* **Member**: UMAR KHAYAM
* **Product Owner**: ARNAVE AHUJA
* **Scrum:** KHUSHI CHOUBEY

1. **Create Solution Architecture (Week 1 and Week 3)**

**Objective**: Create Solution Architecture, including data flow diagram for AI-generated content detection and feedback generation, draw the system architecture and add data flow annotations.

**Use case:**

1. **AI Detection:** Automatically label student writing as AI or human-generated with confidence explanation (show as percentage)
2. **Feedback Generation:** Automatically generate personalized feedback based on rubric and student response (as prompt)

**Solution Architecture:**

1. **AI Detection:**

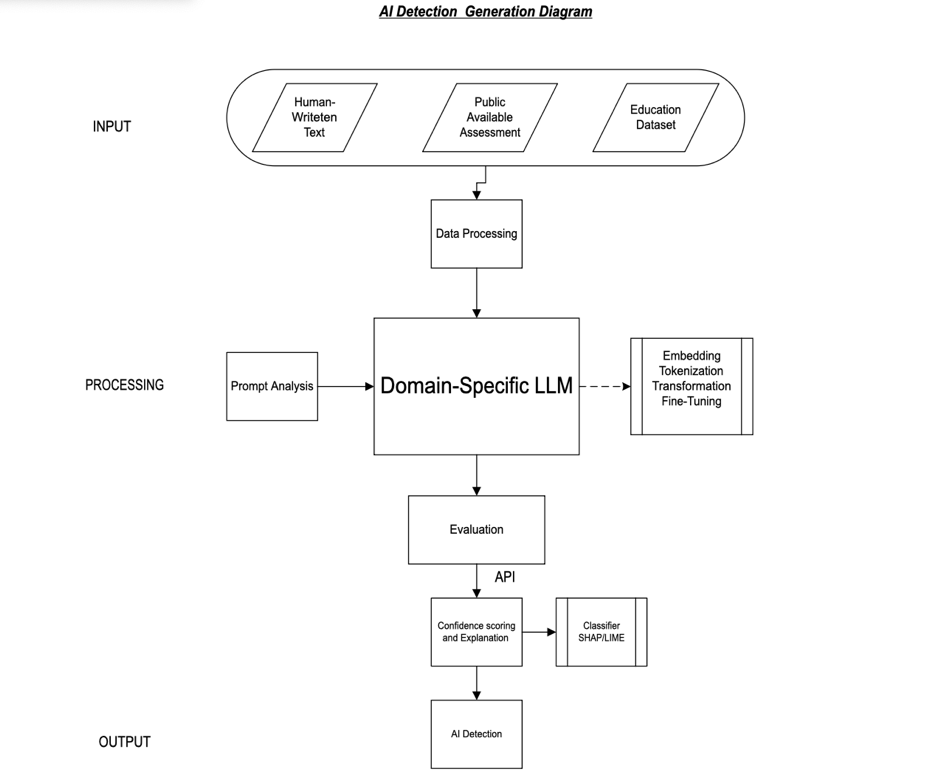
* Input: Student text
* Modules: Pre-processing → Embedding → Classifier → Confidence/ Explanation
* Output: AI/Human label + explanation

1. **Feedback Generation:**

* Input: Student response + rubric
* Modules: Text Cleaning → Feature Extraction → Rubric Matcher / student response → LLM Generator
* Output: Natural language feedback + rubric tags

**Instruction:**

1. Create data/model flow diagram for AI-generated content detection:



* **Data Flow:**
* Input: string/JSON → Output: JSON
* Embeddings: vector shape (768,)
* AI Detection: AI or Human
* **Example:**
* Input: prompt engineering
* Expected Output:

{

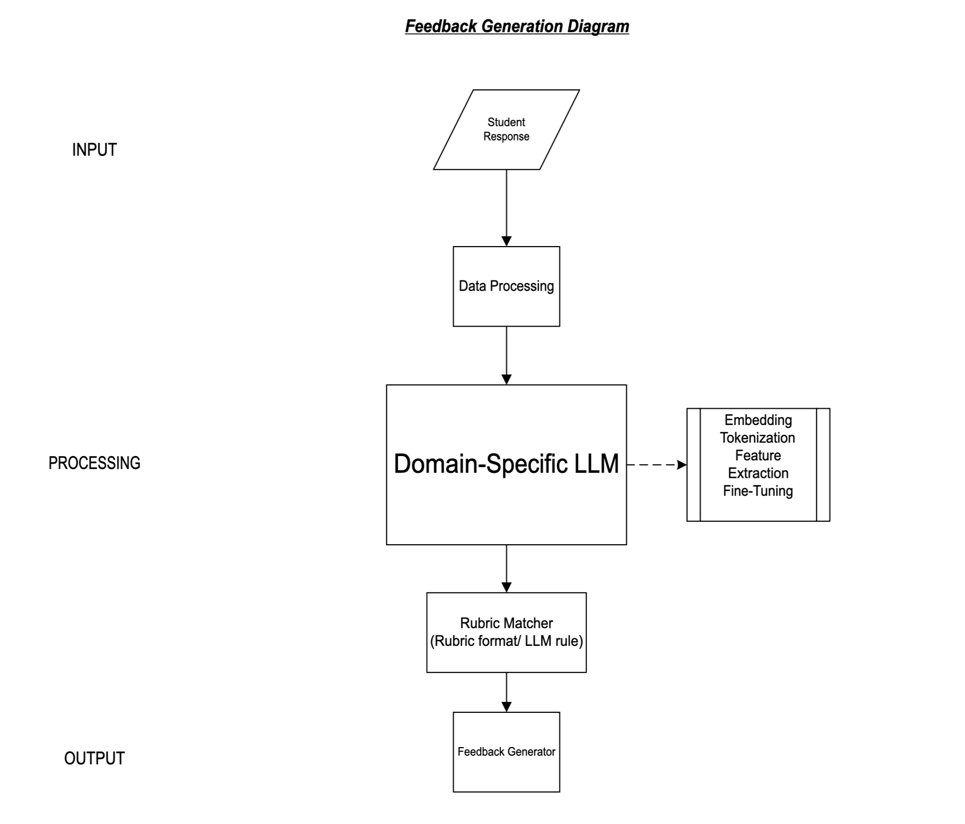
"label": "AI",

"confidence": 0.72,

"explanation": "Unusual phrase repetition similar to GPT"

}

1. Design flow diagram for rubric-aligned feedback generation:



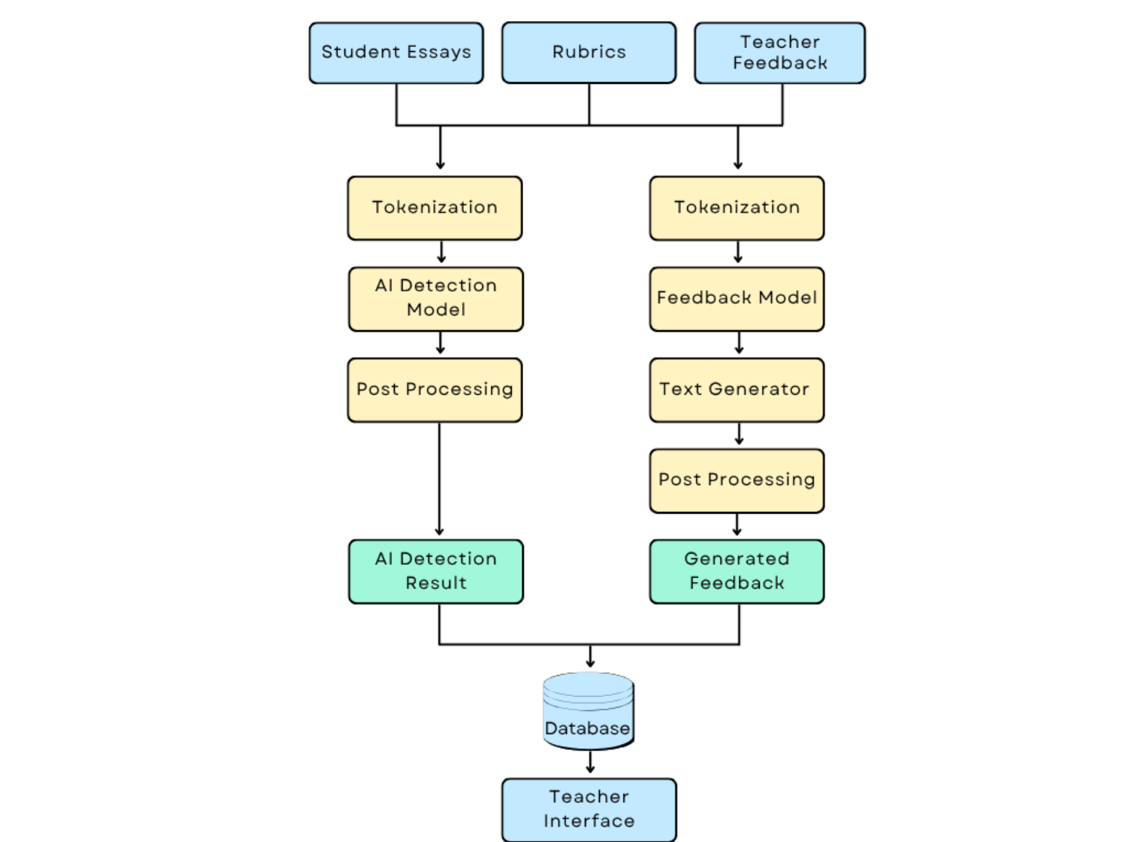
* **Data Flow:**
* Input: string/JSON → Output: JSON
* Embeddings: vector shape (768,)
* Output: Text, String, or JSON
* **Example:**
* Input: Student Response
* Expected Output:

{

"feedback": "Here is feedback from your tutor or unit chair."

}

1. Draw system-level architecture (block diagram): UMAR KHAYAM



* In Database, it will contain:
* Vector DB: Embeddings
* SQL/NoSQL: Rubrics,
* Logs- S3/GCS: Model Checkpoints

**Module Descriptions:**

1. Pre-processing: clean and normalize input text. Using: NLTK, spacy.
2. Tokenizer: tokenize for embedding/model input . Using: BPE, SentencePiece.
3. Embedding: encode text to vector. UsingBERT, SentenceTransformer.
4. Classifier: classify AI vs Human. Using: MLP, logistic regression.
5. Explanation: string explanation. Using: SHAP, LIME.
6. Feature Extractor: extract NLP features. Using: custom NLP.
7. Rubric Matcher: align response to rubric. Using: Rule-based / LLM prompts.
8. Feedback Generator: produce rubric-aligned feedback. Using: GPT prompt, LLM.
9. API Gateway: route frontend/backend requests . Using: FastAPI, Flask.