Research Plan

Intelligent Chatbot for Manufacturing Equipment Safety

The primary objective of this project is to design and implement an LLM-based chatbot capable of processing multimodal inputs (both text and images) to deliver accurate answers grounded in authoritative sources about manufacturing equipment safety. We expect the final system to improve hazard recognition and overall user experience and engagement with safety protocols. This research plan outlines the key phases and activities necessary to achieve that goal.

1. Data Collection

- Compile equipment manuals for all machinery in Miami University's Advanced Manufacturing Hub.
- Compile OSHA, NIOSH, and/or BWC guidelines.
- Take photos of manufacturing equipment at Miami University's Advanced Manufacturing Hub from multiple angles and lighting conditions. Ensure images show each machine's safety features, such as warning labels and hazard zones, and include examples of both correct and incorrect operation.

2. Data Preparation and Storage

- Convert textual data to a more structured input format (e.g., markdown or HTML).
- Create an image database. Each image must be supplemented with relevant metadata to enhance its utility. The metadata includes key details, such as equipment specifications, safety-critical features, and known hazards. Suggestion: Use MongoDB, store each image in GridFS using a proper linking key, and use a schema similar to the one in the appendix.

3. Development

We will compare and develop the following RAG and non-RAG methods:

3.1 RAG Variants

- "Vanilla" RAG as the baseline.
- DPR (Dense Passage Retrieval): Use LangChain and DPRRetriever.
- BM25 + Re-ranking: Use LangChain with BM25Retriever for the first stage and a ContextualCompressionRetriever with a CrossEncoderReranker for the second stage.
- Graph-based RAG (developed by Fadel): Graph-based chunk retrieval via cosine similarity on OpenAI embeddings. Prompt construction with top-k chunks and gpt-4o for generation.

.

3.2 RAG with Vision Guidance:

- Direct image embedding and comparison using CLIP or BLIP.
- Image classification to select relevant documents before RAG.
- From Fadel: Possibly use this instead (https://huggingface.co/ds4sd/SmolDocling-256M-preview) for the documents [RAG].

3.3 Non-RAG

- Context stuffing: Load full or partial documents directly into the prompt based on heuristic filters or image classification.

4. Evaluation

4.1 Quantitative Metrics

- Accuracy: Measure overlap with gold-standard answers (e.g., Exact Match, F1, ROUGE). This requires the creation of a curated question-answer benchmark set from the documents by experts.
- Retrieval precision/recall: Evaluate document chunk relevance.
- Latency & cost: Compare runtime and token/call cost per approach.

4.2 Qualitative Metrics

- Interviews with experts to explore strengths and weaknesses.

- Interviews with users/student to measure usefulness (based on variants of the Technology Acceptance Model?)

4.3 Multimodal Evaluation (?)

- Classifier accuracy for vision-guided retrieval (?)

5. Deployment

Develop a Streamlit application (or an equivalent framework) featuring a RAG selector (?) and visual upload functionality, complemented by a backend that provides persistent graph and embedding storage.

Task Allocation:

- Arthur, Fadel, and Jay:
  - Supervise undergraduate students, including offering technical support, scheduling and attending weekly meetings, project management, etc.
  - Write academic paper
- ??? (data collection)
  - Collect manuals
  - Take pictures of machines
- ??? (evaluation)
  - Create a set of question-answer benchmark based on manuals
  - Formulate interview questions for experts
  - Formulate survey questions for students/users based on TAM
  - Recruit experts and students/users
- Undergraduate RAs (x3)
  - Perform data preparation, development, solution evaluation for the created benchmarks, and deployment.
  - Deliverable #1: Chatbot code on GitHub or Hugging Face.
  - Deliverable #2: Technical report, which will serve as the basis of a future academic paper.

Appendix: Sample Schema


```
{
 "image_id": "uuid ",
 "equipment_name": "Laser Cutter",
 "location": "Room A12",
 "safety_critical_features": ["Laser shield"],
 "operating_condition": "improper",
 "known_hazards": ["Eye damage", "Fire risk"],
 "bounding_boxes": "bounding_boxes": [
        {
          "x": 100,
          "y": 200,
          "width": 50,
          "height": 30,
          "label": "Label on device"
        },
        {
          "x": 300,
          "y": 120,
          "width": 40,
          "height": 40,
          "label": "Hazardous part"
        }
 ],
 "image_file_id": "gridfs_id_or_filename"
```

```
}
```