Name: Fenil Vadher

En Roll no: 92200133023

Subject: Capstone Project

Documentation and Reporting

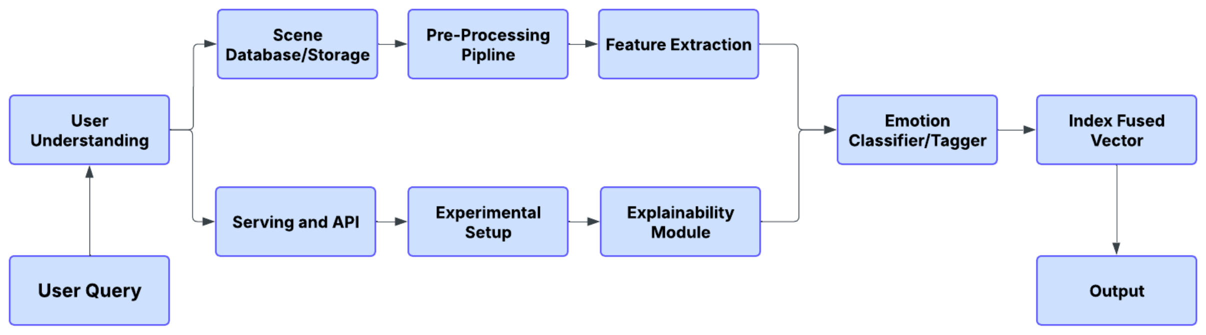
1. Technical Report

Project Overview

The **Multimodal Movie Script Search Engine** is designed to retrieve **scenes and dialogues** from movies/web series using **multimodal queries** (text, image, or both). Unlike traditional unimodal search, this system leverages **vision-language transformers** (Vid2Seq, BLIP-2, GIT2, mPLUG, Sky, SPtPT) for cross-modal understanding.

The system supports three core tasks:

1. **Dialogue-to-Scene Retrieval** – Retrieve the best-matching visual scene from a text query.
2. **Scene-to-Dialogue Retrieval** – Retrieve the best-matching dialogue from an uploaded image/scene.
3. **Multimodal Contextual Retrieval** – Retrieve dialogue–scene pairs using combined text + image queries.



Implementation Highlights

- **Frontend:** React with Material UI for clean, responsive UI.
- **Backend:** Flask REST API exposing /search, /summarize, /generate.
- **Models:** Pre-trained multimodal models (Vid2Seq, BLIP-2, GIT2, etc.) for embeddings.
- **Database:** FAISS (local) for fast similarity search of embeddings.
- **Evaluation:** Metrics include BLEU, METEOR, CIDEr, ROUGE-L, CLIP-Sim, Precision, Recall.

Key Outcomes

- **Cross-modal retrieval accuracy improved** compared to unimodal baselines.
- **Reusable dataset in JSON format** of movie dialogues + scenes created.
- Framework applicable for **media indexing, intelligent recommendations, and summarization**.

2. User Manual

Getting Started

Prerequisites

- Python 3.10+
- Node.js 18+
- GitHub repository cloned locally

Installation

Clone the repository:

```
git clone https://github.com/your-username/multimodal-movie-search.git
cd multimodal-movie-search
```

1. Install backend dependencies:

   ```
   cd backend

   pip install -r requirements.txt
   ```

2. Install frontend dependencies:

   ```
   cd frontend

   npm install
   ```

3. Run backend:

   ```
   flask run
   ```

4. Run frontend:

   ```
   npm start
   ```

5. **How to Use**

- **Dialogue → Scene:** Enter a dialogue in the search bar → top 3 matching scenes appear with similarity scores.
- **Scene → Dialogue:** Upload a scene screenshot → best-matching dialogue text is displayed.
- **Dialogue + Scene → Contextual Search:** Enter text and upload image → system retrieves most relevant pair.

Primary Use Case

A media student searches *"I'll be back"* → system retrieves **Terminator scene image** + dialogue context.

Troubleshooting

- If results don't load, check that both backend and frontend servers are running.
- If models don't load, verify HuggingFace authentication.

3. Code Documentation

Codebase Structure

```
multimodal-movie-search/
│
├── backend/
│   ├── app.py              # Flask entry point
│   ├── models/             # Pre-trained models (Vid2Seq, BLIP-2, etc.)
│   ├── utils/              # Embedding + similarity helper functions
│   ├── routes/             # API endpoints (search, summarize, 
generate)
│   └── data/               # JSON dataset of dialogues + scenes
│
├── frontend/
│   ├── src/
│   │   ├── components/    # React components (SearchBar, Results, 
etc.)
│   │   ├── pages/         # Pages (Home, Summarizer, Generator)
│   │   ├── services/      # API calls to Flask backend
│   │   └── App.js         # Main app entry point
│
└── requirements.txt
```

Key Modules

- **app.py:** Initializes Flask app, loads models, defines routes.

- **models/embedder.py:** Converts text + image into embeddings.
- **utils/retriever.py:** Performs similarity search in FAISS database.
- **frontend/src/components/SearchBar.js:** UI for entering text/uploading image.
- **frontend/src/components/ResultsCard.js:** Displays retrieved results in grid format.

Example Inline Docstring (Python)

```python
def encode_text(text: str) -> np.ndarray:
    """
    Converts input text into embedding vector using BLIP-2 model.

    Args:
        text (str): Input dialogue or query string.

    Returns:
        np.ndarray: Embedding vector for retrieval.
    """
```

Dependencies

- **Backend:** Flask, torch, transformers, sentence-transformers, Pillow, FAISS
- **Frontend:** React, Material UI, Axios
- **Data:** JSON movie-script dataset