Personalized Learning Assistant Project Report

Gowreesh Gunupati, Katyaini Raj, Priyanka Gujar

Introduction

The primary objective of this project is to create an assisted learning system powered by large language models (LLMs) that simplifies the process of generating study materials. The platform allows users to upload text or PDF files, which are then processed to generate contextually relevant questions and summaries. This tool aims to enhance the learning experience by providing tailored educational content, fostering better comprehension and engagement.

The methodology utilizes two specialized models: GEMMA, a sequence-to-sequence model optimized for summarization tasks, and LLAMA 3.2, a decoder model designed for generating high-quality questions. The workflow integrates pre-processing steps, task-specific operations, and user input via an intuitive Streamlit-based interface. The attached flowchart illustrates the system architecture, highlighting the seamless interaction between components such as input processing, task selection, and model execution.

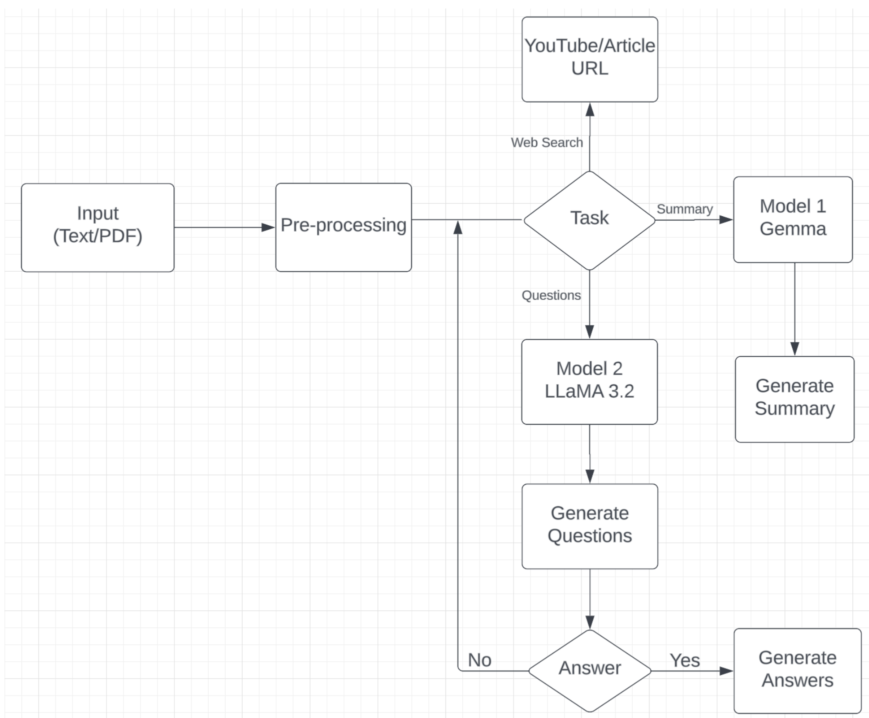

Figure 1: Application workflow

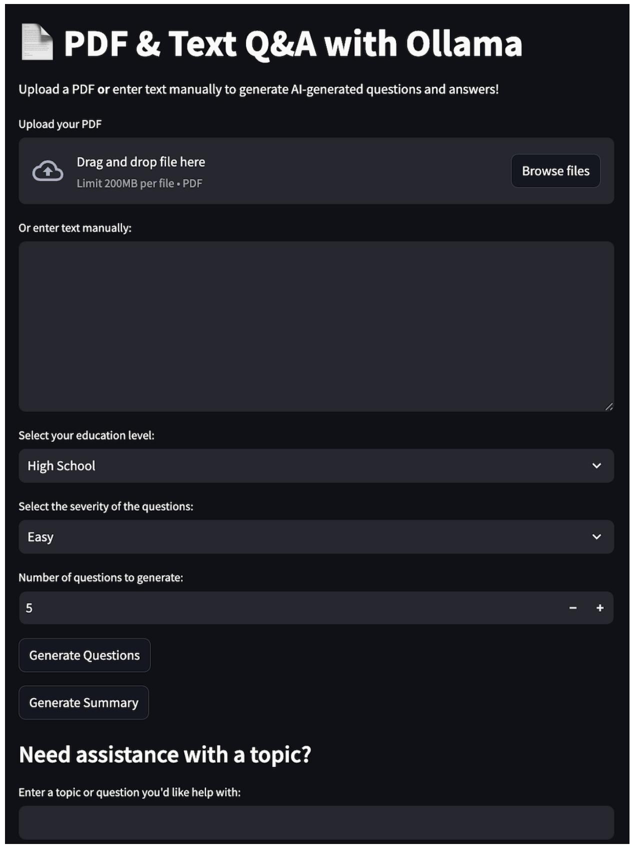

Figure 2: Application Interface

By leveraging cutting-edge LLMs and a user-friendly design, this project addresses the growing need for personalized educational tools in academia and beyond.

Purpose of the Methodology

The chosen methodology is designed to address the challenges of creating personalized study materials efficiently. By employing GEMMA for summarization and LLAMA 3.2 for question generation, the system ensures high-quality outputs tailored to the user's needs.

1. **Data Preprocessing**
   - Load and preprocess the given input using NLP techniques.
   - Tokenize the dataset to prepare inputs for LLMs.
2. **Context and QA Pipeline**
   - **Model 1 GEMMA:** Summarisation
     - **Input:** Pre-processed Text or a PDF file.
     - **Output:** Summary of the input

- **Model 2 LLaMA 3.2:** Question-Answering Generation
  - **Input:** Pre-processed Text or a PDF file.
  - **Output:** Precise answers and questions generated based on context.

3. **Evaluation**
   - Metrics: Use ROGUE score to determine how the model is performing for generating summaries.
   - Using human as a judge to see the quality of questions generated.

4. **Deployment**
   - Build a user-friendly interface using Streamlit for interactive input, knowledge retrieval, question and summary generation.

Insights to Gain

Through this comparison, we aim to understand:

1. **Model Performance:** How well each model performs its designated task.

2. **Task Optimization:** The suitability of each model for specific educational applications.

This methodology ensures that both summarization and question generation are optimized for diverse educational contexts.

Problem Statement

Defining the Problem

The problem addressed by this project lies in the difficulty of creating personalized study materials manually along with students often struggling with synthesizing large volumes of study material and assessing their understanding effectively. Educators and learners often struggle with curating questions or summaries from large volumes of text, which can be time-consuming and inconsistent.

This project aims to bridge this gap by developing an Automated Learning Assistant, powered by advanced LLMs, that processes educational content, retrieves relevant information, and generates personalized quizzes for learners. The goal is to enhance the learning experience through an interactive and intelligent system that tailors knowledge delivery based on user needs.

Type of Problem

This project focuses on LLM-based tasks such as:

- Text summarization
- Question-answer generation
- Assisted learning applications

Significance

Solving this problem is important because it:

- Reduces the time and effort required to create study materials.
- Enhances accessibility to tailored educational content.
- Supports learners with diverse needs by providing customizable inputs.

The relevance of this project extends to academic research, industry applications (e.g., corporate training), and real-world impact in improving education systems globally.

Data Collection and Preparation

Data Sources

The system processes text data provided by users through manual input or uploaded PDF files.

Data Description

The data processed includes:

- Text extracted from PDF files or manual input.
- Summaries generated by GEMMA.
- Questions generated by LLAMA 3.2 based on specific difficulty levels (easy, medium, hard) and education level (High School, Bachelors, Masters, PhD).

Preprocessing Steps

The data processed includes:

1. **Text Extraction:** Extract text from uploaded PDFs using pypdf if necessary.
2. **Tokenization:** Prepare text for model inputs using tokenization strategies compatible with LLMs.
3. **Embedding:** Use embeddings for efficient processing by GEMMA and LLaMA models.

These preprocessing steps ensure that the input data is optimized for accurate summarization and questions generation.

Selection of LLM Models

Model Consideration

For this project, transformer-based models were prioritized over traditional machine learning models due to their superior performance in natural language processing (NLP) tasks such as summarization and question generation. Specifically, the following models were considered:

- **GEMMA (Sequence-to-Sequence Model):** GEMMA was selected for summarization tasks due to its ability to condense large volumes of text while preserving contextual meaning. Its sequence-to-sequence architecture is well-suited for tasks requiring input-output mapping, such as summarization.
- **T5 (Sequence-to-Sequence Model):** T5 was considered for the same reasons as GEMMA due to its sequence-to-sequence architecture being well-suited for tasks like summarization.
- **LLAMA 3.2 (Decoder Model):** LLAMA 3.2 was chosen for question generation because decoder models excel at generating coherent and contextually relevant text. LLAMA's architecture supports fine-tuning for generating questions across different difficulty levels and educational contexts.

Final Model Selection

The final models were selected based on the following criteria:

1. **Task-Specific Performance:** GEMMA's summarization capabilities and LLAMA's proficiency in generating questions made them ideal candidates.
2. **Scalability:** Both models are scalable and can handle diverse input types (text or PDF).
3. **Ease of Fine-Tuning:** GEMMA and LLAMA offer pre-trained versions that can be fine-tuned on specific datasets to improve performance.

Model Development and Training

Architecture and Configuration

GEMMA (Summarization):

- **Transformer Architecture:** GEMMA uses an encoder-decoder framework with multi-head attention mechanisms.
- **Number of Layers:** 12 encoder layers and 6 decoder layers.

LLAMA 3.2 (Question Generation):

- **Decoder-only Transformer:** LLAMA employs a decoder-only architecture optimized for text generation tasks.
- **Attention Mechanisms:** Multi-head self-attention ensures context-aware question generation.

Transfer Learning

Both GEMMA and LLAMA leveraged pre-trained weights from publicly available datasets.

Evaluation and Comparison

Evaluation Metrics

To compare model performance:

Metric $\downarrow$ / Model $\rightarrow$	**GEMMA**	**T5**
ROUGE-1 F1 SCORE	**0.59**	**0.04**
ROUGE-2 F1 SCORE	**0.28**	**0.03**
ROUGE-L F1 SCORE	**0.32**	**0.04**

Table 1: Summarization Metrics Comparison

- **Summarization Tasks:** ROUGE score was used to measure the quality of summaries generated by GEMMA and T5. As you can see, T5 is under performing when compared to GEMMA, hence we moved forward with GEMMA to perform the summarisation.
- **Question Generation Tasks:** We have used human as a judge to check the quality of questions and it is giving pretty good results so far.