Project Report: PolyLingua AI

Project Title: PolyLingua AI: Voice and Text Assistant for PDF-Based Querying

Team Members:

- Karthick Selvam
- Pranav Parasar
- Maria Varghese
- Shoaib Shaikh

1. Abstract

Accessing specific information from lengthy PDF documents can be a tedious and inefficient process. This is especially challenging for users who prefer voice-based interaction or who communicate in languages other than English. Traditional text-search methods often lack the contextual understanding to provide precise answers, forcing users to spend significant time reading through dense material. PolyLingua AI is a multilingual voice and text assistant designed to fundamentally transform this experience. Users can upload a PDF document and ask questions in their natural language. The system leverages a sophisticated pipeline that includes speech-to-text transcription, Retrieval-Augmented Generation (RAG) with FAISS indexing for efficient information retrieval, and Google's powerful Gemini LLM for intelligent, context-aware response generation. The final answer is converted back to high-quality speech using Google Text-to-Speech (gTTS), creating a seamless and interactive conversational flow. For added convenience, users can also download the spoken response as an MP3 audio file, enhancing accessibility across various devices and for diverse user needs.

2. Project Overview

2.1. Problem Statement

The primary challenge this project addresses is the inefficiency and lack of accessibility in extracting information from PDF documents. Key pain points include:

- **Time-Consuming:** Manually searching and reading long documents is a slow process.
- **Lack of Context:** Standard Ctrl+F search functions are keyword-based and often fail to understand the user's intent or the context of the query.
- **Language Barriers:** Most document interaction tools are optimized for English, creating a barrier for non-English speakers.
- **Accessibility Issues:** Text-only interfaces are not ideal for users who prefer or require voice-based interaction due to preference, multitasking needs, or visual impairments.

2.2. Our Solution

PolyLingua AI provides an intelligent, user-friendly interface to "talk" to your documents. By

integrating multiple AI technologies, our solution allows users to:

1. **Upload a PDF:** The user provides a document to serve as the knowledge base.
2. **Ask a Question:** The user can ask a question via voice or text in multiple languages.
3. **Intelligent Processing:** The system transcribes the voice input, uses a RAG model to find the most relevant passages in the PDF, and sends this context to the Gemini LLM to formulate a precise answer.
4. **Receive a Spoken Reply:** The generated text answer is converted into natural-sounding speech, which is played back to the user and made available for download.

This creates an intuitive and efficient workflow that saves time and makes information more accessible to a global audience.

3. Key Features

- **Multilingual Support:** Accepts voice and text queries in various languages.
- **Voice-to-Text & Text-to-Speech:** Enables a full voice-in, voice-out interaction loop.
- **PDF Document Upload:** Users can easily upload their documents to create a custom knowledge base.
- **Retrieval-Augmented Generation (RAG):** Ensures answers are grounded in the content of the provided document, reducing hallucinations and improving accuracy.
- **FAISS Indexing:** Allows for rapid and scalable searching of document content to find relevant information quickly.
- **Downloadable Audio Responses:** Users can save the generated speech as an MP3 file for offline listening or sharing.

4. Technology Stack

- **Large Language Model:** Google Gemini
- **Information Retrieval:** Retrieval-Augmented Generation (RAG)
- **Vector Indexing:** FAISS (Facebook AI Similarity Search)
- **Speech-to-Text:** (Please specify, e.g., OpenAI Whisper, Google Speech-to-Text)
- **Text-to-Speech:** gTTS (Google Text-to-Speech)
- **Backend Framework:** (Please specify, e.g., Python with Flask/Django)
- **Frontend Interface:** (Please specify, e.g., HTML/CSS, React, Streamlit)

5. Current Status & Next Steps

(This is a placeholder section. You can fill this in with your project's current progress.)

Current Status:

- The core pipeline for PDF processing, RAG-based querying, and response generation is complete.
- Voice input and audio output functionalities have been successfully integrated.
- Initial testing with sample documents shows high accuracy in answering context-specific

questions.

Next Steps:

- Expand language support to include more languages.
- Improve the user interface for a more intuitive experience.
- Conduct user testing to gather feedback and identify areas for improvement.
- Explore options for deploying the application as a web service.