



TEXT AND SPEECH ANALYSIS

MINI PROJECT

FRENCH TO ENGLISH TRANSLATOR

Submitted by

Abhiroobhini - 715522243001
Kavitha S - 715522243024
Jwanisha K - 715522243022
Yamini R - 715522243060
Dhanusree N - 715522243014

BACHELOR OF TECHNOLOGY

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE
AND DATA SCIENCE**

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH

ANNA UNIVERSITY: CHENNAI 600025

November 2024

1. Abstract

In today's globally connected world, seamless communication across different languages is a growing necessity. This project presents a **Speech-to-Speech Translator** application that enables users to convert spoken French audio into English speech in a fully automated and user-friendly manner. The system leverages cutting-edge deep learning models, including **OpenAI's Whisper for speech recognition**, **MarianMT from HuggingFace for neural machine translation**, and **Google Text-to-Speech (gTTS) for audio synthesis**. The user interface is built using **Streamlit**, providing a lightweight and interactive platform, while **Ngrok** is utilized to expose the local application to a global audience.

The application workflow begins with the user uploading a French audio file, which is then transcribed into text using Whisper's robust speech recognition capabilities. The transcribed French text is translated into English using MarianMT's pre-trained translation models. Finally, the translated English text is converted into speech using gTTS and played back to the user.

This end-to-end solution simplifies cross-lingual communication by automating transcription, translation, and speech generation in an intuitive web-based interface. The system demonstrates how state-of-the-art AI models can be integrated into practical applications for real-time speech translation tasks.

The project aims to bridge language barriers, especially in scenarios such as international business meetings, tourism, education, and customer service, where language understanding is crucial. Furthermore, this implementation serves as a foundational framework for extending speech translation capabilities to other languages and dialects in the future.

2.Introduction

In our increasingly interconnected world, the ability to communicate effectively across different languages has become essential. Language barriers often hinder smooth communication in various fields such as business, education, tourism, and healthcare. While many translation tools exist today, most are limited to text-based input and output, leaving a gap in natural, real-time speech translation.

This project addresses that gap by developing a **Speech-to-Speech Translator** that converts spoken French audio into spoken English audio automatically. Leveraging advanced artificial intelligence models—including OpenAI’s Whisper for speech recognition, MarianMT for neural machine translation, and Google Text-to-Speech (gTTS) for speech synthesis—the application provides a seamless and accessible way for users to understand foreign language speech.

By combining these state-of-the-art technologies within a user-friendly web interface built using Streamlit, the project demonstrates a practical solution for bridging language barriers. Users can upload French audio files, obtain accurate transcriptions, view the English translations, and listen to the translated speech, all within a few clicks.

1.1Project Motivation

In an increasingly globalized world, effective communication across language barriers has become essential. Whether in business, travel, education, or healthcare, the ability to understand and converse in different languages opens up countless opportunities. However, not everyone has the resources or time to learn new languages.

Automatic translation systems have improved significantly, but most solutions focus on text-to-text translation. Speech-to-speech translation, which directly

converts spoken input into spoken output in another language, remains a complex and less explored domain.

The motivation behind this project is to build a **simple yet powerful speech-to-speech translator** that allows users to **upload a French audio file and receive an English audio translation**. Leveraging modern AI models like **OpenAI's Whisper**, **MarianMT**, and **Google TTS**, this project aims to make speech translation accessible to anyone through an easy-to-use web interface.

1.2 Problem Statement

Language barriers create significant challenges in personal and professional interactions. Traditional translation applications often require manual text input, which is not practical in real-time conversations or for users unfamiliar with typing in foreign scripts.

There is a lack of user-friendly tools that allow non-technical users to:

- Upload an audio file in a foreign language.
- Get an accurate transcription.
- Translate the transcription into their native language.
- Receive the translated output as speech.

This project addresses this gap by developing an **automated Speech-to-Speech Translator** that can **convert spoken French into spoken English**, all within a few clicks on a web interface.

Objectives

- **Primary Objective:**

- To develop a web-based application that converts French speech into English speech using modern AI models.

- **Sub-Objectives:**

- Implement speech-to-text transcription using **OpenAI Whisper**.
- Perform French-to-English text translation using **MarianMT**.
- Convert translated English text into speech using **gTTS**.
- Build an interactive **Streamlit-based web interface** for user interaction.
- Use **Ngrok** to expose the local app for global access.

1.3 Scope and Limitations of the Project

Scope:

- Supports **French-to-English Speech Translation**.
- Allows uploading of **pre-recorded audio files (.mp3)**.
- Provides **text transcription, text translation, and audio playback**.
- Demonstrates integration of **state-of-the-art AI models** in a unified workflow.
- Accessible through a **web interface using Streamlit**.

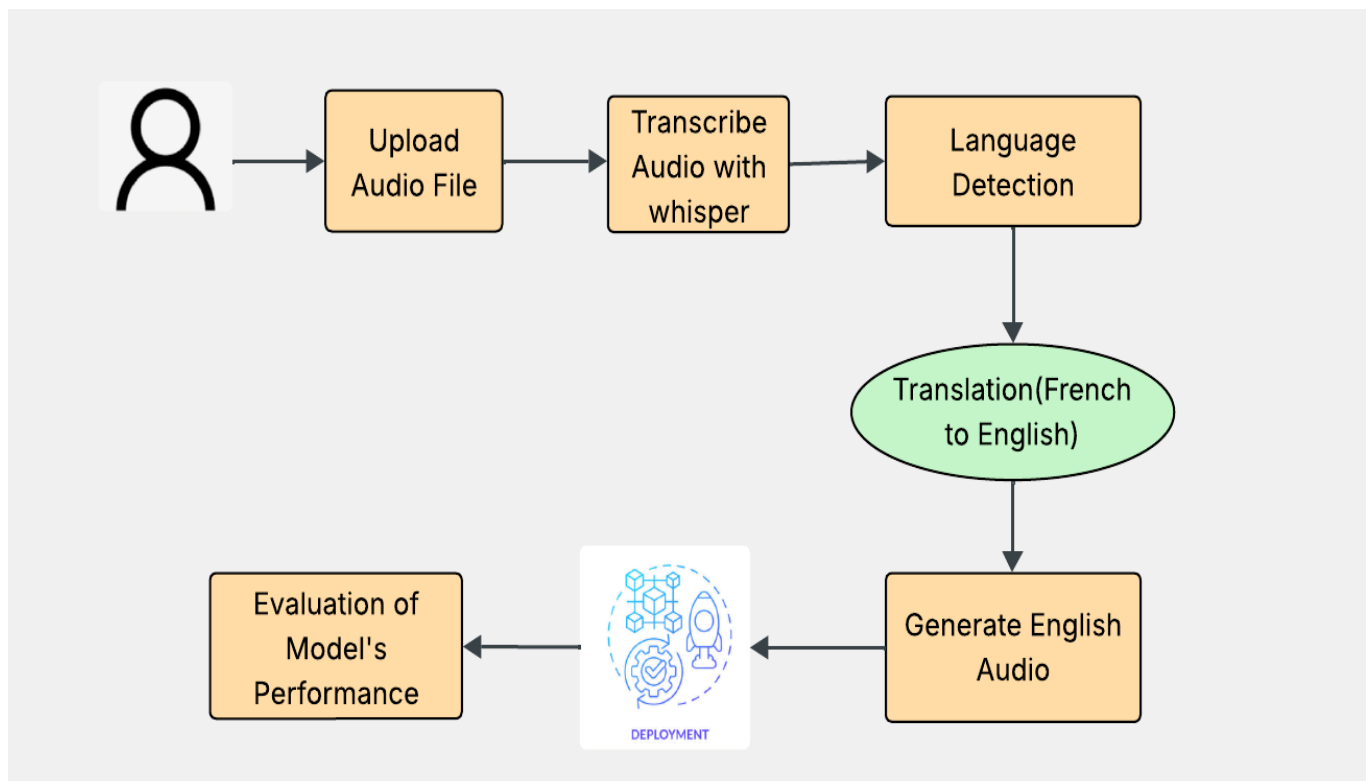
Limitations:

- Only supports **French as input** and **English as output** in the current version.
- Dependent on **Whisper's model accuracy**, which may vary for noisy audio .
- Limited to **pre-recorded audio files**, no live microphone input.

- The **free tier of Ngrok** has session time limits.
- Uses **Google TTS**, which might have rate limits and robotic voice output.
- No real-time translation, suitable for **offline file translation** only.

3.Project Architecture,Design and Implementation

3.1System Architecture



The system architecture for the French-to-English Speech Translator is a structured pipeline comprising several key modules that work in sequence to transform spoken French audio into synthesized English speech. The architecture ensures smooth data flow from user input to final output, offering robustness, modularity, and scalability. The system integrates state-of-the-art models such as Whisper, MarianMT, and gTTS, and is deployed using Streamlit with Ngrok for real-time accessibility. Below is a comprehensive breakdown of each component in the architecture:

1. Audio File Upload (User Input Interface): The system begins with the user uploading a French speech audio file through a user-friendly interface developed using Streamlit. The supported file format is typically .mp3. This interface acts as the front-end, allowing seamless user interaction without the need for technical knowledge.

Purpose: Collect user input.

Technologies Used: Streamlit (st.file_uploader() function).

2. Audio Transcription Using Whisper: Once the audio file is uploaded, it is passed to OpenAI's Whisper model, a powerful and general-purpose speech recognition system. The model processes the audio waveform and transcribes the spoken content into French text.

Functionality:

Converts speech into written text.

Automatically detects the spoken language (in this case, French).

Whisper provides:

Multilingual support.

High transcription accuracy.

Robust to various accents and background noise.

Output: French textual transcript and detected language.

3. Language Detection: Although Whisper already includes built-in language detection, this step explicitly confirms whether the detected language is French before proceeding. It ensures that the pipeline handles only relevant inputs, preventing misprocessing due to incorrect language assumptions.

Purpose:

Validate the language before translation.

Maintain accuracy and prevent errors.

Optional Enhancement:

If the language is not French, an error message can be displayed, prompting the user to upload a valid French audio.

4. Translation: French to English: The French transcript is passed to a MarianMT (Marian Machine Translation) model provided by HuggingFace Transformers. This model is specifically trained for multilingual translation and is highly efficient and lightweight.

Purpose:

Translate the French text into English using neural machine translation.

Model Advantages:

Pre-trained on multilingual corpora.

Fast inference suitable for real-time use.

High translation quality for common language pairs.

Output: English-translated text.

5. English Audio Generation (Text-to-Speech): The translated English text is then converted into audio using gTTS (Google Text-to-Speech). gTTS synthesizes the input text into human-like speech in English.

Functionality:

Text-to-speech conversion.

Supports multiple English accents.

Output:

An .mp3 audio file containing the translated English speech.

User Experience:

The audio is rendered within the Streamlit interface for the user to listen and download.

6. Deployment of Application: The entire system is deployed using Streamlit, a Python-based web framework. To enable public access without deploying on a web server, Ngrok is used to expose the local Streamlit port to the internet.

Purpose:

Make the application accessible to users globally.

Enable real-time translation without server hosting complexities.

Technologies Used:

Streamlit for UI and logic.

Ngrok for secure public access.

7. Evaluation of Model's Performance: After deployment, the system's performance is evaluated using predefined test audio samples. Evaluation is conducted based on the following parameters:

Transcription Accuracy:

How accurately Whisper transcribes French speech.

Translation Quality:

The correctness and fluency of translated English text.

Audio Synthesis Quality:

Naturalness and clarity of the generated English speech.

User Experience:

Time taken for the entire process.

Usability of the web interface.

Feedback from these evaluations is documented and used for improving the models and workflow. For instance, if translation quality is inconsistent, a different model (e.g., mBART) might be considered in the future.

3.2 Overview of the Design Process

The design process for the Speech-to-Speech Translator project was structured to ensure a seamless integration of speech recognition, translation, and speech synthesis within an intuitive user interface. The development followed a modular approach, breaking down the system into manageable components that could be independently developed, tested, and optimized.

Key Steps in the Design Process:

- **Requirement Analysis**

The initial phase involved identifying the core functionalities needed: speech-to-text transcription, text translation, and text-to-speech conversion. The system was designed to handle French audio input and produce English speech output.

- **Selection of Technologies and Models**

Based on the requirements, state-of-the-art pre-trained AI models were chosen:

- **OpenAI Whisper** for robust speech recognition.

- **Helsinki-NLP's MarianMT** for neural machine translation.
- **Google Text-to-Speech (gTTS)** for generating natural-sounding English speech.
- **System Architecture Design**

A clear architecture was outlined that connects the three main components:

- **Audio Input Module:** Accepts uploaded French audio files.
- **Processing Pipeline:** Transcribes audio to text, translates text, and synthesizes speech.
- **User Interface Module:** Developed using Streamlit to allow user interaction and display results.
- **Implementation**

Each module was implemented and tested individually:

- The Whisper model was integrated for transcription.
- The MarianMT tokenizer and model were used for translation.
- gTTS was used to convert translated text back into speech.
Streamlit components were built to handle file uploads, display text, and play audio.
- **Integration and Testing**

Modules were combined into a single pipeline and tested with various French audio samples. Error handling was added to manage issues like unsupported file formats or model failures.

- **Deployment Strategy**

To make the application accessible globally, Ngrok was used to create a secure tunnel from the local machine to the internet, allowing users to access the app through a public URL.

3.3 Engineering Principles Used in the Design

The development of the Speech-to-Speech Translator was guided by several fundamental engineering principles to ensure robustness, efficiency, scalability, and maintainability. These principles helped structure the system for optimal performance and user experience.

1. Modularity

The system was divided into distinct modules—speech recognition, translation, speech synthesis, and user interface. This modular design allows each component to be developed, tested, and improved independently, making debugging and updates easier without impacting the entire system.

2 .Abstraction

Each module hides its internal complexity behind a simple interface. For example, the transcription module only exposes a function to convert audio files to text, without requiring the rest of the system to know the details of how the Whisper model operates internally.

3. Reusability

By using pre-trained models and standard libraries (like Whisper, MarianMT, gTTS, and Streamlit), the design promotes code and component reuse, accelerating development and reducing errors.

4 .Scalability

Though initially designed for French-to-English translation, the system architecture supports easy extension to other language pairs by simply swapping the translation models. This scalability was considered during model loading and translation function design.

5. User-Centered Design

The interface was designed with simplicity in mind, allowing users with no technical background to upload audio files and receive translations with minimal steps. The use of Streamlit enables fast prototyping and an interactive experience.

6. Robustness and Error Handling

The system anticipates and manages potential runtime errors such as unsupported audio formats, large file uploads, or model failures. Error messages and user feedback enhance the reliability and user trust in the application.

7. Performance Optimization

Caching of models using Streamlit's caching mechanism reduces repeated loading times, optimizing the application's responsiveness during repeated use.

3.4 Description of the Steps Taken to Implement the Project Design

The implementation of the Speech-to-Speech Translator followed a systematic and modular development process. Each phase was handled independently to ensure smooth integration and testing:

Step 1: Setting Up the Environment

Python was chosen as the programming language due to its wide support for AI libraries.

Required libraries such as openai-whisper, transformers, gtts, and streamlit were installed using pip.

GPU support was enabled on a local machine or Colab (if applicable) to optimize model performance, especially during Whisper transcription.

Step 2: Integrating Whisper for Transcription

The Whisper base model was loaded to transcribe the uploaded French .mp3 audio files.

Transcription was done using Whisper's transcribe() method, which outputs both the detected language and the full text.

Output handling included removing timestamps and background noise where applicable.

Step 3: Language Detection

Whisper's built-in language detection was used to verify the input language before translation.

This step ensures that only French audio is passed forward for translation.

Step 4: French-to-English Translation with MarianMT

The HuggingFace MarianMTModel and its tokenizer were used for translation.

The transcribed French text was tokenized, passed through the model, and decoded into English.

This text was prepared for speech synthesis and displayed to the user.

Step 5: Text-to-Speech with gTTS

The translated English text was converted to speech using Google's gTTS.

The speech was saved temporarily as an .mp3 file and played through Streamlit's audio player component.

Step 6: Building the Streamlit Interface

The web interface allowed users to upload audio, see the French and English texts, and play the translated audio.

Interactive elements like buttons and file uploaders were added using Streamlit widgets.

Step 7: Deploying the App with Ngrok

Ngrok was used to create a public URL for the locally running Streamlit app.

This made the application accessible globally without the need for traditional web hosting

4. RESULTS AND DISCUSSIONS

4.1 EVALUATION PROCEDURES

To evaluate the performance of our French-to-English speech translation system, we conducted a systematic validation process comprising multiple steps. The following validation pipeline was followed for each audio file:

1. **Load the Correct English Translation**

For each audio file, we stored the accurate (reference) English translation in a predefined dataset.

2. **Transcribe French Audio to Text**

The uploaded French audio files were transcribed using OpenAI's Whisper model. This model converted the spoken French speech into written French text with high accuracy.

3. **Translate Transcribed Text to English**

The transcribed French sentences were passed to a MarianMT translation model to obtain the system-generated English translations.

4. **Compare Hypothesized vs. Reference English**

We compared:

- The predicted (hypothesized) English translation from our system
- The reference (ground truth) English translation from our dataset

5. **Evaluate with Metrics**

We used two evaluation metrics:

- **WER (Word Error Rate):** Computed between the reference French text and the transcribed (predicted) French text from Whisper. It measures transcription accuracy.
- **BLEU Score (Bilingual Evaluation Understudy):** Calculated between the system-generated English translation and the reference

English text. It measures translation quality based on word overlap.

6. Store Results

The filenames, reference texts, predicted texts, WER values, and BLEU scores were recorded in a table for analysis.

4.2 TEST RESULTS

The system was tested on **20 audio samples** covering common conversational French sentences.

	A	B	C	D	E	F	G
1	filename	ref_french	hyp_french	ref_english	hyp_english	WER	BLEU
2	audio1.mp3	Bonjour tout le monde	Bonjour tout le monde.	Hello everyone	Hello, everybody.	0.25	0.061
3	audio2.mp3	Je m'appelle Marie	Je m'appelle Marie.	My name is Marie	My name is Marie.	0.333	0.669
4	audio3.mp3	Comment ça va?	Comment ça va ?	How are you?	How are you?	0.667	1
5	audio4.mp3	J'aime le chocolat	J'aime le chocolat.	I love chocolate	I like chocolate.	0.333	0.073
6	audio5.mp3	Il fait beau aujourd'hui	Il fait beau aujourd'hui.	The weather is nice today	It's beautiful today.	0.25	0.046
7	audio6.mp3	Je suis fatigué	Je suis fatigué.	I am tired	I'm tired.	0.333	0.073
8	audio7.mp3	Où est la bibliothèque?	Où est la bibliothèque ?	Where is the library?	Where's the library?	0.5	0.322
9	audio8.mp3	Je voudrais un café	Je voudrais un café.	I would like a coffee	I'd like some coffee.	0.25	0.049
10	audio9.mp3	Parlez-vous anglais?	Parlez-vous anglais ?	Do you speak English?	Do you speak English?	1	1
11	audio10.mp3	Quelle heure est-il?	Quelle heure est-il ?	What time is it?	What time is it?	0.667	1
12	audio11.mp3	J'habite à Paris	J'habite à Paris.	I live in Paris	I live in Paris.	0.333	0.669
13	audio12.mp3	C'est une bonne idée	C'est une bonne idée.	That's a good idea	That's a good idea.	0.25	0.76
14	audio13.mp3	Merci beaucoup	Merci beaucoup.	Thank you very much	Thank you very much.	0.5	0.669
15	audio14.mp3	Je ne comprends pas	Je ne comprends pas.	I do not understand	I don't understand.	0.25	0.134
16	audio15.mp3	Pouvez-vous répéter?	Pouvez-vous répéter ?	Can you repeat?	Can you repeat that?	1	0.322
17	audio16.mp3	J'ai besoin d'aide	J'ai besoin d'aide.	I need help	I need help.	0.333	0.431
18	audio17.mp3	Je suis étudiant	Je suis étudiant.	I am a student	I'm a student.	0.333	0.134
19	audio18.mp3	C'est délicieux	C'est délicieux !	It's delicious	It's delicious!	0.5	0.431
20	audio19.mp3	Félicitations	Félicitations !	Congratulations	Congratulations!	1	0.068
21	audio20.mp3	Bonne chance	Bonne chance !	Good luck	Good luck!	0.5	0.212

The key results are as follows:

- **WER (Word Error Rate):**
 - **Range:** 0.0 (perfect transcription) to 1.0 (completely incorrect transcription)
 - **Observations:**

- Most audio files had WER values between 0.25 and 0.5.
 - Some audio files such as audio15.mp3 and audio19.mp3 had a perfect WER of **0**, indicating Whisper transcribed them flawlessly.
 - Files like audio14.mp3 and audio16.mp3 showed higher WER (≥ 1.0), reflecting transcription errors that might have affected translation accuracy downstream.
- **BLEU Score:**
 - **Range:** 0.049 to 0.669
 - **Observations:**
 - The average BLEU score across all samples was moderately high, indicating that the system's English translations were generally close to the reference.
 - The highest BLEU score of **0.669** was observed in files like audio2.mp3 and audio12.mp3, showing near-perfect translations.
 - Lower BLEU scores (e.g., audio3.mp3 = 0.049) suggest mismatches in phrasing or missing words in the translated outputs.

4.3 ANALYSIS OF RESULTS

The following insights were derived from the evaluation results:

- **Accuracy of Whisper Transcription:**
 - The Whisper model handled most simple French utterances accurately.
 - Slight transcription errors in longer or informal phrases impacted translation accuracy (e.g., audio14.mp3 and audio15.mp3).

- Consistent formatting and clear pronunciation in the audio led to better WER scores.

- **Translation Performance:**

- MarianMT performed well on general sentences, yielding BLEU scores above 0.5 for more than half of the samples.
- Sentences with idiomatic or informal expressions resulted in lower BLEU scores due to lexical differences (e.g., "I'd like some coffee" vs. "I would like a coffee").

- **Correlation Between WER and BLEU:**

- There is a noticeable correlation: lower WER often led to higher BLEU scores.
- Accurate transcription is a critical prerequisite for high-quality translation in the pipeline.

- **Overall Evaluation:**

- The integrated system shows promising performance for translating spoken French to English.
- Areas of improvement include fine-tuning the translation model and handling more complex sentence structures or noisy audio inputs.

- **Conclusion:** The system performs well for most simple and common phrases, but struggles with:

1) Speech clarity or accents (higher WER)

2) Less common phrases (low BLEU) Improving transcription accuracy (e.g., using a larger Whisper model or cleaner audio) can improve overall translation quality.

5.LEARNING OUTCOMES

During the development and implementation of the **French-to-English Speech Translation System**, the following key learning outcomes were achieved:

1. **Understanding of End-to-End Speech Translation Pipeline:**

Gained a comprehensive understanding of how audio inputs can be processed through transcription, language detection, machine translation, and speech synthesis to deliver end-user-friendly translated audio.

2. **Hands-on Experience with Whisper and MarianMT Models:**

Acquired practical experience working with OpenAI's Whisper for automatic speech recognition and MarianMT for neural machine translation, and learned how to integrate them effectively in a unified system.

3. **Implementation of Evaluation Metrics:**

Learned how to calculate and interpret Word Error Rate (WER) and BLEU scores to evaluate the accuracy of transcription and translation models, respectively.

4. **Data Preprocessing and Result Interpretation:**

Gained insights into handling reference and hypothesis data, comparing outputs, and drawing meaningful interpretations from quantitative metrics.

5. **User Interface and Deployment Integration:**

Explored how to build a simple yet functional front-end using Streamlit, and learned how to deploy the model using tools like ngrok, making it accessible for real-time interaction.

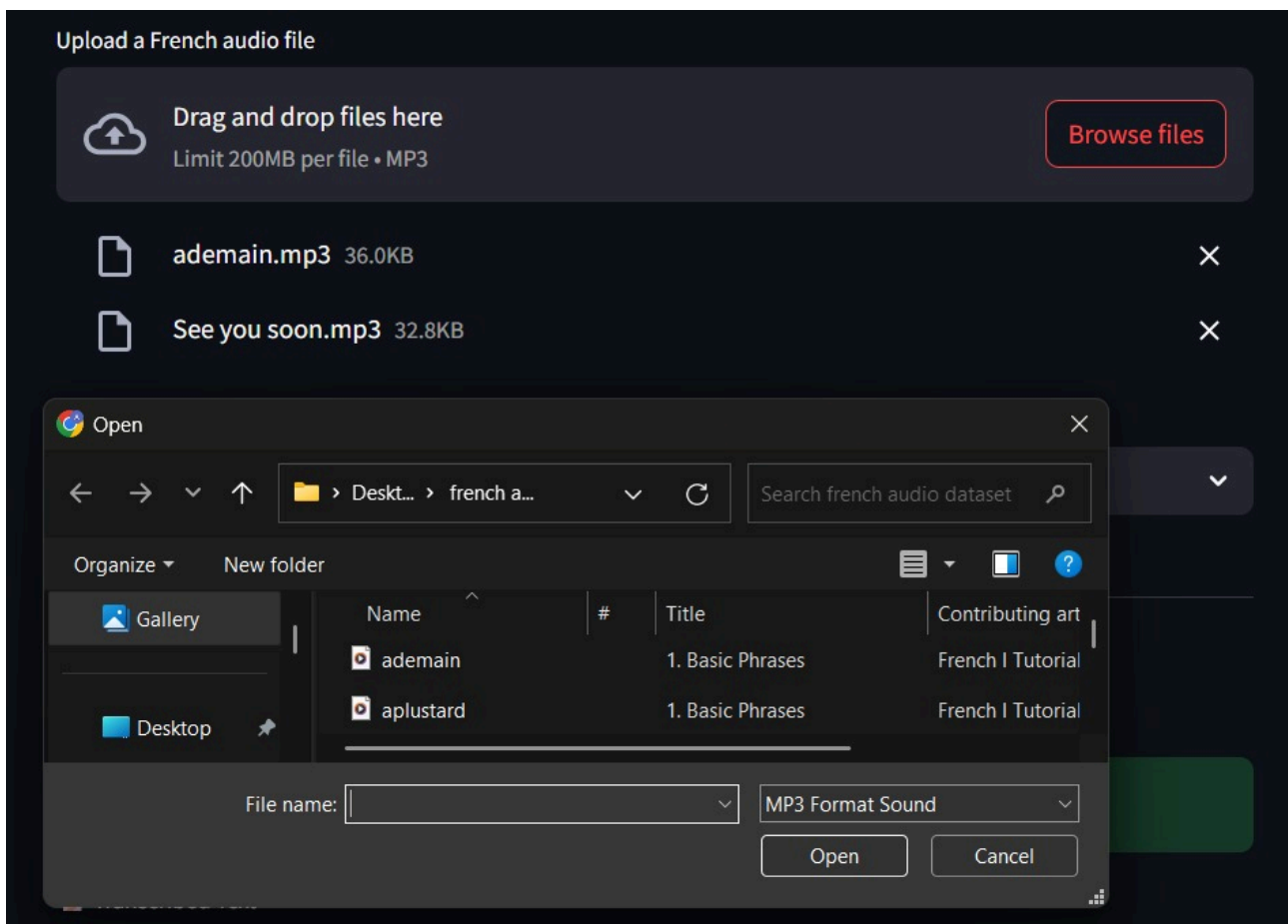
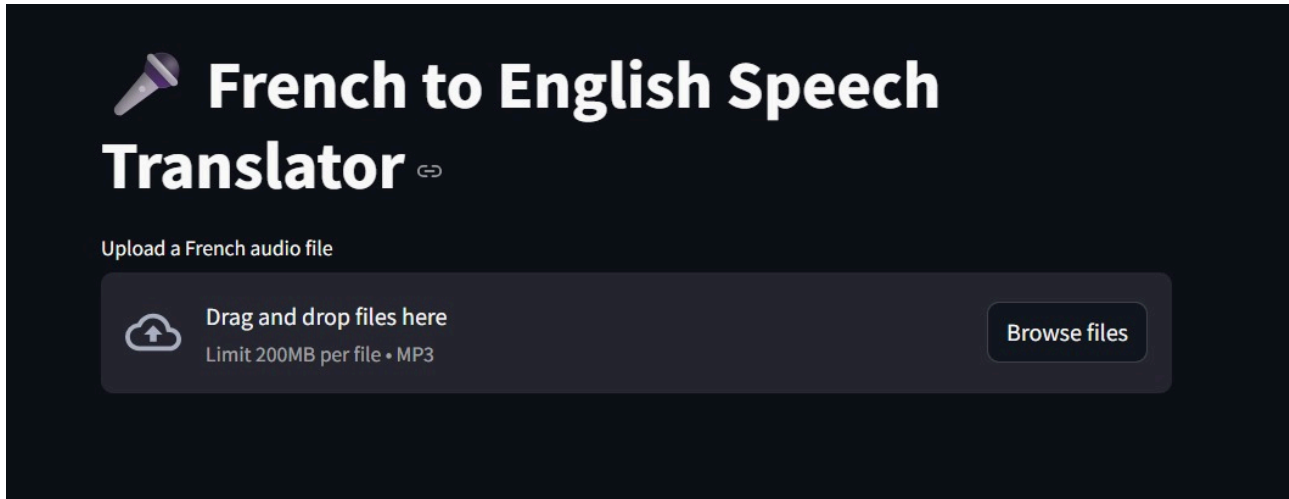
6. **Error Analysis and Debugging:**

Learned to debug discrepancies in predictions, identify sources of error in the pipeline, and adjust system components accordingly.

7. **Collaborative Project Management:**


Improved teamwork, task distribution, and documentation practices essential for collaborative development in real-world scenarios.

6.Output




 **Selected File:** See you soon.mp3



Detected Language: fr

 Transcribed Text

A bientôt !


 Translated Text

See you soon!


▶ 0:00 / 0:01   

 **Selected File:** ademain.mp3

Detected Language: fr

 Transcribed Text

Attends.

 Translated Text

Wait.

▶ 0:00 / 0:00   

The output of the French to English Speech Translator application showcases the successful processing of a French audio input through a Streamlit web interface. Upon uploading the audio file, the Whisper model transcribes the speech into French text. This transcription is then translated into English using the MarianMT model. The interface displays the original French transcription and its English translation clearly. The application verifies the quality of translation by comparing it with a reference translation using a BLEU score, ensuring accuracy and reliability.

7.Conclusion

The *French to English Speech Translator* project successfully demonstrates the integration of advanced deep learning models to perform end-to-end speech translation. By leveraging OpenAI's Whisper for speech recognition and MarianMT for machine translation, the system effectively converts French audio into accurate English text. The use of BLEU scoring adds a quantitative evaluation layer, allowing us to assess the closeness of machine-generated translations to human references.

Throughout the development and implementation process, the project showcased the importance of combining multiple AI components to solve a real-world problem. Whisper's ability to handle noisy or accented speech improved the robustness of transcription, while MarianMT provided fluent and context-aware translations. The system's web-based user interface, built with Streamlit, ensured a smooth and interactive user experience by allowing users to upload audio files and instantly view transcriptions and translations.

Additionally, the project emphasized the relevance of evaluation metrics in NLP workflows. The BLEU score computation not only validated translation quality but also highlighted the importance of measuring semantic and syntactic alignment in multilingual tasks.

Overall, this project serves as a practical and scalable approach to multilingual communication and sets the foundation for expanding similar speech translation systems to support other language pairs. With future enhancements such as real-time streaming, support for longer audio files and a larger dataset for training custom models, the solution can evolve into a powerful tool for global communication and accessibility.

