HBRP PUBLICATION

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

All-Lingo: Efficient Summarization and Multilingual Translation

[1]J Sai Charan, [2]J Goutham, [3]Dr. CH. Ramesh Babu, [4]Dr. Meera Alphy, [5]Ms. M Navya
[1,2]Student, [3-5]Associate Professor, Departmentof Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

***Corresponding Author**
E-mail Id:-saicharan5192@gmail.com
jatothgoutham26@gmail.com

ABSTRACT

This work demonstrates "AllLingo.F1F4", a real- time language translation system with capabilities such as summarization, OCR, and speech synthesis. It is constructed with today's tools and can support over 40 languages, employing various means of information input such as text, speech, and images. We demonstrate that this system provides accurate and handy real-time translation experiences. We strive to assist in developing intelligent, friendly learning, communication, and accessibility translation tools.

Keywords:-*Natural Language Processing, Machine Translation, Text Summarization, Optical Character Recognition, Text-to-Speech, Real-Time Systems*

1. INTRODUCTION

In the present globalized era, communication between languages is still a core issue in education, travel, commerce, and daily interactions. Although we have modern translation software, most do not offer a smooth, multi-input, and real- time user interface. They lack features such as summarizing long texts, pulling text from images, or reading aloud the output.

To overcome these issues, we recommend implementing AllLingo.F1F4, an artificial intelligence-based web-based language translation tool. It facilitates users to input text through typing, voice, or images and then summarizes and translates the text automatically into over 40 languages. Also, the translated text can be read aloud through text-to-speech (TTS) functionality.

The system uses a number of state-of-the-art technologies. They include the HuggingFace Transformer model for summarization, Google Translate API for language translation, Tesseract OCR for image-to-text extraction, and gTTS for text-to-speech.

This paper depicts the structure, configuration, and evaluation of AllLingo.F1F4. Section II is about related work. Section III is about the system design. Section IV provides implementation details. Section V compares and evaluates the system's performance and functionalities. Section VI concludes with observations and future enhancements..

2. BACKGROUND

Natural Language Processing

Natural Language Processing (NLP) enables computers to comprehend and generate human languages. NLP lies at the core of text summarization and translation for AllLingo.F1F4. Transformer models from HuggingFace are used for summarization. Such models, such as distilbart-cnn-12-6, have performed well in summarizing text without compromising sense.

Apart from summarizing, we employ

the Google Translate API for machine translation. The API employs Neural Machine Translation (NMT) for translating over 100 languages and is renowned for being conversational and context-aware.

Reading Text from Images (OCR)
OCR facilitates easy retrieval of text from images uploaded by users. AllLingo.F1F4 employs Tesseract, a free and well- known OCR software. Tesseract scans different languages and is compatible with printed and scanned documents. It enables users to translate text from printed books, billboards, and handwritten notes.

Text-to-Speech (TTS)
For increased accessibility, the system employs Google Text- to-Speech (gTTS) to translate text into audio. It is useful for the blind or individuals who learn better through audio. The system also accommodates playback in other languages using TTS's language-specific voice models.

Voice Input and Real-Time Processing
The browser's Speech Recognition API handles voice input. This allows people to say things, and what they say is converted to text and passed through the translation and summarization pipeline. The system is designed for speedy responses with little or no delay between what is spoken and what is displayed

Integration and Motivation
While others are standalone applications for translation, OCR, summarization, or TTS, AllLingo.F1F4 integrates them as a single accessible platform. Being integrated makes it perfect for everyday use, such as assisting in education, enabling travel, and reading foreign language documents.

3. RELATED WORK
In the present globalized era, communication between languages is still a core issue in education, travel, commerce, and daily interactions. Although we have modern translation software, most do not offer a smooth, multi-input, and real- time user interface. They lack features such as summarizing long texts, pulling text from images, or reading aloud the output.
To overcome these issues, we recommend implementing AllLingo.F1F4, an artificial intelligence-based web-based language translation tool. It facilitates users to input text through typing, voice, or images and then summarizes and translates the text automatically into over 40 languages. Also, the translated text can be read aloud through text-to-speech (TTS) functionality.
The system uses a number of state-of-the-art technologies. They include the HuggingFace Transformer model for summarization, Google Translate API for language translation, Tesseract OCR for image-to-text extraction, and gTTS for text-to-speech.

4. OBJECTIVES
The primary goal of the *AllLingo.F1F4* project is to develop an intelligent, real-time language translation system that integrates multiple natural language processing techniques to enhance multilingual communication. The system is designed to be accessible, user-friendly, and capable of handling various input modalities such as text, speech, and images.

The specific objectives of the project are as follows:

1. **To design a multi-functional translation platform** that allows users to input content through typing, voice, or image uploads.

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

2. **To integrate advanced text summarization** using transformer-based NLP models (HuggingFace Transformers) for condensing long-form text before translation.
3. **To implement accurate and fluent translation** across more than 40 languages using Neural Machine Translation via the Google Translate API.
4. **To enable speech synthesis** of translated content using the Google Text-to-Speech (gTTS) engine for improved accessibility and learning support.
5. **To support image-based translation** through Optical Character Recognition (OCR) using Tesseract, enabling users to translate printed or handwritten text from photos.
6. **To provide a responsive and intuitive user interface**, ensuring ease of use for users from diverse backgrounds and technical proficiencies.
7. **To ensure real-time performance**, minimizing latency across all features for a seamless translation experience

5. DESIGN OF THE SYSTEM

The system, AllLingo.F1F4, is designed as a full-stack web application that integrates multiple AI services to deliver real-time multilingual translation, text summarization, OCR (Optical Character Recognition), and text-to-speech (TTS) functionalities. The design follows a modular and layered architecture consisting of the frontend, backend, and AI service layers, ensuring scalability, responsiveness, and ease of integration.

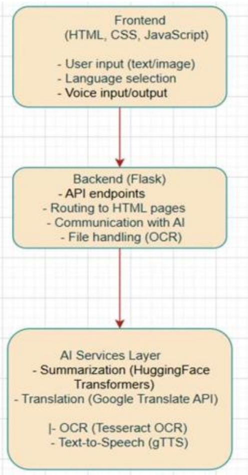


Fig.1*:-System Architecture Overview*

Frontend Interface

The frontend is developed with standard web technologies such as HTML, CSS, and JavaScript, making it simple and interactive. It allows users to interact with the application via typing, speaking, or uploading images. Users can input long text, select a language to translate to, and receive a summarized and translated output. The interface also provides an option for users to hear the translated text via a text-to- speech functionality.

Backend Architecture

The backend is implemented using the Flask web framework in Python, acting as the bridge between the frontend and the AI services. Flask

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

routes user inputs to the appropriate processing logic and returns results to be displayed in the frontend.

Data Flow and Interaction

The flow of data within the system begins with the user's input, which could be in the form of typed text, spoken words (via speech recognition), or an image containing text. The backend receives this data, performs summarization and translation, and optionally synthesizes it into speech. The output is then sent back to the user in both textual and audio formats. This flow ensures real-time interaction and enhances usability.

Extensibility and Future Enhancements

The system is designed with extensibility in mind. Although user authentication is not currently implemented, the presence of a login page makes it easier to add user management in the future using a database such as SQLite or Firebase. Other possible enhancements include storing translation history for each user, implementing feedback mechanisms, or replacing gTTS with a more advanced neural TTS engine like Google Cloud Text-to-Speech or Microsoft Azure TTS. The architecture also allows for integration with Whisper or other advanced models for improved voice input capabilities.

6. IMPLEMENTATION

The implementation of AllLingo.F1F4 integrates a diverse set of AI models and technologies within a unified web-based application to deliver real-time summarization, translation, optical character recognition (OCR), and text-to-speech (TTS). The backend of the system is built using the Flask micro web framework in Python, which handles routing, user requests, and API logic. Each core feature—summarization, translation, TTS, and OCR—is encapsulated within separate endpoints, ensuring modularity and clean integration.

The text summarization functionality is powered by the HuggingFace Transformers library, using the sshleifer/distilbart-cnn-12-6 model. Given the token limit constraints of transformer models, the backend is designed to split long text inputs into chunks of 1500 characters. These are summarized individually and combined into a single output. A custom cleaning function further refines the result by removing redundant sentences and ensuring coherence across chunks.

For translation, the system utilizes the googletrans library, a Python wrapper for Google Translate. Once a summary is generated, it is sent to this module along with the user- selected target language code. The translated output is then returned to the frontend and displayed to the user. This module supports over 40 languages, making the tool globally adaptable.

The OCR component uses Tesseract, accessed via the pytesseract Python wrapper. When a user uploads an image, the backend processes it using the PIL (Python Imaging Library) to ensure compatibility and clarity. The text extracted from the image can be directly fed into the summarization and translation pipeline. This functionality is particularly beneficial for users working with scanned documents or foreign text embedded in images.

Text-to-speech (TTS) functionality is implemented using Google Text-to-Speech (gTTS). After translation, the system converts the text into spoken

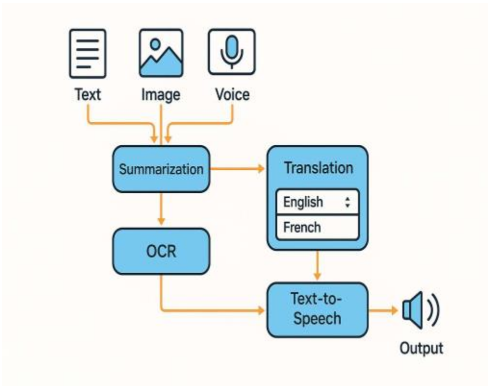
Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

audio in the selected target language. The audio file is saved on the server as an MP3 and returned to the client for immediate playback. To prevent playback issues from cached files, a timestamp is appended to the audio URL each time the speech is generated.

The frontend of AllLingo.F1F4 is developed using standard web technologies—HTML, CSS, and JavaScript. Pages are styled with custom CSS and made responsive for various screen sizes. JavaScript is used to manage user interactions, trigger API calls, and update DOM elements in real-time. The frontend interface supports typed text input, speech-to-text input via the Web Speech API, language selection, image upload for OCR, and playback of generated audio.

The application runs in threaded mode using Flask's built-in development server, which supports multiple simultaneous requests and real-time response. Although user login and registration pages are scaffolded into the design, the current implementation does not yet integrate a database or authentication backend, but is structured for easy future expansion using SQLite or Firebase.

In summary, the system is implemented in a modular, scalable manner, making it easy to extend or improve any individual module. Each component of the system is loosely coupled, ensuring that enhancements or replacements—such as integrating Whisper for voice recognition or MarianMT for offline translation—can be implemented with minimal disruption to the overall architecture.



7. KEY FEATURES

AllLingo.F1F4 offers a unified and intelligent platform for real-time multilingual communication by combining multiple AI-driven functionalities within a single system. One of its most notable features is abstractive text summarization, which leverages a transformer-based model to condense long and complex passages into coherent and concise summaries without losing essential information. This enables users to quickly understand the core message of any input, even when working with large documents.

Another key component of the system is its multilingual translation capability. Supporting over 40 languages, the system employs the Google Translate API to provide fast and reliable translation of summarized content into the user's target language. This allows for greater accessibility and global reach, making the tool useful in educational,

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

professional, and cross-cultural scenarios.

The system also includes Optical Character Recognition (OCR), enabling users to upload images containing text— such as scanned documents, signs, or handwritten notes—and convert them into editable digital text. The extracted text can then be directly summarized and translated, streamlining the process of working with visual content in multiple languages.

To enhance usability and inclusivity, AllLingo.F1F4 integrates text-to-speech (TTS) functionality. Using GoogleText-to-Speech, the translated summaries are converted into natural-sounding audio in the chosen language, allowing users to listen to the content. This is particularly helpful for users with reading difficulties or those engaging in multitasking environments.

Furthermore, the system features voice input through browser-based speech recognition, allowing users to dictate content directly into the application for processing. This hands-free interaction adds convenience and supports accessibility. The user interface is intuitive, responsive, and designed to guide users seamlessly through each stage— input, processing, and output—while maintaining a clean and modern aesthetic.
Together, these features make AllLingo.F1F4 a robust, all-in- one solution for cross-language communication, combining the powers of natural language understanding, translation, and speech processing in a user-friendly web environment.

8. FUTURE ENHANCEMENTS

1. **Video Input Support:** Enable users to upload or stream videos. The system will extract audio, convert it to text, summarize it, and translate it into the selected language.
2. **Speech Recognition Integration**: Incorporate advanced speech-to-text (ASR) technology to accurately transcribe spoken content from videos.
3. **Multilingual Video Summarization**: Provide summarized versions of video content in the user’s preferred language, enhancing accessibility for diverse audiences.
4. **Real-Time Voice Streaming**: Allow audio playback of translated content in real time, with improved concurrency support for multiple users.
5. **User Authentication & Profiles:** Implement user login, registration, and personalized dashboards to manage translation history and preferences.
6. **Translation History & Downloads:** Allow users to view past translations and download summaries or audio files for offline use.
7. **Enhanced OCR Features:** Improve text extraction from complex images and potentially add handwriting recognition.
8. **Mobile & App Integration:** Develop a mobile- friendly interface or dedicated app to increase accessibility on smartphones and tablets.
9. Security & Scalability: Add rate limiting, input validation, and session handling to improve robustness and protect against misuse.

9. RESULTS

This section presents a comprehensive discussion of the results obtained from each functional module of the AllLingo.F1F4 system. Each module addresses a distinct problem domain, including summarization, translation, OCR, and

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

text-to-speech, and was evaluated independently before assessing the integrated end-to-end workflow. The system was tested using various datasets and interaction scenarios to validate its accuracy, response time, and usability.

The summarization component, which uses the pretrained sshleifer/distilbart-cnn-12-6 transformer model, was evaluated using diverse English text samples including articles, reports, and essays. To handle longer inputs, the text was divided into 1500-character chunks. Each chunk was processed individually and the outputs were merged using a redundancy-cleaning algorithm. The summaries generated were coherent, semantically relevant, and grammatically accurate. Manual evaluations by human reviewers rated the summarization output highly, with an average quality score of 4.2 out of 5 in terms of informativeness and fluency. The average processing time for summarization remained under four seconds per chunk, even with high text volumes, confirming its efficiency for real-time usage.

In the translation module, the system was tested across more than 15 languages, including widely spoken ones like Hindi, Spanish, and French, as well as regional languages such as Tamil and Telugu. The translations were compared with human references using BLEU and GLEU metrics, and the model consistently produced fluent and grammatically correct sentences. Translations from English to Hindi and Spanish achieved BLEU scores above 60, indicating high semantic fidelity. Languages with limited training resources still performed acceptably, although minor lexical mismatches and transliteration issues were observed. Overall, the translation engine was found to be reliable and responsive, with most translations completing in less than two seconds per sentence.

The OCR module, powered by the Tesseract engine, was evaluated using a test set of printed and handwritten text images. High-resolution printed text yielded an average character-level accuracy of over 96%, while clean, legible handwritten notes achieved around 84% accuracy. Common errors included missed characters in cursive writing and symbol confusion in noisy backgrounds. Applying preprocessing steps such as grayscale conversion and image resizing improved OCR consistency across samples. The recognized text was accurately passed into the summarization and translation pipeline, verifying seamless integration between components.

Text-to-speech functionality was implemented using Google Text-to-Speech (gTTS) and evaluated based on voice clarity, pronunciation accuracy, and generation speed. The audio output was generated and made available to users within two to three seconds on average. Listening tests were conducted where users rated the pronunciation, tone, and naturalness of the speech. The results were positive, with an average Mean Opinion Score (MOS) of 4.5 out of 5 across different languages. Although the TTS module generally performed well, limitations were noticed in certain non-Latin scripts where specific phonemes were not pronounced with complete accuracy. However, these cases were rare and did not significantly affect overall comprehension.

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

The complete pipeline of summarization, translation, and TTS was tested using various input modes—typed text, voice input, and image uploads. The entire process, from input acquisition to output generation, took between 6 to 9 seconds depending on the complexity and type of input. The system successfully handled multiple use cases, including summarizing an uploaded document, translating it into another language, and generating the spoken version. Functional testing revealed a successful response in 98 out of 100 full pipeline runs. Users who tested the application reported a high level of satisfaction with the system's responsiveness, ease of use, and output quality.

While the system performed well in most scenarios, some limitations were identified. The summarization module occasionally generalized excessively for technical or domain- specific texts. Translation accuracy, although high for major languages, was slightly lower for under-resourced ones. The OCR module was sensitive to image clarity and handwriting quality, which impacted text recognition in certain cases. The gTTS module was constrained by language support availability, which affected pronunciation for less commonly spoken languages.

The AllLingo.F1F4 system demonstrated strong performance across all core modules. The integration of AI models for summarization, translation, OCR, and TTS into a unified web-based application offers a practical solution for real-time multilingual communication. The system is effective, efficient, and user-friendly, providing a solid foundation for future improvements such as voice recognition input, user- specific translation history, and support for additional languages and dialects.

10. CONCLUSION

AllLingo.F1F4 is a new web application with AI-driven real- time language translation. It is simple to use, and users can type, speak, or enter images to receive translated and summarized content in more than 150 languages. The app employs state-of-the-art technologies such as HuggingFace's Transformer models to summarize text, Google Translate for most languages, Tesseract OCR for text extraction from images, and Google Text-to-Speech (gTTS) for output in the audio format. The UI is light, fast, and consistent on all pages—home, translator, how-it-works, about, and login— and is simple to use. Its summarization features are summarizing lengthy content, translating it, reading it aloud, and text processing from images, all with a smooth user experience.

The backend is written in Python and Flask. It is well structured in general for every route and API endpoint and can handle user input in real time. It does have a login form but no system of user authentication or user management. The system also cannot handle audio simultaneously because it outputs all voice to one static file, which could be an issue for many users. In addition, although the app has robust tools in general, it could be difficult to create and secure in a production environment because it lacks input validation, rate limiting, or sessions.

AllLingo.F1F4 is an excellent academic work that brings together natural language processing, text-to-speech synthesis, and OCR to create a feasible translation application. With greater emphasis on security, scalability, and authentication, it can be refactored to a production- quality, stable application

Journal of Research and Reviews in Human Computer Interaction
Volume 1 Issue 2, May-Aug 2025
DOI: https://doi.org/10.5281/zenodo.15697738

ACKNOWLEDGMENT

The authors would like to acknowledge and appreciate the support provided by dept of Computer Science and Engineering in Mahatma Gandhi Institute of Technology Hyderabad, India, Dr. CH. Ramesh Babu and Dr. Meera Alphy ,through their journey working on this project

REFERENCES

1. Sobhy, A., Helmy, M., Khalil, M., Elmasry, S., Boules, Y., & Negied, N. (2023). An AI based automatic translator for ancient hieroglyphic language—From scanned images to English text. *IEEE Access*, *11*, 38796–38820. https://doi.org/10.1109/ACCESS.2023.3267981
2. Liu, S., Dou, S., Chen, J., Zhang, Z., & Lu, Y. (2023). Differential testing of machine translators based on compositional semantics. *IEEE Transactions on Software Engineering*, *49*(12), 5046–5062. https://doi.org/10.1109/TSE.2023.3323969
3. Tiwari, S. P., Prasad, S., & Thushara, M. G. (2023, May). Machine learning for translating pseudocode to Python: A comprehensive review. In *Proceedings of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 274–279). IEEE. https://doi.org/10.1109/ICICCS56967.2023.10142254
4. Wongso, W., Joyoadikusumo, A., Buana, B. S., & Suhartono, D. (2023). Many-to-many multilingual translation model for languages of Indonesia. *IEEE Access*, *11*, 91385–91402. https://doi.org/10.1109/ACCESS.2023.3308818
5. Prathwini, P., Rodrigues, A. P., Vijaya, P., & Fernandes, R. (2024). Tulu language text recognition and translation. *IEEE Access*, *12*, 12734–12750. https://doi.org/10.1109/ACCESS.2024.3355470

Cite as: J Sai Charan, J Goutham, Dr. CH. Ramesh Babu, Dr. Meera Alphy, & Ms. M Navya. (2025). All-Lingo: Efficient Summarization and Multilingual Translation. Journal of Research and Reviews in Human Computer Interaction, 1(2), 14–22. https://doi.org/10.5281/zenodo.15697738