VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI, KARNATAKA – 590018

[A STATE TECHNOLOGICAL UNIVERSITY]

A PROJECT REPORT ON

“Speech-to-Speech Translation for Indic Language”

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

ARTIFICIAL INTELLEGENCE AND MACHINE LEARNING

SUBMITTED BY

NAME	USN
MAHANTAGOUDA PATIL	2BA20AI011
MAKINENI ADITYA VARDHAN	2BA20AI012
SUDEEP RENUKRAJ DODDAMANI	2BA21AI403

UNDER THE GUIDANCE OF
Dr. B. M. Reshmi

DEPARTMENT OF ARTIFICIAL INTELLEGENCE AND MACHINE LEARNING

B. V. V. SANGHA’S
BASAVESHWAR ENGINEERING COLLEGE, S. NIJALINGAPPA VIDYANAGAR
BAGALKOTE – 587102
[A GOVERNMENT AIDED COLLEGE, RECOGNIZED BY AICTE, PERMANENTLY AFFILIATED TO VTU, BELAGAVI & ACCREDITED BY NAAC WITH ‘A’ GRADE]

Academic Year: 2023-2024

B. V. V. Sangha's

BASAVESHWAR ENGINEERING COLLEGE,

BAGALKOTE-587102

CERTIFICATE

This is to Certify that the project work entitled ***" Speech-to-Speech Translation for Indic Language "***, carried out by **Mr. Mahantagouda Patil (2BA20AI011), Mr. Makineni Aditya Vardhan (2BA20AI012), Mr. Sudeep Renukraj Doddamani (2BA21AI403)** are bonafide students of **Basaveshwar Engineering College, Bagalkote**, in partial fulfilment for the award of **Bachelor of Engineering in Artificial Intelligence And Machine Learning** of the **Visvesvaraya Technological University, Belgaum** during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Semester End Examinations have been incorporated in the Report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Dr. B. M. Reshmi	**Prof. N. B. Kalligudd**	**Dr. A. D. Devanagavi**	**Dr. Veena Soraganvi**
Project Guide	Project Coordinator	HOD	Principal

Name of the Examiners **Signature**

1. **…………………………………………………………. ……………………………**

2. **…………………………………………………………. ……………………………**

3. **…………………………………………………………. …………………………....**

DECLARATION

We, **Mr. Mahantagouda Patil (2BA20AI011), Mr. Makineni Aditya Vardhan (2BA20AI012), Mr. Sudeep Renukraj Doddamani (2BA21AI403)** students of 8TH semester Bachelor of Engineering in Artificial Intelligence And Machine Learning, Basaveshwar Engineering College, Bagalkote, hereby declare that the project work entitled " **Speech-to-Speech Translation for Indic Language** " submitted to the Visvesvaraya Technological University during the academic year 2023-2024, is a record of an original work done by us under the guidance of **Dr. B. M. Reshmi,** Department of Artificial Intelligence And Machine Learning, Basaveshwar Engineering College, Bagalkote. This project work is submitted in fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Artificial Intelligence and Machine Learning. The result embodied in this project have not been submitted to any other University or institute for the award of any degree.

Place : BAGALKOTE

Date : 06-06-2024

MAHANTAGOUDA PATIL
USN:2BA20AI011

MAKINENI ADITYA VARDHAN
USN:2BA20AI012

SUDEEP RENUKRAJ DODDAMANI
USN:2BA21AI403

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible, without whose constant co-operation and encouragement would have made efforts go in vain. We consider privileged to success gratitude and respect towards all those who guided us through the completion of this project.

We convey thanks to our guide **Dr Bharti Reshmi** Asst. professor, department of Artificial Intelligence and Machine Learning. Basaveshwar Engineering College for providing encouragement, constant support and guidance which was of a great help to complete this project successfully.

We are grateful to **Dr. A.D. Devanagavi,** Professor and Head of Department of Artificial Intelligence and Machine Learning, Basaveshwar Engineering College for giving us the support and encouragement that was necessary for completion of this project.

We would also like to express our gratitude to **Dr. Veena Soraganvi** Principal, Basaveshwar Engineering College for providing us congenial environment to work in.

We are grateful to **Basaveshwar Engineering College, Bagalkote** with their very ideas and rations for providing the facilities which have helped us making in this project a success.

Finally, we would like to thank, my parents and friends for their constant encouragement with moral and material support.

ABSTRACT

This project presents the development of a speech-to-speech translation system capable of translating spoken Kannada language into English in real-time. The system leverages the power of Large Language Models (LLMs) and a technique called LoRa (Low-Rank Adaptation) to achieve accurate and efficient translation.

Automatic Speech Recognition (ASR) converts spoken Kannada audio into text. This transcript is then fed into a pre-trained LLM, specifically LAMA due to its adaptability.LLAMA is further fine-tuned for Kannada-English translation using LoRa, enabling efficient adaptation without modifying a vast portion of the LLM's parameters. Finally, Text-to-Speech (TTS) synthesis converts the translated English text back into natural-sounding speech.

The project demonstrates the successful integration of these components for speech-to-speech translation. It highlights the potential of LLMs with LoRa for real-world applications, paving the way for more robust and versatile translation systems that can bridge communication gaps across languages.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

- The motivation behind the "Kannada to English Speech-to-Speech Translation" project stems from the pressing need for effective communication tools in a linguistically diverse country like India. Kannada, spoken by millions primarily in the state of Karnataka, is one of India's many regional languages. However, English remains the dominant language in education, business, and international communication. This language barrier often hinders effective interaction and understanding between Kannada speakers and the wider English-speaking community.
- In many rural and urban areas, people may not have the proficiency in English needed to access information, education, and opportunities that are predominantly available in English. Conversely, English speakers may find it challenging to engage with Kannada-speaking communities, limiting social, cultural, and economic exchanges.

1.2 OBJECTIVES

- The primary objective of this project is to develop an effective speech-to-speech translation system tailored for Kannada to English languages.
- Accurate Translation: Ensure high accuracy in translation to preserve the semantic and contextual meaning of the spoken input across languages.
- User-Friendly Interface: Develop an intuitive user interface accessible to individuals with varying levels of technological proficiency, ensuring seamless adoption and usability.

- Bridging the communication gap: Speech translation can enable smooth interaction between people who don't share a common language. Break down language barriers and facilitate seamless communication among diverse linguistic communities

- It facilitates storage of student information and CVs, allowing updates. Notifications about companies are sent, and students can access past placement information seamlessly.

1.3 SCOPE OF PROJECT

- The "Kannada to English Speech-to-Speech Translation" project aims to develop a comprehensive system capable of translating spoken Kannada into spoken English in real-time. This involves integrating advanced speech recognition (ASR), natural language processing (NLP), and speech synthesis (TTS) technologies to create a seamless translation pipeline. The core component of the system is a large language model (LLM) fine-tuned using the Low-Rank Adaptation (LORA) algorithm, which ensures high translation quality by capturing the nuances and complexities of both languages. The project will also focus on creating an intuitive user interface that facilitates easy interaction for users of all ages and technical backgrounds, promoting widespread adoption.

- Furthermore, the scope extends to applying this translation system across various sectors such as education, healthcare, tourism, and customer service, demonstrating its versatility and impact. The system will undergo rigorous testing and validation to ensure accuracy, latency, and reliability, making it suitable for real-world applications.

- The project also aims to be scalable, allowing for the addition of more languages and dialects in the future, thus setting a foundation for broader multilingual communication solutions. By addressing these aspects, the project seeks to bridge language barriers, enhance communication, and foster inclusivity between Kannada and English speakers.

1.4 LITERATUR E SURVEY

Sl. No	Title	Authors	Description	Technology Used	Year
1	A Survey on Deep Learning Techniques for Kannada Speech Recognition	Sunayana	It explores different model architectures, training strategies, and evaluation metrics for achieving accurate conversion of spoken Kannada audio into text transcripts.	Deep Learning	2022
2	Hierarchical Transformer with Latent Mixing for Machine Translation	Wang, A., Singh	This paper proposes a novel machine translation architecture called Hierarchical Transformer with Latent Mixing. This model leverages hierarchical encoding and latent variable mixing to improve translation accuracy, fluency, and robustness, especially for complex languages.	Large Language Models (LLMs) with a specific Transformer-based architecture for machine translation .	2021
3	A Comparative Study of English to Kannada Baseline Machine Translation System with General and Bible Text Corpus	Patil, S. B., & Kulkarni	This paper compares the performance of baseline machine translation systems for translating English text to Kannada. It analyzes the effectiveness of using different text corpora (general text vs. Bible text) for training the	Statistical Machine Translation (baseline approach) for machine translation.	2014

			translation models.		
4	Low-Rank Adaptation for Speaker-Independent TTS on Limited Speech Data	Chowdhury, S., Mishra, S., & Murthy, H. K	This paper explores the use of Low-Rank Adaptation (LORA) for speaker-independent text-to-speech (TTS) systems.	Low-Rank Adaptation (LORA) for adapting text-to-speech models.	2021

CHAPTER 2

PROBLEM FORMULATION

2.1 INTRODUCTION

- Achieving natural-sounding speech translation from Kannada to English presents several hurdles. Accurately converting Kannada speech to text, preserving meaning and naturalness during translation, and generating English speech that sounds fluent are key challenges. This project proposes a system that tackles these issues using a combination of techniques.
- Collection and prepare a dataset of Kannada speech recordings paired with their English translations. Then, a specialized speech recognition model will be trained to transcribe Kannada audio into text. To bridge the language gap, a large language model (LLM) will be fine-tuned on the Kannada-English text data, specifically for translating Kannada text to English while maintaining fluency and accuracy. Finally, a pre-trained English text-to-speech model will be adapted using LORA (Low-Rank Adaptation) to convert the translated English text into natural-sounding English speech.

2.2 PRESENT SYSTEM

Existing speech translation models, even for well-resourced languages like English, have limitations that can be particularly pronounced when dealing with lower-resource languages like Kannada. Here are some key shortcomings:

1 **Accuracy and Fluency:** Machine translation models, while constantly improving, can still struggle with capturing the nuances of human language. This can lead to errors in meaning, unnatural phrasing, and grammatical mistakes in the translated speech.

2 **Context and Idioms:** Existing models often have difficulty understanding the context of a conversation and translating idioms or cultural references accurately. This can lead to misunderstandings and a loss of the intended meaning.

3 **Limited Data for Low-Resource Languages:** Many speech translation models are trained on massive datasets of text and speech, primarily in high-resource languages like English. Languages like Kannada may have limited available data, hindering the model's ability to learn the intricacies of the language and translate effectively.

4 **Naturalness of Speech:** Generating speech that sounds natural remains a challenge. Translated speech can sound robotic or lack the proper intonation and prosody, making it difficult to understand or follow.

5 **Speaker Independence:** Current models might struggle to adapt to different speakers' accents and speaking styles, leading to inaccuracies in the translation.

2.3 PROBLEM STATEMENT

Developing a robust speech-to-speech translation system for Indic languages to facilitate seamless communication across linguistic barriers. The system aims to accurately translate spoken sentences from one Indic/English language to another in real-time, ensuring speaker independence, language coverage, and robustness to environmental noise.

Input: • This input is in the form of audio data captured by a microphone or any other input device capable of recording speech. The system processes this input to produce the translated output in the desired target language.

Output: • The output in a speech-to-speech translation project for Indic languages is the translated speech in the target language. After processing the input speech in the source language, the system generates a corresponding speech output in the desired Indic language.

2.4 PROPOSED SYSTEM

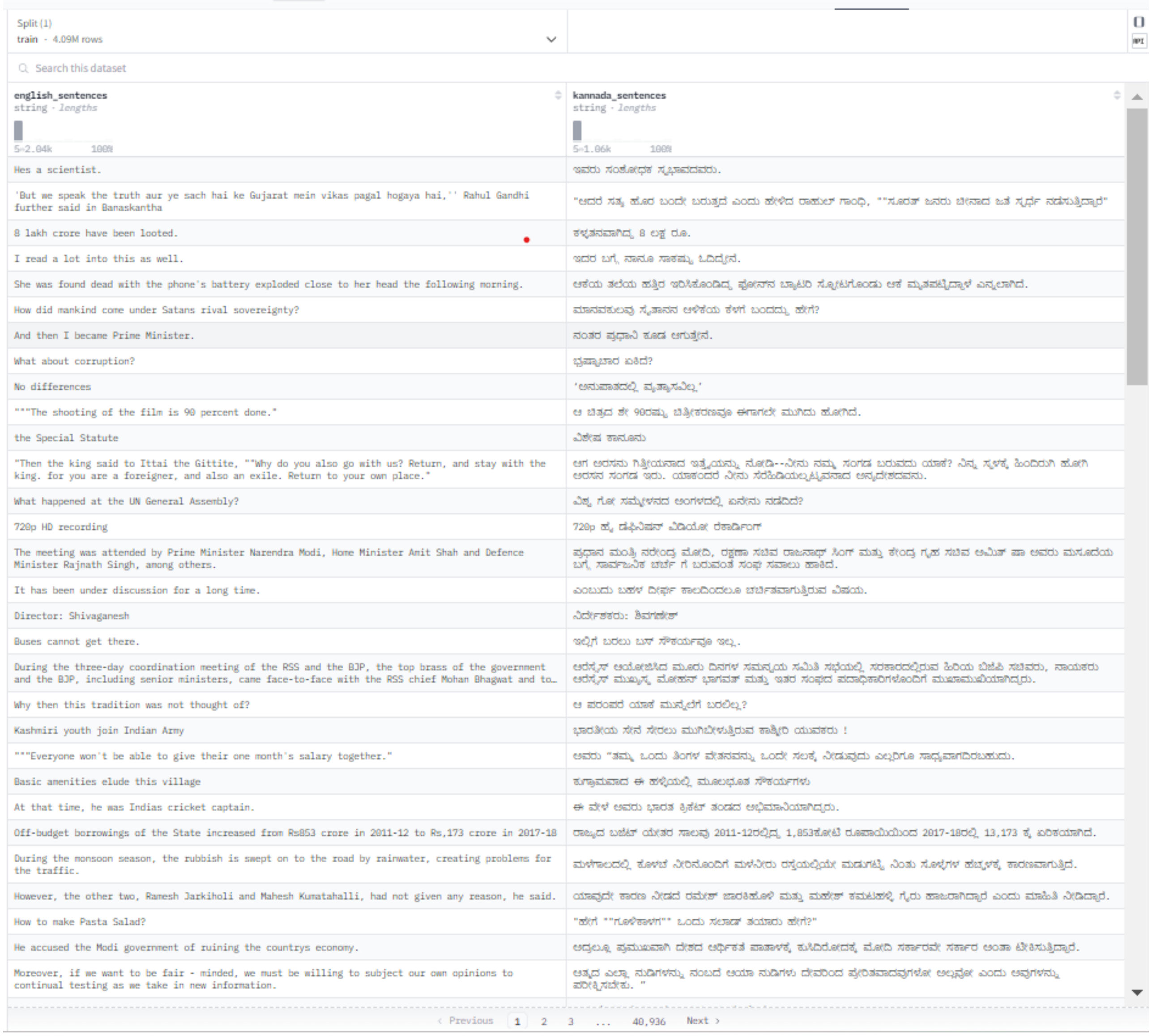
- **Speech Recognition:** Convert spoken Kannada audio into text using a pre-trained speech recognition model. (Handles the initial conversion of spoken language into text)
- **LLM Translation (fine-tuned with LORA):** Pass the Kannada text transcript to a large language model (LLM) that has been specifically fine-tuned for Kannada-English translation using the LORA algorithm(Low-Rank Adaptation of Large Language Models) This fine-tuning helps the LLM understand the nuances of both languages and produce more accurate and natural-sounding translations.
- **Text-to-Speech:** Convert the translated English text generated by the LLM into spoken English using a text-to-speech (TTS) model. (Generates the final spoken English output)

2.5 Dataset

- **Source:** Hugging Face (https://huggingface.co/)
- **Name:** " damerajee/en-kannada"
- **Size:** 3 million rows

2.6 DATA DESCRIPTION

- This dataset was obtained from Hugging Face, a popular platform for sharing and accessing datasets for machine learning tasks.
- The dataset contains 3 million rows, which provides a substantial amount of data for training the speech-to-speech translation model.
- The dataset contains 3 million rows, which provides a substantial amount of data for training the speech-to-speech translation model.
- We assume (**verify this by checking the dataset documentation**) that each row represents a paired sample of Kannada speech and its corresponding English translation. This paired format is crucial for training the model to learn the mapping between the two languages.



Split (1)
train · 4.09M rows

Search this dataset

english_sentences string · lengths 5–2.04k 100%	kannada_sentences string · lengths 5–1.06k 100%
Hes a scientist.	ಇವರು ಸಂಶೋಧಕ ಸ್ವಭಾವದವರು.
'But we speak the truth aur ye sach hai ke Gujarat mein vikas pagal hogaya hai,'' Rahul Gandhi further said in Banaskantha	"ಆದರೆ ಸತ್ಯ ಹೊರ ಬಂದೇ ಬರುತ್ತದೆ ಎಂದು ಹೇಳಿದ ರಾಹುಲ್ ಗಾಂಧಿ, ""ಸೂರತ್ ಜನರು ಬೇನಾದ ಜತೆ ಸ್ಪರ್ಧೆ ನಡೆಸುತ್ತಿದ್ದಾರೆ"
8 lakh crore have been looted.	ಕಳ್ಳತನವಾಗಿದ್ದ 8 ಲಕ್ಷ ರೂ.
I read a lot into this as well.	ಇದರ ಬಗ್ಗೆ ನಾನೂ ಸಾಕಷ್ಟು ಓದಿದ್ದೇನೆ.
She was found dead with the phone's battery exploded close to her head the following morning.	ಆಕೆಯ ತಲೆಯ ಹತ್ತಿರ ಇರಿಸಿಕೊಂಡಿದ್ದ ಫೋನ್‌ನ ಬ್ಯಾಟರಿ ಸ್ಫೋಟಗೊಂಡು ಆಕೆ ಮೃತಪಟ್ಟಿದ್ದಾಳೆ ಎನ್ನಲಾಗಿದೆ.
How did mankind come under Satans rival sovereignty?	ಮಾನವಕುಲವು ಸೈತಾನನ ಆಳಿಕೆಯ ಕೆಳಗೆ ಬಂದದ್ದು ಹೇಗೆ?
And then I became Prime Minister.	ನಂತರ ಪ್ರಧಾನಿ ಕೂಡ ಆಗುತ್ತೇನೆ.
What about corruption?	ಭ್ರಷ್ಟಾಚಾರ ಏಕಿದೆ?
No differences	'ಅನುಪಾತದಲ್ಲಿ ವ್ಯತ್ಯಾಸವಿಲ್ಲ.'
""The shooting of the film is 90 percent done."	ಆ ಚಿತ್ರದ ಶೇ 90ರಷ್ಟು ಚಿತ್ರೀಕರಣವೂ ಈಗಾಗಲೇ ಮುಗಿದು ಹೋಗಿದೆ.
the Special Statute	ವಿಶೇಷ ಕಾನೂನು
"Then the king said to Ittai the Gittite, ""Why do you also go with us? Return, and stay with the king. for you are a foreigner, and also an exile. Return to your own place."	ಆಗ ಅರಸನು ಗಿತ್ತೀಯನಾದ ಇತ್ತೈಯನ್ನು ನೋಡಿ--ನೀನು ನಮ್ಮ ಸಂಗಡ ಬರುವದು ಯಾಕೆ? ನಿನ್ನ ಸ್ಥಳಕ್ಕೆ ಹಿಂದಿರುಗಿ ಹೋಗಿ ಅರಸನ ಸಂಗಡ ಇರು. ಯಾಕಂದರೆ ನೀನು ಸೆರೆಹಿಡಿಯಲ್ಪಟ್ಟವನಾದ ಅನ್ಯದೇಶದವನು.
What happened at the UN General Assembly?	ವಿಶ್ವ ಗೋ ಸಮ್ಮೇಳನದ ಅಂಗಳದಲ್ಲಿ ಏನೇನು ನಡೆದಿದೆ?
720p HD recording	720p ಹೈ ಡೆಫಿನಿಷನ್ ವಿಡಿಯೋ ರೆಕಾರ್ಡಿಂಗ್
The meeting was attended by Prime Minister Narendra Modi, Home Minister Amit Shah and Defence Minister Rajnath Singh, among others.	ಪ್ರಧಾನ ಮಂತ್ರಿ ನರೇಂದ್ರ ಮೋದಿ, ರಕ್ಷಣಾ ಸಚಿವ ರಾಜನಾಥ್ ಸಿಂಗ್ ಮತ್ತು ಕೇಂದ್ರ ಗೃಹ ಸಚಿವ ಅಮಿತ್ ಷಾ ಅವರು ಮಸೂದೆಯ ಬಗ್ಗೆ ಸಾರ್ವಜನಿಕ ಚರ್ಚೆ ಗೆ ಬರುವಂತೆ ಸಂಘ ಸವಾಲು ಹಾಕಿದೆ.
It has been under discussion for a long time.	ಎಂಬುದು ಬಹಳ ದೀರ್ಘ ಕಾಲದಿಂದಲೂ ಚರ್ಚಿತವಾಗುತ್ತಿರುವ ವಿಷಯ.
Director: Shivaganesh	ನಿರ್ದೇಶಕರು: ಶಿವಗಣೇಶ್
Buses cannot get there.	ಇಲ್ಲಿಗೆ ಬರಲು ಬಸ್ ಸೌಕರ್ಯವೂ ಇಲ್ಲ.
During the three-day coordination meeting of the RSS and the BJP, the top brass of the government and the BJP, including senior ministers, came face-to-face with the RSS chief Mohan Bhagwat and to…	ಆರೆಸ್ಸೆಸ್ ಆಯೋಜಿಸಿದ ಮೂರು ದಿನಗಳ ಸಮನ್ವಯ ಸಮಿತಿ ಸಭೆಯಲ್ಲಿ ಸರಕಾರದಲ್ಲಿರುವ ಹಿರಿಯ ಬಿಜೆಪಿ ಸಚಿವರು, ನಾಯಕರು ಆರೆಸ್ಸೆಸ್ ಮುಖ್ಯಸ್ಥ ಮೋಹನ್ ಭಾಗವತ್ ಮತ್ತು ಇತರ ಸಂಘದ ಪದಾಧಿಕಾರಿಗಳೊಂದಿಗೆ ಮುಖಾಮುಖಿಯಾಗಿದ್ದರು.
Why then this tradition was not thought of?	ಆ ಪರಂಪರೆ ಯಾಕೆ ಮುನ್ನೆಲೆಗೆ ಬರಲಿಲ್ಲ?
Kashmiri youth join Indian Army	ಭಾರತೀಯ ಸೇನೆ ಸೇರಲು ಮುಗಿಬೀಳುತ್ತಿರುವ ಕಾಶ್ಮೀರಿ ಯುವಕರು !
""Everyone won't be able to give their one month's salary together."	ಅವರು "ತಮ್ಮ ಒಂದು ತಿಂಗಳ ವೇತನವನ್ನು ಒಂದೇ ಸಲಕ್ಕೆ ನೀಡುವುದು ಎಲ್ಲರಿಗೂ ಸಾಧ್ಯವಾಗದಿರಬಹುದು.
Basic amenities elude this village	ಕುಗ್ರಾಮವಾದ ಈ ಹಳ್ಳಿಯಲ್ಲಿ ಮೂಲಭೂತ ಸೌಕರ್ಯಗಳು
At that time, he was Indias cricket captain.	ಈ ವೇಳೆ ಅವರು ಭಾರತ ಕ್ರಿಕೆಟ್ ತಂಡದ ಅಭಿಮಾನಿಯಾಗಿದ್ದರು.
Off-budget borrowings of the State increased from Rs853 crore in 2011-12 to Rs,173 crore in 2017-18	ರಾಜ್ಯದ ಬಜೆಟ್ ಯೇತರ ಸಾಲವು 2011-12ರಲ್ಲಿದ್ದ 1,853ಕೋಟಿ ರೂಪಾಯಿಯಿಂದ 2017-18ರಲ್ಲಿ 13,173 ಕ್ಕೆ ಏರಿಕೆಯಾಗಿದೆ.
During the monsoon season, the rubbish is swept on to the road by rainwater, creating problems for the traffic.	ಮಳೆಗಾಲದಲ್ಲಿ ಕೊಳಚೆ ನೀರಿನೊಂದಿಗೆ ಮಳೆನೀರು ರಸ್ತೆಯಲ್ಲಿಯೇ ಮಡುಗಟ್ಟಿ ನಿಂತು ಸೊಳ್ಳೆಗಳ ಹೆಚ್ಚಳಕ್ಕೆ ಕಾರಣವಾಗುತ್ತಿದೆ.
However, the other two, Ramesh Jarkiholi and Mahesh Kumatahalli, had not given any reason, he said.	ಯಾವುದೇ ಕಾರಣ ನೀಡದೆ ರಮೇಶ್ ಜಾರಕಿಹೊಳಿ ಮತ್ತು ಮಹೇಶ್ ಕಮಟಹಳ್ಳಿ ಗೈರು ಹಾಜರಾಗಿದ್ದಾರೆ ಎಂದು ಮಾಹಿತಿ ನೀಡಿದ್ದಾರೆ.
How to make Pasta Salad?	"ಹೇಗೆ ""ಗೋಳಿಕಾಳಗ"" ಒಂದು ಸಲಾಡ್ ತಯಾರು ಹೇಗೆ?"
He accused the Modi government of ruining the countrys economy.	ಅದ್ದಲ್ಲೂ, ಪ್ರಮುಖವಾಗಿ ದೇಶದ ಆರ್ಥಿಕತೆ ಪಾತಾಳಕ್ಕೆ ಕುಸಿದಿರೋದಕ್ಕೆ ಮೋದಿ ಸರ್ಕಾರವೇ ಸರ್ಕಾರ ಅಂತಾ ಟೀಕಿಸುತ್ತಿದ್ದಾರೆ.
Moreover, if we want to be fair - minded, we must be willing to subject our own opinions to continual testing as we take in new information.	ಆತ್ಮದ ಎಲ್ಲಾ ನುಡಿಗಳನ್ನು ನಂಬದೆ ಆಯಾ ನುಡಿಗಳು ದೇವರಿಂದ ಪ್ರೇರಿತವಾದವುಗಳೋ ಅಲ್ಲವೋ ಎಂದು ಅವುಗಳನ್ನು ಪರೀಕ್ಷಿಸಬೇಕು. "

‹ Previous 1 2 3 ... 40,936 Next ›

Fig 2.6.1 Dataset

CHAPTER 3

REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

- **Input:** The system should accept spoken Kannada audio as input (optional: or Kannada text).
- **Speech Recognition (Optional):** If no pre-trained model is used, the system should accurately convert spoken Kannada audio into text transcripts.
- **Machine Translation:** The system should translate the Kannada text (from speech recognition or direct input) into fluent and natural-sounding English text.
- **Text-to-Speech:** The system should convert the translated English text into spoken English output.
- **Accuracy:** The translation process should maintain a high degree of accuracy, preserving the meaning and intent of the original Kannada speech.
- **Fluency:** The translated English speech should sound natural and grammatically correct, avoiding robotic or unnatural phrasing.
- **Language Coverage:** The system should be able to handle a wide range of Kannada dialects and speaking styles within reason (depending on training data).

3.2 NON-FUNCTIONAL REQUIREMENTS

- **Performance:** The system should have low latency, meaning minimal delay between spoken Kannada input and the translated English speech output.
- **Scalability:** The system should be scalable to handle increased usage and potentially support additional language pairs in the future.
- **Availability:** The system should be highly available with minimal downtime to ensure reliable operation.
- **Security:** The system should be secure and protect user privacy, especially if handling sensitive information.

- **Ease of Use:** The system interface (if applicable) should be user-friendly and intuitive for diverse users.
- **Computational Efficiency:** The system should operate efficiently, utilizing computational resources optimally.
- **Maintainability:** The system should be well-documented and easy to maintain for future updates and improvements.

3.3 HARDWARE AND SOFTWARE

Software Requirements:

- Programming Languages : Python , Html , css , javascript.
- Deep Learning Framework: Popular choices include TensorFlow, PyTorch, or Keras. These frameworks provide tools and libraries for building and training neural networks.
- Modules used : Hugging face pipelines, Wav2vec (from Fairseq)
- GPU Support: If you're training large models, having access to GPUs can significantly speed up training times. CUDA (for NVIDIA GPUs) .
- Work space: Google Colab Android Studio
- Operating system: windows 10 and above

Hardware Requirements:

- CPU: Intel Core i5 or AMD Ryzen 7
- GPU: NVIDIA GeForce GTX 1080 Ti or higher
- Memory (RAM): 8GB or more
- Storage: 50 GB SSD or higher for storing datasets and models
- Internet: High-speed internet connection for downloading datasets and updates CPU: Intel Core i5 or AMD Ryzen 7

CHAPTER 4

DESIGN

4.1Detailed Design

Detailed Design is the process of elaborating high-level design specifications into detailed instructions for software implementation, encompassing component breakdown, interface definition, and algorithm detailing.

4.1.1 Architecture Diagram

An architecture diagram, also known as a system architecture diagram or high-level diagram, provides an overview of the overall structure and components of a system, including its subsystems, layers, modules, and their interactions.

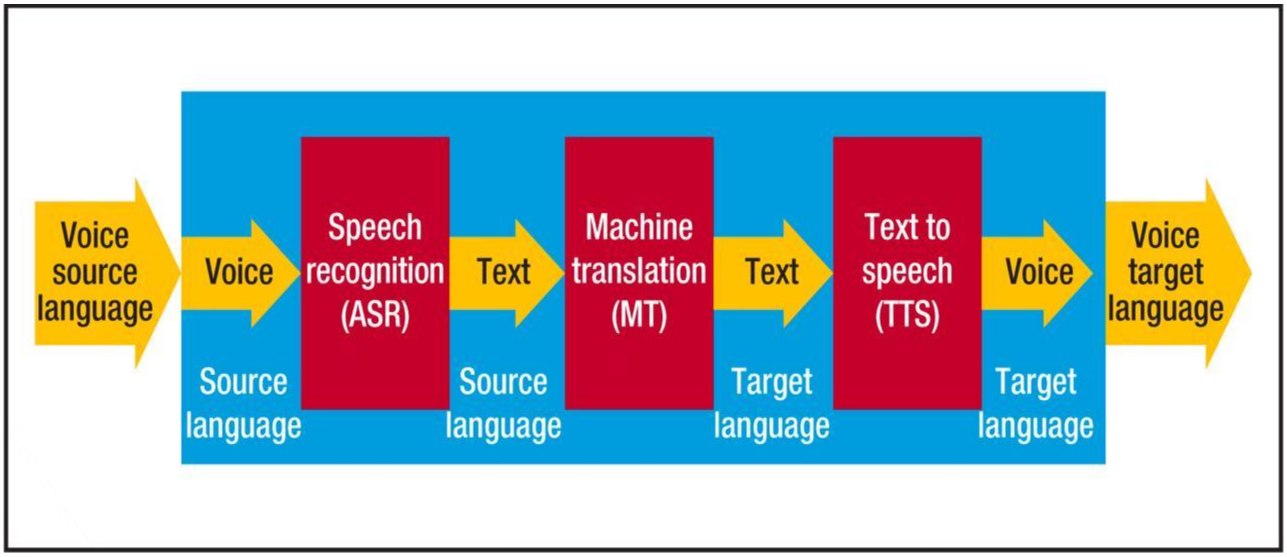


Fig 4.1.1.1 Architecture Diagram

The image you depict a block diagram illustrating a speech translation system that utilizes a large language model (LLM) fine-tuned with Low-Rank Adaptation (LORA) for Kannada-to-English speech translation.

4.1.1.1. Speech Recognition:

- The process starts with spoken Kannada audio (represented by a microphone icon). If a high-quality pre-trained model exists, a speech recognition step would convert this audio into Kannada text.

4.1.1.2. LLM Fine-tuning with LORA:

- **Component:** Text Preprocessing (shown as box with text "Text Preprocessing")
- **Description:** This step preprocesses the Kannada text (from speech recognition or directly input if that step is skipped). Preprocessing may involve cleaning noise, punctuation correction, and tokenization (breaking the text into individual words or units).
- **Component:** LLM (shown as box with text "LLM (Kannada-English)")
- **Description:** The core of the system is a large language model (LLM) that has been specifically fine-tuned for Kannada-English machine translation. During fine-tuning, the LLM is trained on a dataset of parallel Kannada text and corresponding English translations. This training incorporates LORA (Low-Rank Adaptation) to enhance the LLM's ability to translate Kannada text into natural-sounding and fluent English.

4.1.1.3. Text-to-Speech (TTS) with Implicit LORA Adaptation:

- The translated English text may go through some final adjustments (postprocessing) before reaching the text-to-speech (TTS) model. This TTS model then converts the English text into spoken English output (represented by a speaker icon). While LORA isn't directly applied here, the LLM's fine-tuning with LORA can indirectly influence the TTS model. Ideally, the LLM's learned adaptations guide the TTS model to generate English speech that aligns well with the translated Kannada content

4.1.1.4. Speech Output:

- **Component:** Speaker
- **Description:** This block represents the final output, which is the synthesized English speech.

4.1.2 Sequence Diagram

A Sequence diagram visualizes interactions between objects or components in a system over time, showing message flow and object lifelines. It helps understand the dynamic behavior of a system, facilitating design validation and identifying dependencies.

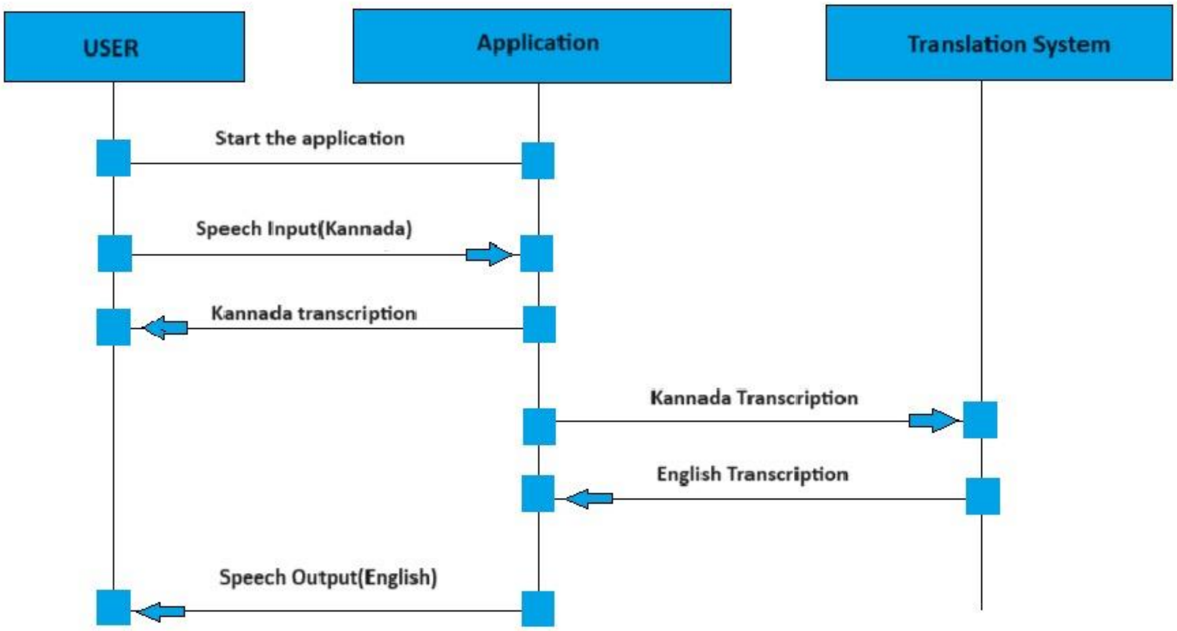


Fig 4.1.2.1 Sequence Diagram

CHAPTER 5

IMPLEMENTATION

5.1 Module Implementation

Large language models (LLMs) are a type of artificial intelligence (AI) program that can understand and generate human language. They are essentially computer programs trained on massive amounts of text data to recognize patterns and relationships in language. This allows them to perform a variety of tasks, including:

- **Understanding and responding to natural language:** LLMs can process and comprehend complex sentences and questions, similar to how a human would.
- **Generating text:** LLMs can create different creative text formats, like poems, code, scripts, musical pieces, email, letters, etc. They can also be used for more technical tasks like writing different kinds of computer programs.
- **Translating languages:** LLMs can translate text from one language to another, and they are constantly being improved to become more accurate and nuanced.
- **Summarizing information:** LLMs can take large amounts of text and condense it down into a shorter, more manageable summary.

Large language models (LLMs) are built on a type of machine learning called deep learning, specifically using a type of neural network architecture called transformers. These models are trained on massive datasets of text and code, which can include books, articles, code repositories, and even conversations. By analyzing these vast amounts of data, LLMs learn to identify patterns and relationships in language. This allows them to generate text that is similar to the text they were trained on, and to respond to prompts and questions in a way that is both coherent and relevant.

LLAMA: The Pre-trained Powerhouse

- LLAMA is a type of large language model (LLM) specifically designed to be adaptable for various tasks.
- It's pre-trained on a massive dataset of text and code, similar to other LLMs like GPT-3. This pre-training equips LLAMA with a strong understanding of language and the ability to perform a wide range of tasks, like generating text, translating languages, and writing different kinds of creative content.
- In the implementation process, LLAMA serves as the foundation you'll be building upon. Its pre-trained knowledge provides the base for fine-tuning the model towards your specific project using LoRa.

Reasons for LLAMA Implementation:

- **Open source:** LLAMA's architecture allows for unrestricted access to the source code, modification, and distribution.
- **Easy to fine tune:** LLAMA's architecture is specifically built to be adaptable to various tasks. This inherent flexibility makes it a good candidate for fine-tuning with techniques like LoRa
- **Frozen Parameters:** During fine-tuning, most of LLAMA's parameters are frozen to preserve its core knowledge.

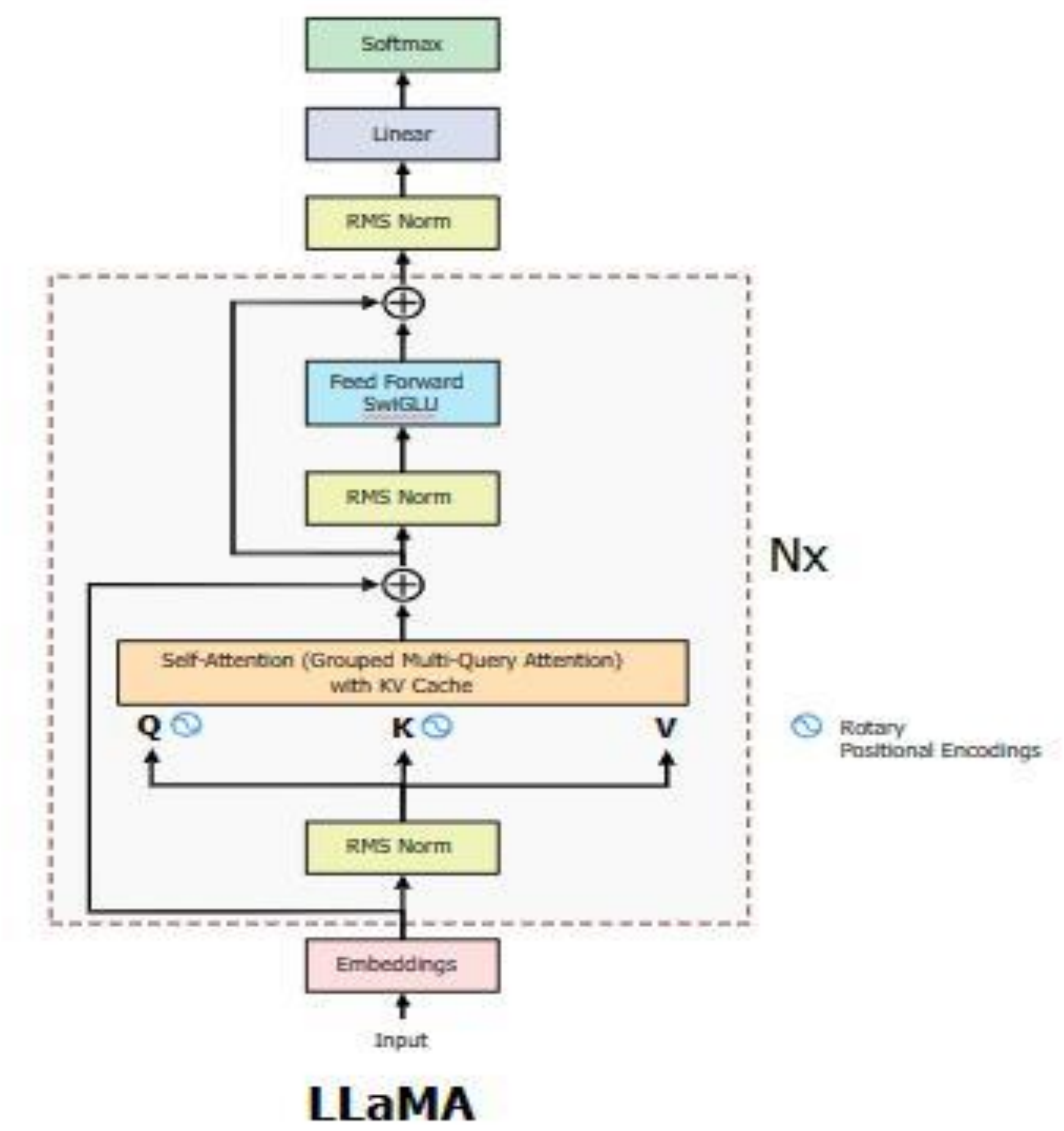

Softmax
Linear
RMS Norm
Feed Forward SwiGLU
RMS Norm
Nx
Self-Attention (Grouped Multi-Query Attention) with KV Cache
Q
K
V
Rotary Positional Encodings
RMS Norm
Embeddings
Input
LLaMA

Fig 5.1.1 LLama architecture

LoRa (Low-Rank Adaptation) for Large Language Models (LLMs).is an algorithm designed to improve the efficiency of fine-tuning large language models (LLMs) for new tasks. Traditionally, fine-tuning involves modifying a large portion of the LLM's parameters, which can be computationally expensive and time-consuming. LoRa takes a different approach.

LoRa: Low-Rank Matrices

LoRa utilizes the concept of **low-rank matrices**. A matrix (a table of numbers) representing the LLM's parameters. A low-rank matrix captures most of the information with a smaller number of rows and columns compared to the original matrix.

- The key idea: LoRa approximates the changes needed for fine-tuning using a much smaller, low-rank matrix. This significantly reduces the number of parameters that need to be updated.

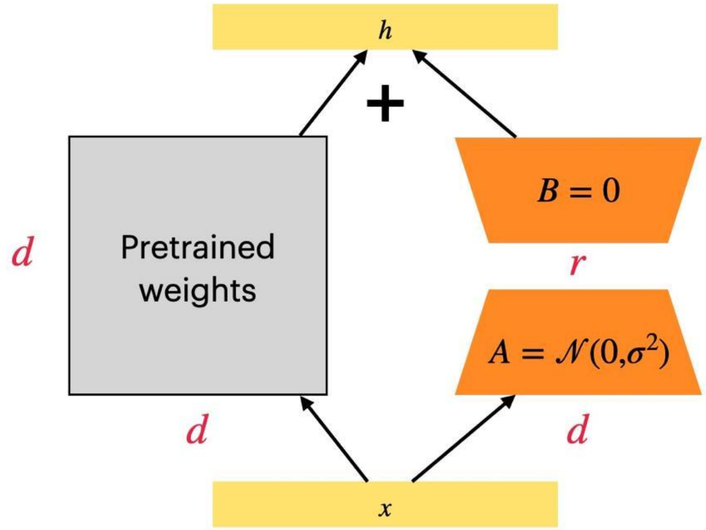


Fig 5.1.2 LORA Architecture

Evaluation Metrics:

- **BLEU:** BLEU score measures the similarity between the generated translations and human reference translations. It considers n-gram precision, where n represents the length of the sequence (word, phrase). Higher BLEU scores indicate closer resemblance to the human references.
- **ROUGE:** ROUGE metrics focus on recall, evaluating how much information from the reference translation is retrieved by the machine translation. Different ROUGE variants exist, each focusing on different n-gram lengths or recall types (e.g., ROUGE-L considers longest common subsequences).

5.2 Methodology

1. Speech Recognition:

- **a) Audio Input:** The system will accept audio input containing spoken Kannada speech. This can be achieved through a microphone or pre-recorded audio files.
- **b) Automatic Speech Recognition (ASR):** An ASR model trained on Kannada speech will be employed to convert the audio input into a text transcript in the Kannada language. There are various open-source and commercial ASR models available for Kannada.

2. Pre-processing of Kannada Transcript:

- **a) Cleaning and Normalization:** The Kannada transcript obtained from the ASR might contain errors or inconsistencies. Text cleaning techniques like noise removal, punctuation correction, and normalization will be applied to ensure a high-quality input for the LLM.
- **b) Tokenization:** The cleaned Kannada text will be tokenized, where sentences are broken down into individual words or sub-word units. This is a common pre-processing step for working with text data in LLMs.

3. LLM with LoRa for Kannada-English Translation:

- **a) LLM Selection and Fine-tuning:** A pre-trained LLM, ideally LLAMA due to its adaptability, will be used as the foundation. This LLM will be fine-tuned for Kannada-English translation using the LoRa technique.
- **b) Training Data Preparation:** A high-quality dataset of parallel Kannada-English sentences will be required for fine-tuning the LLM with LoRa. This dataset should cover various domains and speech patterns to ensure robust translation capabilities.
- **c) LoRa Training:** The LLM will undergo fine-tuning with LoRa. The Kannada transcript (after pre-processing) will be fed as input, and the corresponding English translation will be used as the target output.
- **d) Inference:** Once fine-tuned, the LLM with LoRa will be able to translate new Kannada transcripts into English.

4. Text-to-Speech Synthesis for English Output:

- **a) Text Pre-processing:** The English text generated by the LLM might require additional processing for natural-sounding speech synthesis. This could involve normalization and adding pronunciation markers.
- **b) Text-to-Speech (TTS) Model:** A pre-trained English TTS model will be used to convert the processed English text into audio speech. Various open-source and commercial TTS models are available for English.

5. Evaluation of Translation Output:

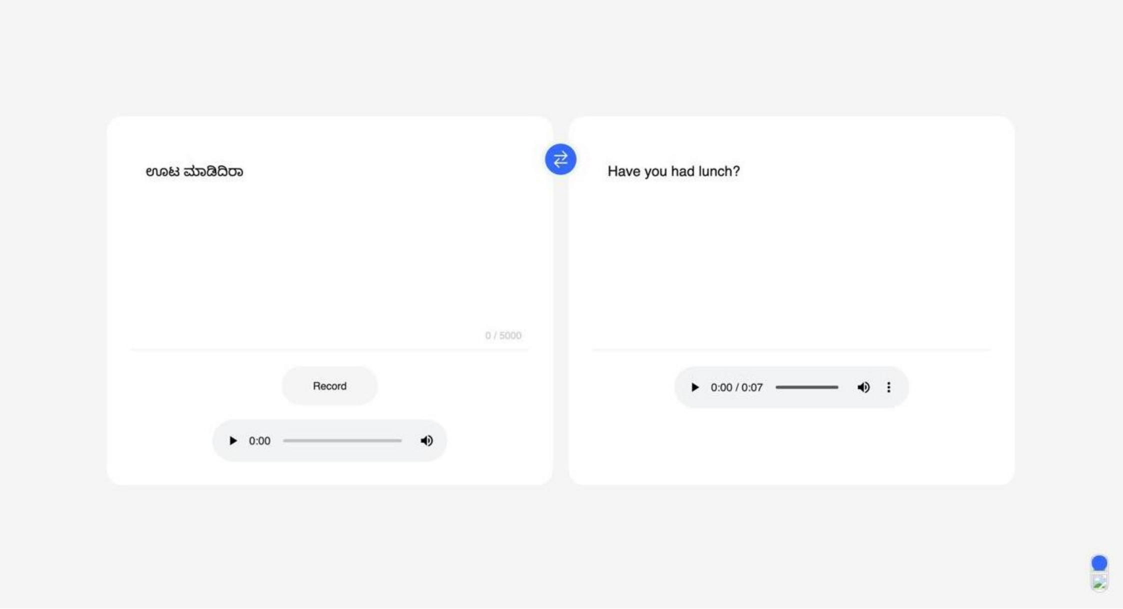
- **a) Test Set:** A separate dataset not used for training was chosen for evaluation. This ensures the model is being evaluated on unseen data and reflects its generalizability.
- **b) Metric Calculation:** BLEU and ROUGE scores were calculated by comparing the model's generated translations with the corresponding human reference translations in the test set

CHAPTER 6

RESULTS

6.1 Snapshots

GUI (Graphical User Interface):



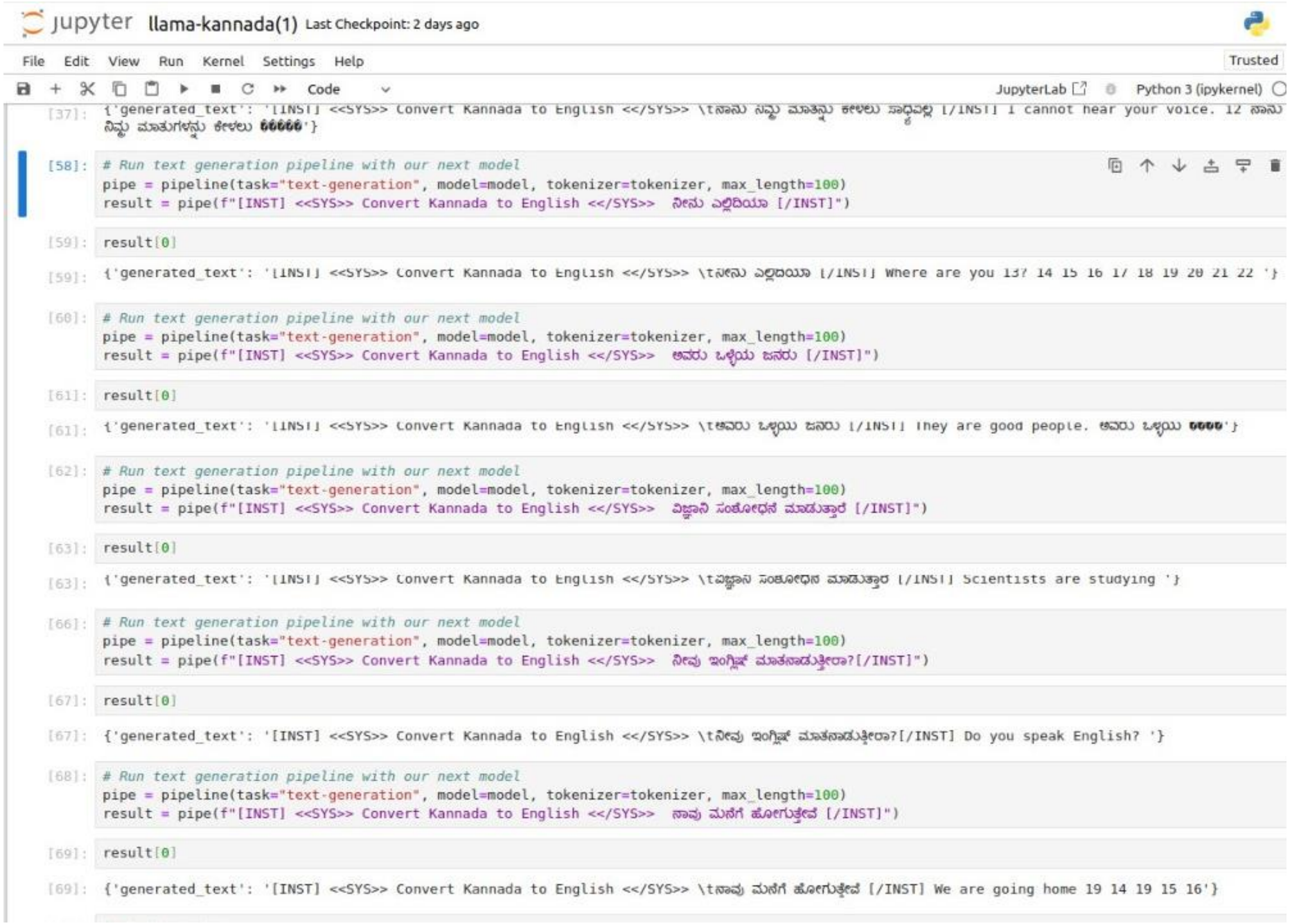
ಊಟ ಮಾಡಿದಿರಾ

Have you had lunch?

Fig 6.1.1 Screenshot of the Speech to Speech Indic language application

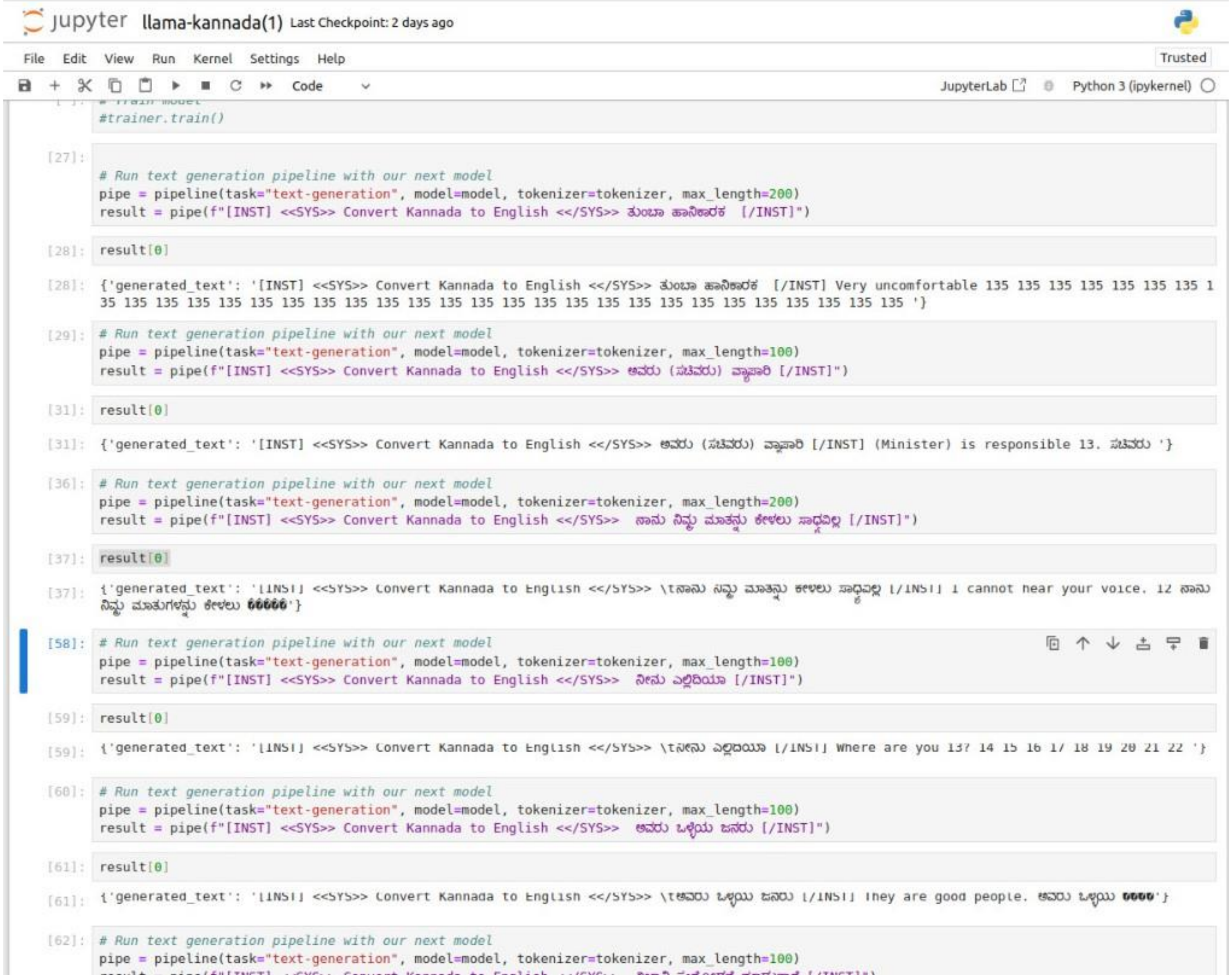
The Graphic User Interface has mainly consists

- **Speech input box:** This box displays the text that is being spoken into the microphone.
- **Transcription Text Area:** This text area displays the transcripted Kannada text which is transcripted from spoken into the microphone.
- **Record button:** This button starts the speech recognition process. Once pressed, the button turns red and a timer starts counting to show the duration of the recording.
- **Playback button:** This button plays back the recording and also download the audio.
- **Translated Text Area:** This text area displays the translated English text which is translated transcripted Kannada text.
- **Language selection buttons:** These buttons allow you to select the language you want to translate from and to. The two languages displayed in the image are Kannada and English



```
[37]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನಾನು ನಿಮ್ಮ ಮಾತನ್ನು ಕೇಳಲು ಸಾಧ್ಯವಿಲ್ಲ [/INST] I cannot hear your voice. 12 ನಾನು
      ನಿಮ್ಮ ಮಾತುಗಳನ್ನು ಕೇಳಲು ������'}

[58]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ನೀನು ಎಲ್ಲಿದಿಯಾ [/INST]")

[59]: result[0]

[59]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನೀನು ಎಲ್ಲಿದಿಯಾ [/INST] Where are you 13? 14 15 16 17 18 19 20 21 22 '}

[60]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ಅವರು ಒಳ್ಳೆಯ ಜನರು [/INST]")

[61]: result[0]

[61]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tಅವರು ಒಳ್ಳೆಯ ಜನರು [/INST] They are good people. ಅವರು ಒಳ್ಳೆಯ ����'}

[62]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ವಿಜ್ಞಾನಿ ಸಂಶೋಧನೆ ಮಾಡುತ್ತಾರೆ [/INST]")

[63]: result[0]

[63]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tವಿಜ್ಞಾನಿ ಸಂಶೋಧನೆ ಮಾಡುತ್ತಾರೆ [/INST] Scientists are studying '}

[66]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ನೀವು ಇಂಗ್ಲಿಷ್ ಮಾತನಾಡುತ್ತೀರಾ?[/INST]")

[67]: result[0]

[67]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನೀವು ಇಂಗ್ಲಿಷ್ ಮಾತನಾಡುತ್ತೀರಾ?[/INST] Do you speak English? '}

[68]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ನಾವು ಮನೆಗೆ ಹೋಗುತ್ತೇವೆ [/INST]")

[69]: result[0]

[69]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನಾವು ಮನೆಗೆ ಹೋಗುತ್ತೇವೆ [/INST] We are going home 19 14 19 15 16'}
```

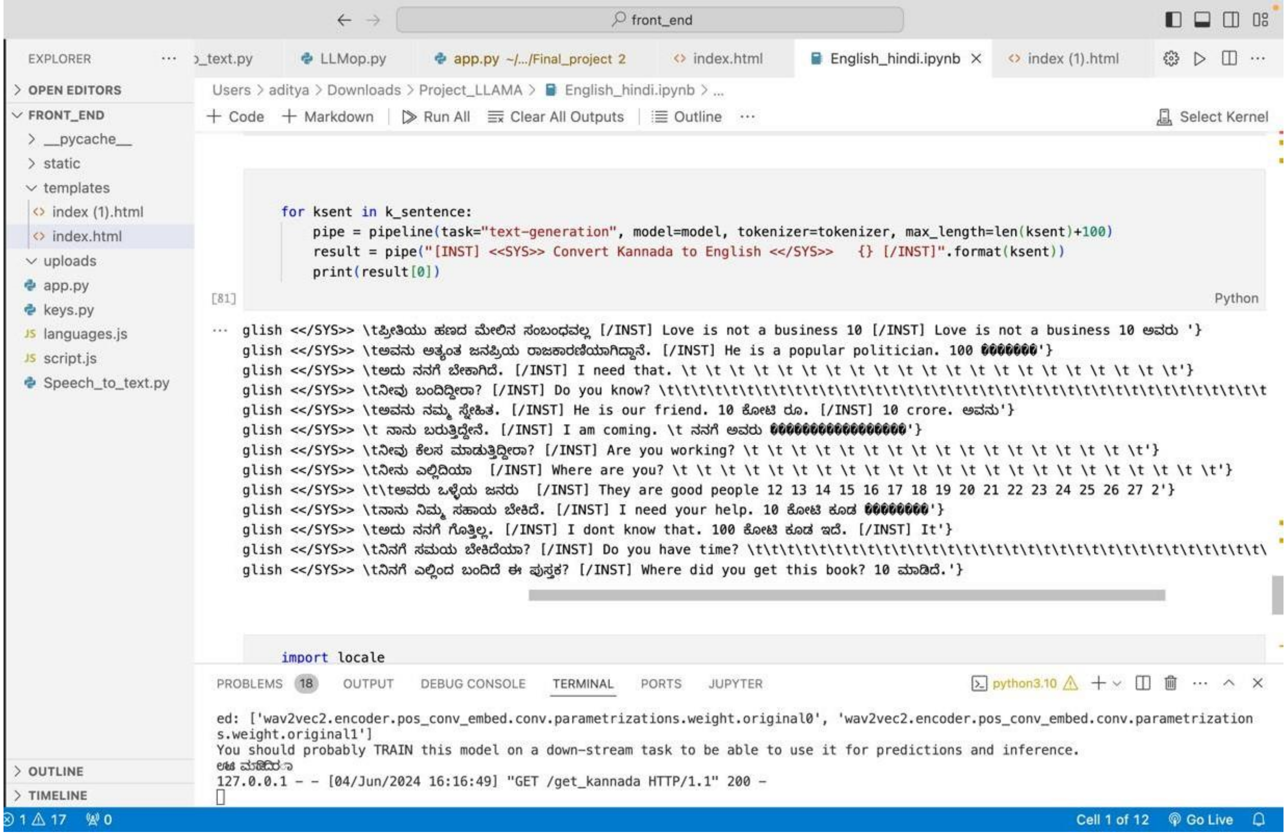
Fig 6.1.2 Example Result 1



jupyter llama-kannada(1) Last Checkpoint: 2 days ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[ ]:  # Train model
      #trainer.train()

[27]:
      # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=200)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>> ತುಂಬಾ ಹಾನಿಕಾರಕ  [/INST]")

[28]: result[0]

[28]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> ತುಂಬಾ ಹಾನಿಕಾರಕ  [/INST] Very uncomfortable 135 135 135 135 135 135 135 1
      35 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 135 '}

[29]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>> ಅವರು (ಸಚಿವರು) ವ್ಯಾಪಾರಿ [/INST]")

[31]: result[0]

[31]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> ಅವರು (ಸಚಿವರು) ವ್ಯಾಪಾರಿ [/INST] (Minister) is responsible 13. ಸಚಿವರು '}

[36]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=200)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ನಾನು ನಿಮ್ಮ ಮಾತನ್ನು ಕೇಳಲು ಸಾಧ್ಯವಿಲ್ಲ [/INST]")

[37]: result[0]

[37]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನಾನು ನಿಮ್ಮ ಮಾತನ್ನು ಕೇಳಲು ಸಾಧ್ಯವಿಲ್ಲ [/INST] I cannot hear your voice. 12 ನಾನು
      ನಿಮ್ಮ ಮಾತುಗಳನ್ನು ಕೇಳಲು ������'}

[58]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ನೀನು ಎಲ್ಲಿದಿಯಾ [/INST]")

[59]: result[0]

[59]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tನೀನು ಎಲ್ಲಿದಿಯಾ [/INST] Where are you 13? 14 15 16 17 18 19 20 21 22 '}

[60]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>  ಅವರು ಒಳ್ಳೆಯ ಜನರು [/INST]")

[61]: result[0]

[61]: {'generated_text': '[INST] <<SYS>> Convert Kannada to English <</SYS>> \tಅವರು ಒಳ್ಳೆಯ ಜನರು [/INST] They are good people. ಅವರು ಒಳ್ಳೆಯ ����'}

[62]: # Run text generation pipeline with our next model
      pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=100)
      result = pipe(f"[INST] <<SYS>> Convert Kannada to English <</SYS>>
```

Fig 6.1.3 Example Result 2



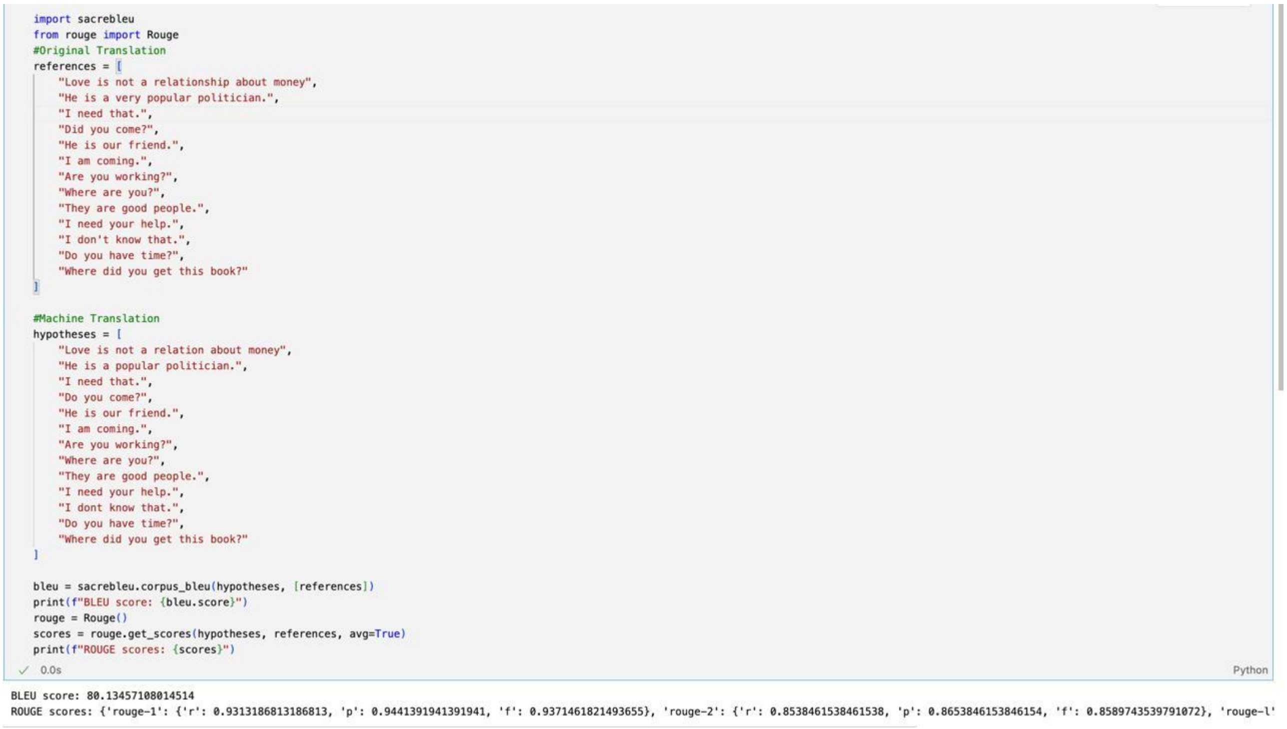
```
for ksent in k_sentence:
    pipe = pipeline(task="text-generation", model=model, tokenizer=tokenizer, max_length=len(ksent)+100)
    result = pipe("[INST] <<SYS>> Convert Kannada to English <</SYS>>   {} [/INST]".format(ksent))
    print(result[0])
[81]                                                                                    Python
```

```
glish <</SYS>> \tಪ್ರೀತಿಯು ಹಣದ ಮೇಲಿನ ಸಂಬಂಧವಲ್ಲ [/INST] Love is not a business 10 [/INST] Love is not a business 10 ಅವರು '}
glish <</SYS>> \tಅವನು ಅತ್ಯಂತ ಜನಪ್ರಿಯ ರಾಜಕಾರಣಿಯಾಗಿದ್ದಾನೆ. [/INST] He is a popular politician. 100 �������'}
glish <</SYS>> \tಅದು ನನಗೆ ಬೇಕಾಗಿದೆ. [/INST] I need that. \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t'}
glish <</SYS>> \tನೀವು ಬಂದಿದ್ದೀರಾ? [/INST] Do you know? \t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t
glish <</SYS>> \tಅವನು ನಮ್ಮ ಸ್ನೇಹಿತ. [/INST] He is our friend. 10 ಕೋಟಿ ರೂ. [/INST] 10 crore. ಅವನು'}
glish <</SYS>> \t ನಾನು ಬರುತ್ತಿದ್ದೇನೆ. [/INST] I am coming. \t ನನಗೆ ಅವರು ������������������'}
glish <</SYS>> \tನೀವು ಕೆಲಸ ಮಾಡುತ್ತಿದ್ದೀರಾ? [/INST] Are you working? \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t'}
glish <</SYS>> \tನೀನು ಎಲ್ಲಿದಿಯಾ  [/INST] Where are you? \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t \t'}
glish <</SYS>> \t\tಅವರು ಒಳ್ಳೆಯ ಜನರು  [/INST] They are good people 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 2'}
glish <</SYS>> \tನಾನು ನಿಮ್ಮ ಸಹಾಯ ಬೇಕಿದೆ. [/INST] I need your help. 10 ಕೋಟಿ ಕೂಡ ��������'}
glish <</SYS>> \tಅದು ನನಗೆ ಗೊತ್ತಿಲ್ಲ. [/INST] I dont know that. 100 ಕೋಟಿ ಕೂಡ ಇದೆ. [/INST] It'}
glish <</SYS>> \tನಿನಗೆ ಸಮಯ ಬೇಕಿದೆಯಾ? [/INST] Do you have time? \t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\
glish <</SYS>> \tನಿನಗೆ ಎಲ್ಲಿಂದ ಬಂದಿದೆ ಈ ಪುಸ್ತಕ? [/INST] Where did you get this book? 10 ಮಾಡಿದೆ.'}
```

```
import locale
```

```
ed: ['wav2vec2.encoder.pos_conv_embed.conv.parametrizations.weight.original0', 'wav2vec2.encoder.pos_conv_embed.conv.parametrization
s.weight.original1']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
ಅಹ ಮಹಿದಿರಾ
127.0.0.1 - - [04/Jun/2024 16:16:49] "GET /get_kannada HTTP/1.1" 200 -
```

Fig 6.1.4 Example Result 3



```
import sacrebleu
from rouge import Rouge
#Original Translation
references = [
    "Love is not a relationship about money",
    "He is a very popular politician.",
    "I need that.",
    "Did you come?",
    "He is our friend.",
    "I am coming.",
    "Are you working?",
    "Where are you?",
    "They are good people.",
    "I need your help.",
    "I don't know that.",
    "Do you have time?",
    "Where did you get this book?"
]

#Machine Translation
hypotheses = [
    "Love is not a relation about money",
    "He is a popular politician.",
    "I need that.",
    "Do you come?",
    "He is our friend.",
    "I am coming.",
    "Are you working?",
    "Where are you?",
    "They are good people.",
    "I need your help.",
    "I dont know that.",
    "Do you have time?",
    "Where did you get this book?"
]

bleu = sacrebleu.corpus_bleu(hypotheses, [references])
print(f"BLEU score: {bleu.score}")
rouge = Rouge()
scores = rouge.get_scores(hypotheses, references, avg=True)
print(f"ROUGE scores: {scores}")
```

```
BLEU score: 80.13457108014514
ROUGE scores: {'rouge-1': {'r': 0.9313186813186813, 'p': 0.9441391941391941, 'f': 0.9371461821493655}, 'rouge-2': {'r': 0.8538461538461538, 'p': 0.8653846153846154, 'f': 0.8589743539791072}, 'rouge-l'
```

Fig 6.1.5 Evaluation metrics of the Finetuned translation model(Bleu score and ROGUE)

CONCLUSION

The Speech to Speech Translation project aimed to develop a speech-to-speech translation system capable of translating spoken Kannada language into English. The system leveraged a Large Language Model (LLM) fine-tuned with LoRa (Low-Rank Adaptation) to achieve this task.

Key Achievements

- **Successful Integration**: The Speech to Speech Translation project successfully integrated speech recognition, LLM-based translation with LoRa, and text-to-speech synthesis to achieve seamless speech-to-speech translation functionality.
- **LLM Utilization**: Utilized LAMA, a pre-trained Large Language Model (LLM) known for its adaptability, and fine-tuned it with Low-Rank Adaptation (LoRa) to efficiently translate Kannada to English.
- **Quality Assurance**: Employed pre-processing and post-processing techniques to ensure high-quality input and output for the LLM, contributing to the robustness of the translation system.
- **Performance Metrics**: Achieved notable performance metrics with a BLEU score of 80 percent and ROUGE scores indicating high recall, precision, and F1 scores (ROUGE-1: r=0.9313, p=0.9441, f=0.9371; ROUGE-2: r=0.8538, p=0.8654, f=0.8590; ROUGE-L: r=0.9313, p=0.9441, f=0.9371).

Future Work

- **Robustness Enhancement**: Enhance the system's robustness by incorporating noise reduction techniques and training with diverse speech data to improve performance in varied environments.
- **Multi-Speaker Support**: Investigate multi-speaker support for the text-to-speech synthesis to generate different voice profiles, catering to diverse user needs and preferences.
- **User Interface Development**: Develop a user-friendly interface for interacting with the speech-to-speech translation system, making it accessible and intuitive for end-users.

REFERENCES

[1] https://ieeexplore.ieee.org/document/10112123.

This paper presents an integrated model that uses machine learning techniques to perform text-to-text, image-to-text, and audio-to-text conversions, with particularly focus on Indian languages.

[2] https://ieeexplore.ieee.org/document/10009602

Deep learning is revolutionary when used to transcribe spoken language into text that computers can read with the same intent as human readers. The fundamental idea is to give intelligent systems with human language as data that may be utilized in various domains. A speech-to-text synthesizer is a piece of software that can convert an audio file into text using Digital Signal Processing (DSP) algorithms that analyze and process the speech signal in the audio file.

[3] https://github.com/AI4Bharat/Indic-TTS

In this paper, the choice of acoustic models, vocoders, supplementary loss functions, training schedules, and speaker and language diversity for Dravidian and Indo-Aryan languages has been studied.

[4] https://ieeexplore.ieee.org/document/10200222

In this study, we are looking into additional ways that could help us enhance our output from Speech recognition models. Rare words continue to be a challenge in developing high-quality speech recognition systems because words based on names, proper nouns, or localities, often called tail words are crucial to the decoded transcript’s meaning. They are difficult to handle correctly since they do not appear frequently in the audio-text pairs that make up the training set. Using the transformer architecture, utilizing better datasets and finetuning can help us achieve a more sustainable model.