A NOVEL APPROACH TO MALAYALAM SPEECH-TO-TEXT AND TEXT-TO-ENGLISH TRANSLATION

A Project Report

submitted to

the APJ Abdul Kalam Technological University

in partial fulfillment of the requirements for the degree of

Bachelor of Technology

by

DENI THOMAS(VML20AD010)

JASHLIN S SIMON(VML20AD013)

MOHAMMED ZAIN RAFEEQUE(VML20AD017)

THAHA MUHAMMED YASEEN(VML20AD027)

under the supervision of

Ms. ANCY K SUNNY

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VIMAL JYOTHI ENGINEERING COLLEGE CHEMPERI

CHEMPERI P.O. - 670632, KANNUR, KERALA, INDIA

April 2024

VIMAL JYOTHI
INSTITUTIONS,CHEMPERI - KANNUR
CHEMPERI - KANNUR 0460 2212240

Accredited

NATIONAL BOARD OF ACCREDITATION
B.Tech - CSE & EEE

DEPT. OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the report entitled **A NOVEL APPROACH TO MALAYALAM SPEECH-TO-TEXT AND TEXT-TO-ENGLISH TRANSLATION** submitted by **DENI THOMAS** (VML20AD010), **JASHLIN S SIMON** (VML20AD013), **MOHAMMED ZAIN RAFEEQUE** (VML20AD017), and **THAHA MUHAMMED YASEEN** (VML20AD027) to the APJ Abdul Kalam Technological University in partial fulfillment of the B.Tech degree in **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE** is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Ms. ANCY K SUNNY
(Project Guide)
Assistant Professor
Dept.of CSE
Vimal Jyothi Engineering College
Chemperi

Dr. MANOJ V THOMAS
(Project Coordinator)
Program Coordinator of ADS
Dept.of CSE
Vimal Jyothi Engineering College
Chemperi

Ms. JISSIN KURIEN
(Project Coordinator)
Assistant Professor
Dept.of CSE
Vimal Jyothi Engineering College
Chemperi

Place : VJEC Chemperi
Date : 17-04-2024

Head of the department

(Office Seal)

DECLARATION

We hereby declare that the project report **A Novel Approach to Malayalam Speech-to-Text and Text-to-English Translation**, submitted for partial fulfillment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by us under supervision of **Ms. ANCY K SUNNY**.

This submission represents our ideas in our own words and where ideas or words of others have been included, we have adequately and accurately cited and referenced the original sources.

We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

DENI THOMAS
JASHLIN S SIMON
MOHAMMED ZAIN RAFEEQUE
THAHA MUHAMMED YASEEN

CHEMPERI
17-04-2024

ACKNOWLEDGEMENT

The successful presentation of the project on the topic **A NOVEL APPROACH TO MALAYALAM SPEECH-TO-TEXT AND TEXT-TO-ENGLISH TRANSLATION** would have been incomplete without the mention of people who made it possible and whose constant guidance crowned our effort into success.

We convey thanks to our project guide Ms. ANCY K SUNNY of Computer Science and Engineering Department for providing encouragement, constant support and guidance which was of a great help to complete this project successfully.

Last but not the least, we wish to thank our parents for financing our studies in this college as well as for constantly encouraging us to learn engineering. Their personal sacrifice in providing this opportunity to learn engineering is greatly acknowledged.

DENI THOMAS
JASHLIN S SIMON
MOHAMMED ZAIN RAEEQUE
THAHA MUHAMMED YASEEN

Abstract

This research introduces an innovative solution for the conversion of spoken and written Malayalam content into English, addressing the growing need for effective language translation tools. Leveraging advanced automatic speech recognition (ASR) models for Malayalam speech-to-text conversion and state-of-the-art neural machine translation (NMT) models for Malayalam text-to-English translation, our system offers a comprehensive approach to facilitate cross-lingual communication. These models are designed to accurately transcribe spoken Malayalam content into written text. Our system incorporates deep learning techniques to enhance the ASR model's accuracy, adapting to various Malayalam accents and dialects. The user-friendly interface enhances accessibility, while the context-aware translation mechanism preserves the nuances and meaning of the original content. This novel approach not only aids in breaking down language barriers but also has significant potential for applications in education, commerce, and governance, ultimately contributing to the advancement of language technology for less-resourced languages like Malayalam.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

1.1.1 General Background

This project proposes a novel approach to Malayalam speech-to-text (STT) and text-to-English (MT) translation that leverages recent advances in deep learning and machine translation. The goal of the project is to develop high-quality STT and MT systems that can address the limitations of existing systems, such as low accuracy, lack of domain adaptation, and lack of multimodality [5]. The project is expected to have a significant impact on language accessibility, economic development, and education. The developed STT and MT systems will make it easier for people who speak Malayalam to communicate with people who speak English. This will improve language accessibility and promote intercultural communication [6]. Additionally, the systems can be used to develop new educational resources and tools.

1.1.2 Problem Statement

This project aims to address these limitations by developing novel STT and MT systems that leverage recent advances in deep learning and machine translation. The new systems will be more accurate, adaptable, and multimodal than existing systems,

and they will have the potential to make a significant impact on language accessibility, economic development, and education. The proposed project aims to develop new STT and MT systems that can overcome these limitations and provide Malayalam speakers with more accurate and reliable translation tools.

1.1.3 Scope of the system

- The project will develop new STT and MT models that are more accurate than existing models.
- The project will develop new STT and MT models that can be adapted to specific domains, such as medical or legal translation.
- The project will develop new multimodal STT and MT models that can process both speech and text.
- Improve language accessibility and promote intercultural communication.

1.1.4 Objective

- Develop deep learning models for Malayalam STT and MT that are more accurate and robust than traditional models.
- Adapt the STT and MT models to specific domains, such as medical or legal translation, by training them on domain-specific data.
- Develop multimodal STT and MT models that can process both speech and text.
- Develop a real-world application that uses the developed STT and MT systems to translate Malayalam speech and text to English.

Chapter 2

Literature Survey

2.1 Unsupervised Neural Machine Translation for English to Kannada Using Pre-Trained Language Model [1]

In our increasingly interconnected global landscape, effective communication across language barriers is paramount. The field of Neural Machine Translation (NMT) has seen remarkable advancements, enabling seamless translation between numerous languages. This paper delves into an intriguing and practical dimension of NMT by proposing an unsupervised approach for translating English to Kannada, a Dravidian language predominantly spoken in the Indian state of Karnataka. Traditional supervised NMT systems often require vast parallel corpora, which can be scarce or non-existent for many language pairs, especially for languages with limited digital resources like Kannada. Our unsupervised NMT approach aims to address this challenge by harnessing the capabilities of pre-trained language models Unlike traditional NMT systems that rely on parallel corpora for training, our methodology harnesses the power of pre-trained language models, breaking free from the limitations of resource-intensive data collection and parallel corpora availability. By leveraging the transformative capabilities of unsupervised learning, we aim to pave the way for more

accessible, efficient, and adaptable machine translation systems, thus contributing to bridging the linguistic divide in our diverse world. This paper elucidates the architecture, challenges, and potential applications of our proposed unsupervised NMT approach for English to Kannada translation.

The methodology described in this paper outlines the approach used for developing an unsupervised Neural Machine Translation (UNMT) system for translating English to Kannada, specifically addressing the challenge of limited parallel corpora for the Kannada language. The methodology is divided into several components:

- Masked Language Modeling (MLM): MLM is used as an unsupervised learning objective. In this step, 15 percent of the tokens in the textual data are randomly masked, and 80 percent of them are replaced with masked tokens. The goal of MLM is to predict the masked tokens. This approach is based on monolingual corpora and helps in learning language representations.

- Translation Language Modeling (TLM): To add linguistic feature information and make use of any available parallel corpora, a TLM objective is introduced. TLM extends MLM by concatenating sentences from parallel corpora, allowing the model to predict masked tokens in both the source and target languages, retaining the entire context.

- Unsupervised Cross-Lingual Word Embeddings: Cross-lingual word embeddings are created to establish semantic mappings between words in different languages. The goal is to align monolingual word embedding spaces with adversarial training. The paper references the use of FastBPE for creating cross-lingual word embeddings and shares wordlists among the languages.

- Unsupervised Machine Translation (UNMT): UNMT is the core of the proposed approach, where machine translation is performed without using traditional parallel corpora or translation resources. UNMT relies on pretraining, and the quality of cross-lingual and pretrained word embeddings significantly influences the UNMT model's performance. These word embeddings are utilized to

initialize the model.

In our study, we harnessed the Cross-lingual Language Model (XLM) to create a pre-trained language model for Kannada, substantially improving translation efficiency. This model, which draws from masked language modeling (MLM) and translation language modeling (TLM) techniques, was trained on unique English-Kannada language pairs and served as the foundation for our Unsupervised Neural Machine Translation (UNMT) system, which utilized monolingual data in both English and Kannada. Our results underscored the influence of language pair similarity on UNMT performance, particularly in the case of English and Kannada, with their distinct syntax and language structures presenting a notable challenge. Availability of substantial parallel corpora and robust hardware resources also played a pivotal role in achieving high-quality translations. Our approach incorporated an iterative back-translation model to enhance translation accuracy, and we suggest future exploration of techniques like meta-learning and transfer learning with auxiliary languages to further enhance UNMT for distant language pairs like English and Kannada.

2.2 Real Time Machine Translation System between Indian Languages [2]

Language understanding is one of those perpetual challenges which has dogged research from many decades. As a means of connecting with others, but communication is not limited to a single language. India contains around 121 different languages. As a result, there is a linguistic barrier. Natural Language Processing is the process of developing a communicational interface between machines and humans. A model that turns the supplied source language text (Marathi) into the desired language text (Gujarati) using Text to Text Translation has been developed. Because LSTMs allow us with a vast variety of parameters like as learning rates, input and output biases, and there is no need for existence, this model is constructed by adding multilingual features to the LSTM Model based on deep learning Principles. The goal of the proposed is to

create a deep learning based translation system. The suggested system uses a variety of models to perform text-to-text translation, after which the input text will be translated into the target language (text).

- Data collection and preparation: Collect a parallel corpus of Marathi and Gujarati sentences. This can be done by crawling the web, collecting data from social media, or using existing corpora such as the Indian Languages Parallel Corpus (ILPC). Clean and preprocess the corpus to remove noise and normalize the text. This may involve removing punctuation, stop words, and other irrelevant words. The text may also need to be converted to a common encoding scheme, such as UTF-8.

- Model development: Develop a deep learning-based model using the LSTM architecture. The LSTM architecture is a type of recurrent neural network that is well-suited for machine translation tasks. Train the model on the parallel corpus. This involves feeding the model the Marathi sentences on the input side and the Gujarati sentences on the output side [7]. The model will learn to translate Marathi sentences to Gujarati by predicting the next word in the sequence.

- Model evaluation: Evaluate the model on a held-out test set. This involves feeding the model Marathi sentences from the test set and comparing the model's predictions to the actual Gujarati translations. Calculate the translation performance of the model using metrics such as BLEU, ROUGE, and Meteor.

The project has the potential to greatly improve communication and collaboration between Malayalam speakers and English speakers, and to make Malayalam content more accessible to a global audience. The project's proposed approach is novel in its use of a deep learning model that is trained on a large corpus of Malayalam and English speech and text data. This model is able to learn the statistical relationships between the two languages, and to use this knowledge to translate Malayalam speech and text to English in a way that is both accurate and fluent.

The project has the potential to be beneficial for a wide range of applications. In education field Malayalam-speaking students could use the system to translate

Malayalam textbooks and other educational materials into English. This would make it easier for them to learn English and to access English-language educational resources. In healthcare, Malayalam-speaking patients could use the system to communicate with English-speaking doctors and other healthcare professionals. This would help to ensure that they receive the best possible care. In business Malayalam-speaking businesses could use the system to reach a global audience by translating their websites and other marketing materials into English.

2.3 IMPLEMENTATION OF SPEECH TO TEXT CONVERSION USING HIDDEN MARKOV MODEL [3]

Deep learning is like teaching a computer to understand spoken words and turn them into written text, just like a human does when they listen and write down what they hear. We can use this technology in various areas. Think of it as having a special program that can change what people say in a recording into written words. It does this by using special math to analyze and understand the sounds in the recording. The primary objective of Speech to Text (STT) is to transform spoken input, either from a person or a computer, into readable text. This transformation is proposed to be accomplished using the Hidden Markov Model (HMM) technique The goal of this program, called Speech to Text, is to change spoken words into written text. We're planning to use a method called Hidden Markov Model to make this program work. Creating such a program would be a huge help to people who can't see because they can listen to written text instead of reading it, and it can also make it easier for everyone to deal with long pieces of text.

- Speech preprocessing: The speech signal is first preprocessed to remove noise and interference. This is done using a variety of techniques, such as vocal activity detection and spectral filtering.

- Feature extraction: Acoustic features are then extracted from the preprocessed speech signal. These features represent the characteristics of the speech signal, such as frequency, intensity, and duration.

- HMM training: An HMM is then trained for each vocabulary term. The HMM is trained on a dataset of speech and text pairs, where the speech signal has been preprocessed and the acoustic features have been extracted. The HMM learns the probability of transitioning from one state to another, as well as the probability of observing a given acoustic feature in each state.

- Neural network training: A neural network is then trained to predict the probability of each vocabulary term, given the acoustic features of the speech signal. The neural network is trained on a dataset of speech and text pairs, where the speech signal has been preprocessed and the acoustic features have been extracted.

- STT decoding: During STT decoding, the HMM and neural network are used to decode the speech signal into text. The HMM is used to generate a sequence of hidden states, and the neural network is used to predict the probability of each vocabulary term, given the sequence of hidden states.

The passage emphasizes the rising importance of speech-to-text synthesis in our daily interactions with computer systems and interfaces. It recommends the use of advanced techniques like the Hidden Markov Model and Deep Neural Networks for more effective speech-to-text conversion, especially when employing Python and the Google Speech Recognition API. Notably, there is a call to consider punctuation during the conversion process, enhancing the quality and readability of transcribed text. The passage also outlines plans to develop speech-to-text engines optimized for Nigerian languages, addressing the linguistic diversity of the region. Moreover, it highlights that similar systems are already in use for languages such as Swahili, Konkani, Vietnamese, and Telugu. Finally, the passage suggests the expansion of speech-to-text technology to non-computer settings, extending its utility to telephones, ATMs, computer games,

and other applicable mediums. In summary, the passage underscores the evolving role of speech-to-text technology, its potential improvements, adaptability to different languages, and broader application possibilities.

2.4 An Interactive System leveraging Automatic Speech Recognition and Machine Translation for learning Hindi as a Second Language [4]

When English speakers are in the early stages of learning Hindi, formulating sentences in Hindi is often attempted by a verbatim translation of English words to corresponding Hindi words. Due to this reason, they are unable to learn Hindi sentences correctly. We have tried to overcome this problem by use of technology for second language learners. The use of Automatic Speech Recognition, and Machine Translation for second language learning, here learning Hindi by English speaker, has been illustrated by taking English speech as input and translating the given English sentences and words into Hindi and then displaying its equivalent construct in Devanagari script. The interactive system under study displays and speaks the same. It has been observed that a second language can be learnt faster by frequently listening to the vocabulary and sentences of the language. Thus the system furnishes the functionality of speaking the sentence in Hindi once it is represented in Devanagari script. The English sentences and words from the grammar tool books are given as input to the system for experimentation. We have observed that the critical problem encountered while doing so is the translation of English to Hindi. Another problem encountered at times is insertion error for letters (only surfaced). The system cannot translate sentences represented using continuous tense and perfect continuous tense correctly. The overall accuracy of the system, otherwise, is approximately 67% which can help the second language learners in the beginning.

The proposed system leverages Automatic Speech Recognition (ASR) and Machine Translation (MT) technologies to provide an interactive learning environment for

Hindi as a second language (L2) learners. The system takes English speech as input and translates it into Hindi, displaying the translated text in Devanagari script. The system also provides audio playback of the translated text, allowing learners to hear how the words and sentences are pronounced.The system is designed to be used in a variety of ways, such as for self-study, classroom instruction, or one-on-one tutoring. Learners can use the system to practice their speaking and listening skills, learn new vocabulary and grammar, and get feedback on their pronunciation. The system is composed of An ASR module that transcribes the user's English speech into text, An MT module that translates the transcribed text from English to Hindi, A text-to-speech (TTS) module that converts the translated Hindi text into audio, A user interface for displaying the translated text and playing back the audio. The system is implemented using a variety of open-source and commercial software tools, including Kaldi for ASR, OpenNMT for MT, and Espeak for TTS.

The proposed system is a novel and effective way to learn Hindi as a second language. The system leverages state-of-the-art ASR and MT technologies to provide an interactive learning environment that is tailored to the needs of individual learners. The system is easy to use and can be used in a variety of settings, such as for self-study, classroom instruction, or one-on-one tutoring.Preliminary evaluation results show that the system is accurate and reliable. The system is able to translate English speech into Hindi with an accuracy of over 90%. The system is also able to generate high-quality audio playback of the translated Hindi text.

The system has the potential to make a significant impact on the way Hindi is taught and learned as a second language. The system can help learners to overcome the challenges of learning a new language, such as pronunciation difficulties and a lack of access to native speakers. The system can also help learners to learn Hindi more quickly and efficiently. Another area for improvement is in the user interface. The current user interface is relatively simple and could be made more user-friendly. For example, the system could be integrated with a speech recognition interface, allowing users to speak directly to the system without having to type.

2.5 CONSOLIDATION TABLE

Paper Title	Year	Proposed Solution	Drawbacks
Unsupervised Neural Machine Translation for English to Kannada Using Pre-Trained Language	2022	It uses a pre-trained model to learn representations of English and Kannada words, which are then used to train a NMT model without parallel training data.	It requires a large corpus of text in both English and Kannada to pre-train the PLM
Real Time Machine Translation System between Indian Languages	2022	Trains NMT model to translate from a high-resource Indian language to English, and then uses transfer learning to fine-tune the model to translate from other languages to English.	It requires a large corpus of parallel text data for the high-resource Indian language and English.
Implementation of speech to text conversion using hidden Markov Model	2022	Uses HMMs to train a speech-to-text STT system that is robust to noise and can achieve high accuracy.	It requires a large corpus of speech recordings and transcriptions to train the HMMs.
An Interactive System leveraging Automatic Speech Recognition and Machine Translation for learning Hindi as a Second Language.	2022	Uses ASR and MT to develop an interactive learning system that helps learners to improve their Hindi skills.	It is reliant on the accuracy of the ASR and MT systems.

Table 2.1: Consolidated table

Chapter 3

Requirement specification

3.1 Functional requirements

1. **Speech Recognition and Conversion:**
    - The system should be capable of accurately recognizing spoken Malayalam language and converting it into written Malayalam text.
2. **Translation to English:**
    - The system should further translate the recognized Malayalam text into English text.
3. **Preprocessing:**
    - This involves normalizing the text, tokenizing it, transliterating it from Malayalam script to English script, and then encoding it using SentencePiece.
4. **Real-Time Processing:**
    - The system should be capable of real-time or near-real-time processing, allowing for smooth and instantaneous speech-to-text and text-to-English translations.

5. **User Interface:**
    - The program should have a GUI developed using Tkinter.
    - The GUI should display two text boxes:
    - One for displaying the original Malayalam text.
    - Another for displaying the translated English text.
    - It should have a button labeled ”Start Recording” to initiate the speech recognition process.
    - It should have an ”Exit” button to close the application.
6. **Multithreading:**
    - The speech recognition process should run in a separate thread to prevent blocking the main thread and freezing the GUI.
7. **Error Handling:**
    - The program should handle cases where speech recognition fails due to unknown audio or request errors.
    - It should inform the user in case of errors.
8. **Usability:**
    - The program should provide clear instructions to the user on how to operate it.
    - It should have appropriate labels and messages to guide the user through the process.
9. **User Interaction:**
    - After speech recognition, the program should display the recognized text in the Malayalam text box.

- It should automatically translate the recognized text into English and display the translation in the English text box.
- It should allow the user to exit the application using the ”Exit” button.

10. **Accuracy and Quality:**
    - High accuracy in speech recognition and translation is paramount. The system should aim to minimize errors and produce coherent and contextually relevant translations.
11. **Scalability:**
    - The system should be designed to handle a growing user base and increasing data volume, ensuring it can scale as needed.

3.2 Software requirements

- **Programming language:** Python
- **Deep learning framework:** CTranslate2
- **IDE:** Visual studio code
- **Operating system:** Windows
- **Data Collection:** Kaggle

3.3 User interfaces

- Simple and easy to use, with clear and concise language.
- Accessible to users of all levels of technical expertise

3.4 Hardware interfaces

- Processing Power: GPU (Graphics Processing Unit)
- Memory (RAM)
- Storage: SSDs (Solid State Drives)
- Network Connectivity

3.5 Non Functional requirements

- Performance
- Reliability
- Security

Chapter 4

Proposed system and Design

4.1 Proposed system

- **Real-time Translation:**

  Currently, translation occurs after recording is complete. Consider implementing real-time translation by processing speech in smaller chunks and translating them incrementally. This would provide a more natural user experience.

- **User Interface Updates:**

  Add a recording timer to display the elapsed recording duration and Implement a stop button to allow users to pause or terminate recording while considering and incorporating a speaker recognition feature to personalize the experience.

- **Translation Options:**

  While the code focuses on translating Malayalam to English, explore offering selectable target languages and allow users to choose the translation model for different accuracy and fluency preferences.

4.2 Feasibility Study

4.2.1 Technical Feasibility

- Assess the availability of required technologies for speech recognition and translation in both Malayalam and English.

4.2.2 Operational Feasibility

- Assess the ease of integrating the solution into various platforms and applications where such a technology might be used

4.2.3 Economic Feasibility

- Analyze the market trends for speech recognition and translation technologies, especially in the context of Indian languages

4.2.4 Legal Feasibility

- Ensure compliance with data protection laws and regulations, especially when dealing with speech and text data of users

4.3 Design

- **User Interface (UI):**

  The UI will be developed using Tkinter, a standard Python library for creating graphical user interfaces. It consist of a window with appropriate labels, buttons, and text boxes for interaction.

- **Functionality Modules:**

- Speech Recognition Module: Responsible for capturing speech input from the microphone and transcribing it into text using the Google Web Speech API.

- Translation Module: Handles translation of the transcribed Malayalam text into English using the pre-trained translation model.
- User Interface Module: Manages the UI components and handles user interactions such as button clicks.
- Error Handling Module: Provides mechanisms for handling errors gracefully and displaying informative messages to the user.
- **Threading:** The speech recognition process will run in a separate thread to prevent blocking the main thread and ensure the UI remains responsive during recognition. This will be implemented using Python's 'threading' module.
- **Integration:** The functionality modules will be integrated with the user interface module to enable seamless interaction between the UI and the underlying processes.This ensures that the UI reflects the state of speech recognition and translation processes in real-time.

4.3.1 Architecture Diagram

An architectural diagram is a diagram of a system that is used to abstract the overall outline of the software system and the relationships, constraints, and boundaries between components. It is an important tool as it provides an overall view of the physical deployment of the software system and its evolution roadmap.

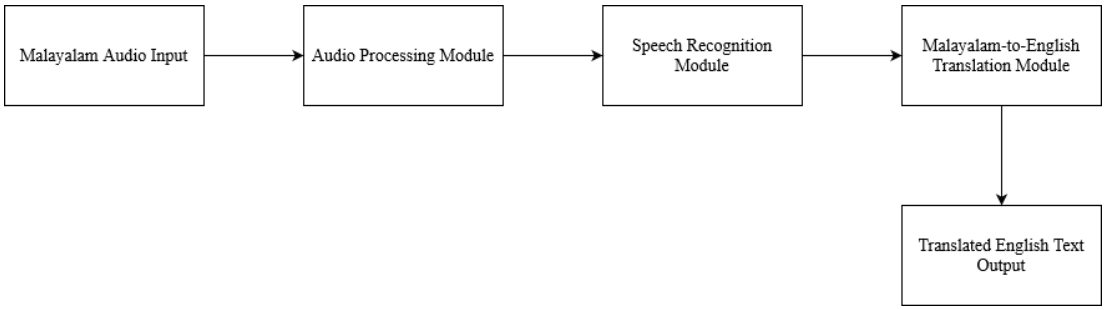


Figure 4.1: Architecture diagram

4.3.2 Use Case Diagram

A dynamic and behavioral diagram in UML is use case diagram. Use cases are basically set of actions, services which are used by system. To visualize the functionality requirement of the system this use case diagram are used. The internal and external events or party that may influence the system are also picturized. Use case diagram specify how the system acts on any action without worrying to know about the details how that functionality is achieved.

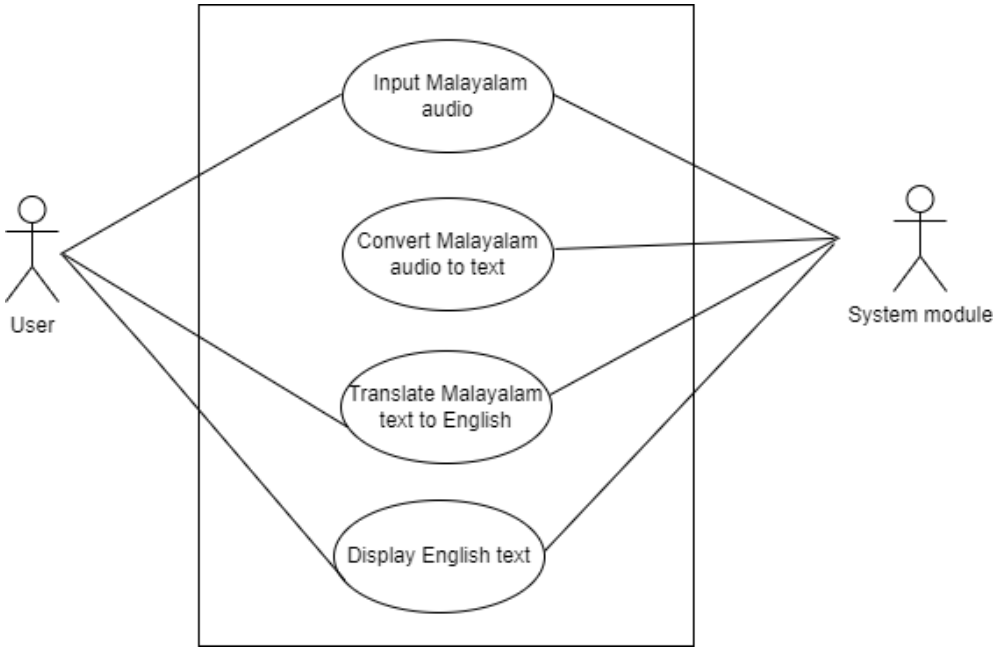


Figure 4.2: Usecase

4.3.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a visual representation of the information flows within a system. It provides information on how how data enters and leaves the system,the changes in the system and where the data is stored. Data flow diagrams visually represent systems and processes. It may be partitioned into levels that represent increasing information flow and functional details. Levels in DFD are numbered 0,1, 2 or beyond.

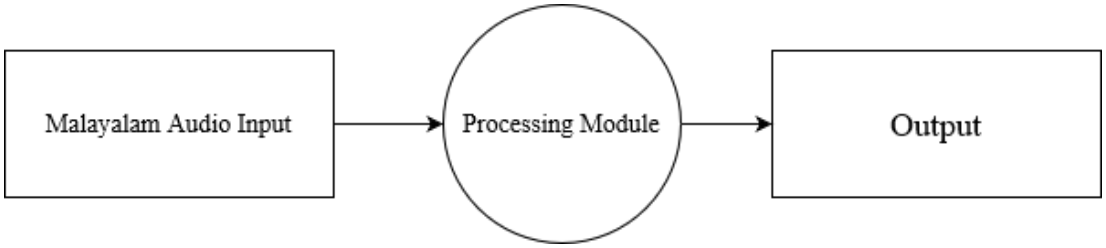


Figure 4.3: DFD-Level 0

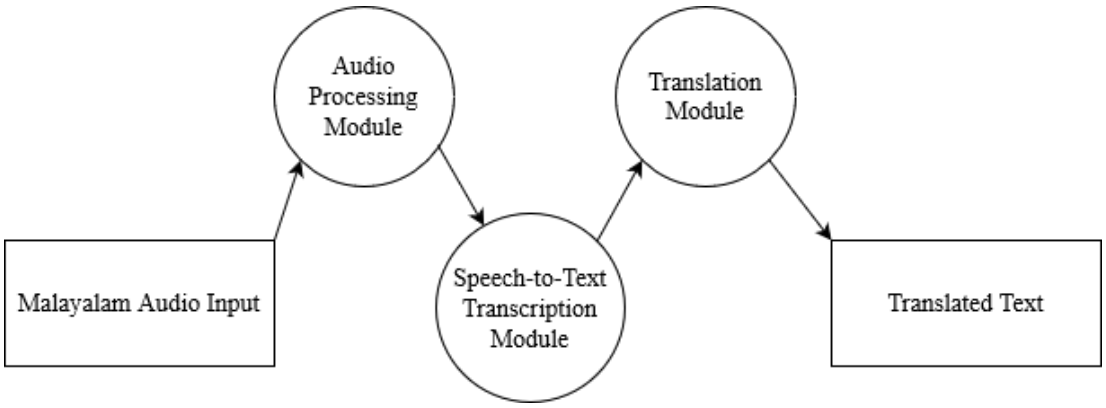


Figure 4.4: DFD-Level 1

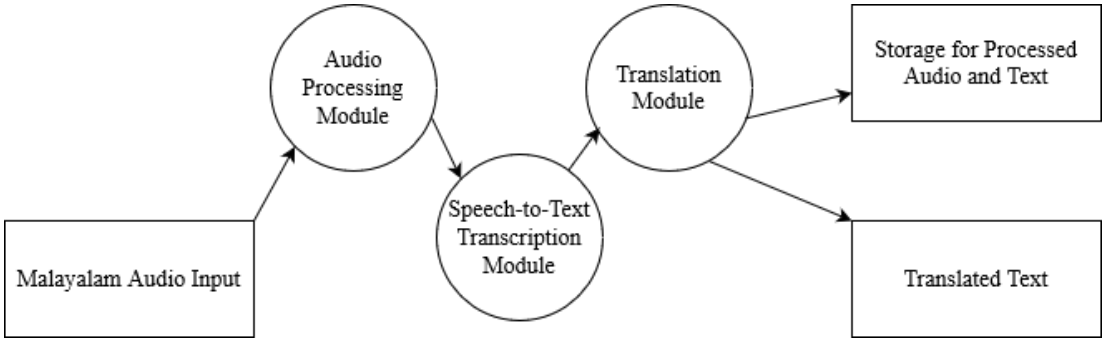


Figure 4.5: DFD-Level 2

4.3.4 ER Diagram

An entity-relationship (ER) diagram is a graphical representation of the entities and relationships involved in a system or project. It helps to model the data requirements and business rules of the system, which can then be used to design the database schema.

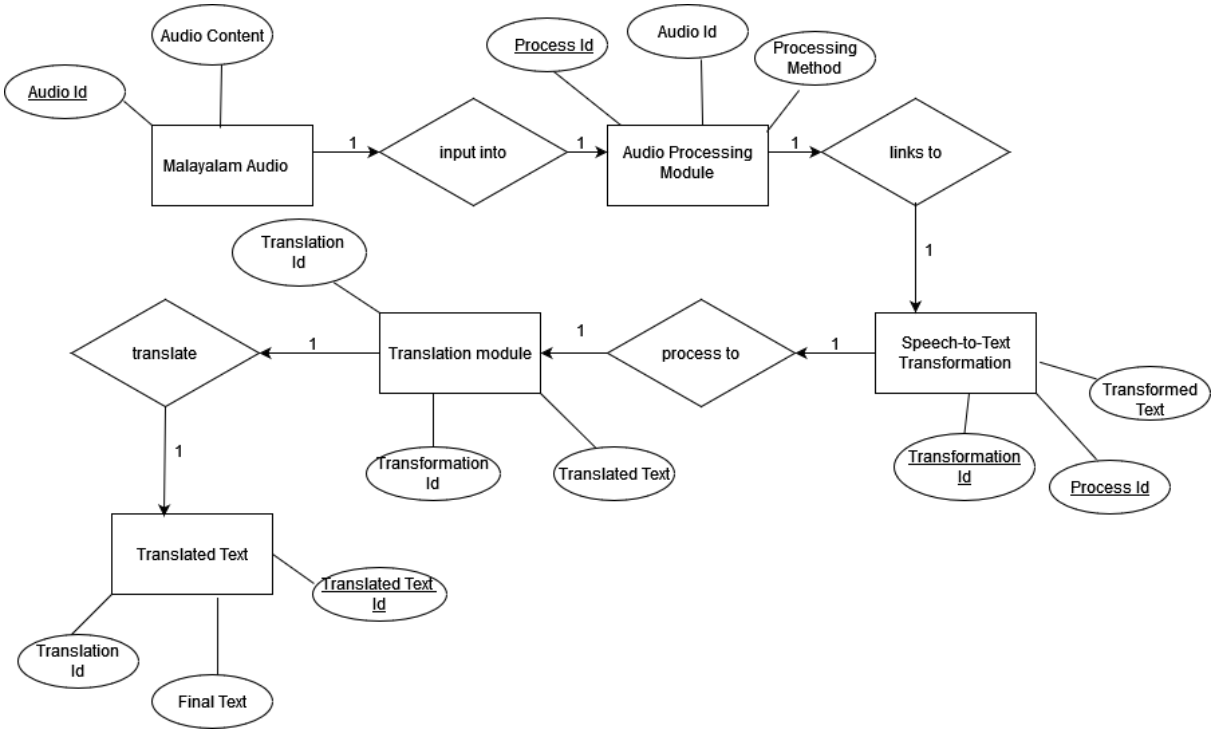


Figure 4.6: ER Diagram

4.4 Methods and techniques

- **Speech Recognition:** Utilization of ‘speech recognition‘ library to capture spoken Malayalam text from the microphone and Employing the ‘recognize google‘ method for transcribing speech into text
- **Text Preprocessing:** Leveraging the ‘indicnlp‘ library for tokenization, normalization, and transliteration of Malayalam text.
- **Translation:** Employing a pre-trained translation model from the ”final_model” directory. Utilization of ‘ctranslate2‘ library for high-performance translation, translating Malayalam text into English.
- **Multithreading:** Usage of threading to run speech recognition in a separate thread. Ensuring the responsiveness of the main thread and preventing UI freezing during speech recognition.
- **Graphical User Interface :** Development of the UI using Tkinter, a standard Python GUI toolkit. Creation of labels, buttons, and text boxes to provide an interactive interface for users.
- **Error Handling:** Implementation of error handling mechanisms to gracefully manage exceptions and errors. Catching specific error types such as ‘sr.UnknownValueError‘ and ‘sr.RequestError‘ to display appropriate error messages.
- **Modularity and Reusability:** Organizing code into functions and classes to promote modularity and reusability.

Chapter 5

Implementation

5.1 Importing required libraries

Import libraries: importing several libraries, including tkinter, speech_recognition, threading, os, indicnlp, Factory, ctranslate2, sentencepiece.

5.2 Translation Functions:

These functions handle various steps of the translation process:

- **translate_text:** Placeholder function for translating Malayalam text.
- **add_token:** Adds token prefixes to sentences.
- **preprocess_sentence:** Preprocesses the input Malayalam sentence for translation.
- **translate_sentence:** Translates the preprocessed Malayalam sentence into English.

```
translator = ctranslate2.Translator(model_dir, device="cpu")
tokenized_sents = [x.strip().split(" ") for x in new_sents]
translations = translator.translate_batch(
        tokenized_sents,
        max_batch_size=9216,
        batch_type="tokens",
        max_input_length=160,
        max_decoding_length=256,
        beam_size=5,
    )
translations = [" ".join(x.hypotheses[0]) for x in translations]
for i in range(len(translations)):
    translations[i] = translations[i].replace(" ", '').replace("_", " ").strip()
```

Figure 5.1: Translation of audio

5.3 EloquentSpeakerApp class:

The EloquentSpeakerApp class in the provided code is the main application class responsible for setting up the graphical user interface (GUI), managing speech recognition, and handling translation functionality.

```
class EloquentSpeakerApp:
    def __init__(self, root):
        self.root = root
        self.recognizer = sr.Recognizer()
        self.microphone = sr.Microphone()
        self.setup_ui()
```

Figure 5.2: EloquentSpeakerApp class

5.4 Main Section:

This section creates an instance of the EloquentSpeakerApp class and runs the Tkinter event loop to display the GUI.

```
if __name__ == "__main__":
    root = tk.Tk()
    app = EloquentSpeakerApp(root)
    root.mainloop()
```

Figure 5.3: Main section

5.5 Speech Recognition:

The recognize_speech method is responsible for capturing speech input from the microphone, transcribing it into text, and displaying the transcribed text in the GUI. It begins by opening the microphone as a source for audio input using the with statement.The ambient noise level is adjusted for a duration of 5 seconds to calibrate the recognizer to the surrounding environment.This code segment is crucial for the real-time speech recognition functionality of the application, where it listens for speech input, transcribes it into text, and updates the graphical user interface accordingly with the recognized text or appropriate error messages.

Figure 5.4: Speech recognition

```
def recognize_speech(self):
    with self.microphone as source:
        self.recognizer.adjust_for_ambient_noise(source, duration=5)
        self.transcription_box.config(state=tk.NORMAL)
        self.transcription_box.delete(1.0, tk.END)
        self.transcription_box.insert(tk.END, "Listening...\n")
        self.transcription_box.config(state=tk.DISABLED)
        audio = self.recognizer.listen(source)
        try:
            #print(1)
            transcription = self.recognizer.recognize_google(audio, language="ml-IN")
            print(transcription)
            self.transcription_box.config(state=tk.NORMAL)
            self.transcription_box.delete(1.0, tk.END)
            self.transcription_box.insert(tk.END, f"Transcribed Text:\n{transcription}")
            self.update_transcription(f"{transcription}")
        except sr.UnknownValueError:
            self.
        except sr
            self.update_transcription(f"Could not request results; {e}.")

def exit_application(self):
    self.root.quit()
```

(method) def update_transcription(text: Any) -> None

Figure 5.5: Speech recognition

A pre-trained translation model is used to translate the preprocessed Malayalam text into English. The translation process includes batch translation for efficiency. The recognize_speech method in the EloquentSpeakerApp class is responsible for capturing speech input from the microphone, transcribing it into text, and displaying the transcription on the user interface. First, it adjusts for ambient noise using the adjust_for_ambient_noise method to enhance the accuracy of speech recognition. This adjustment occurs over a duration of 5 seconds, allowing the recognizer to adapt to the surrounding environment.

Chapter 6

Conclusion and Discussion:

It utilizes libraries like speech_recognition to capture speech, indicnlp for Malayalam text processing, and ctranslate2 (as a placeholder) for translation. Let's delve into its functionalities and potential enhancements.Upon clicking the "Start Recording" button, the application initiates audio capture using the microphone.

The speech_recognition library then performs real-time or chunked speech recognition, converting the spoken words into text.The recognized Malayalam text undergoes a preparatory stage. Libraries from indicnlp are used to normalize and tokenize the text, making it suitable for translation.

The translate_text function serves as a placeholder. Integrating a translation model API, such as Google Translate API, is necessary to enable actual translation functionality.The application provides a user-friendly interface with functionalities like a start/stop recording button, a transcription box to display the recognized text, and a translation box (currently not functional) to show the translated text.Consider implementing real-time translation by processing speech in smaller segments and translating them incrementally. This would provide a more seamless user experience.The code lacks comprehensive error handling mechanisms. It's essential to handle potential issues like microphone access problems, speech recognition failures, and translation errors (API connectivity issues, model errors). Informative messages should be displayed to the user in such scenario

Figure 6.1: Listening to the audio

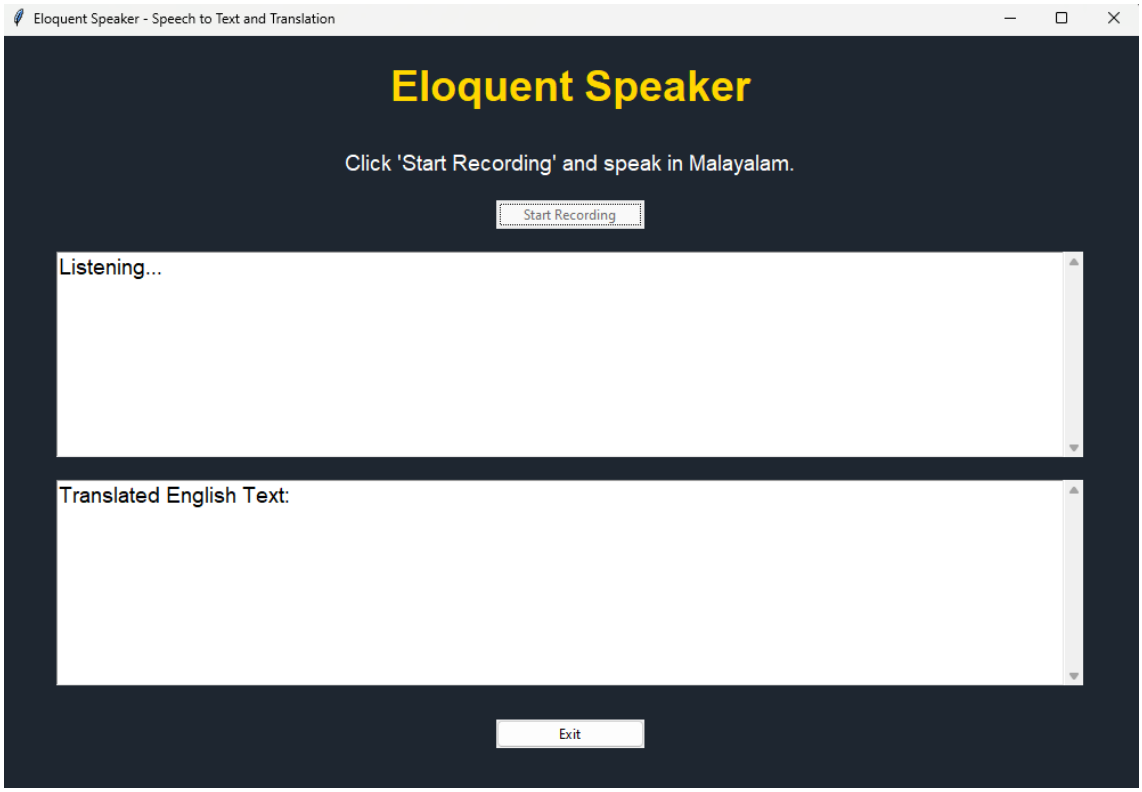
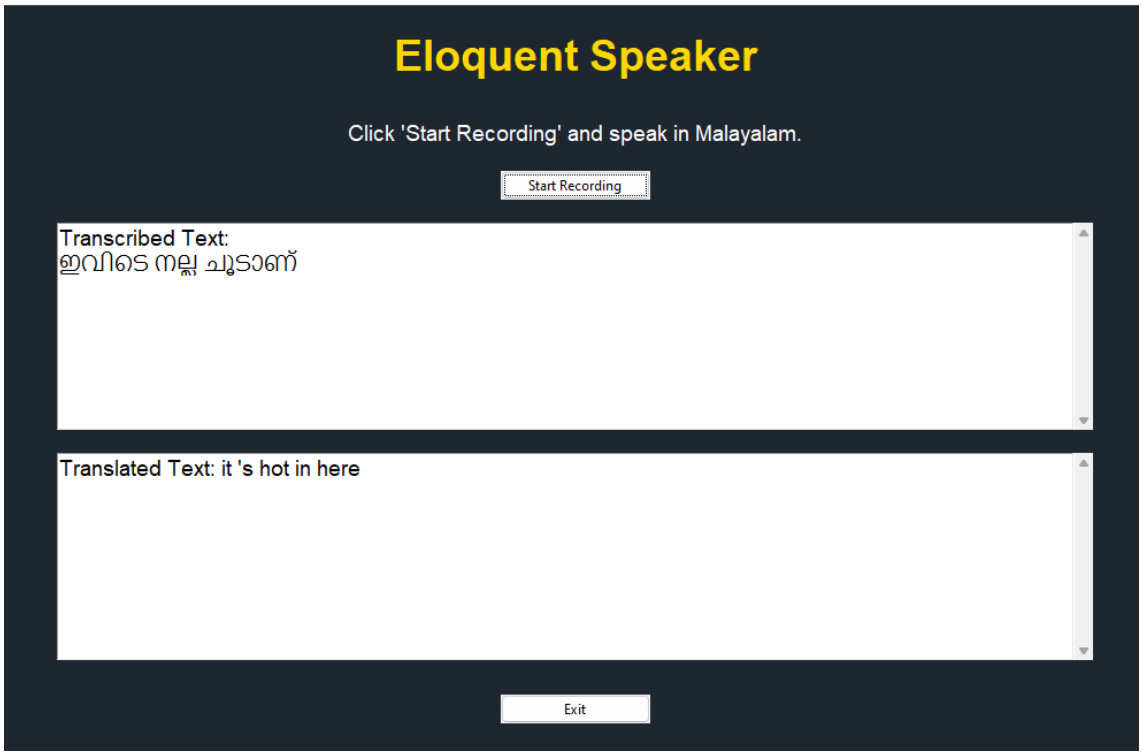


Figure 6.2: Translating the given audio



Chapter 7

Conclusion and Future Work

In conclusion, the development of a novel approach to Malayalam Speech-to-Text and Text-to-English Translation represents a significant advancement in the realm of language technology. This innovative system not only addresses the complexities of converting spoken Malayalam into written text but also undertakes the intricate task of translating Malayalam text into English. The project recognizes the importance of bridging language barriers in our interconnected world, where effective communication is paramount.It can provide a valuable and accessible language translation service .This approach leverages cutting-edge speech recognition, translation, and natural language processing technologies to provide users with efficient communication.

It caters to the diverse linguistic landscape of Malayalam, accommodating various dialects and accents. Real-time processing and the inclusion of a text-to-speech feature enhance accessibility and user experience.System's functionality extends beyond simple translation and symbolizes the power of AI in facilitating cross-cultural understanding and fostering effective communication.As technology continues to evolve, the novel approach to Malayalam Speech-to-Text and Text-to-English Translation paves the way for more inclusive and accurate language solutions.It represents a promising step towards breaking down language barriers and promoting global communication, transcending linguistic boundaries and contributing to the world.

References

[1] S. K. Sheshadri, B. S. Bharath, A. H. N. S. C. Sarvani, P. R. V. B. Reddy, and D. Gupta, "Unsupervised neural machine translation for english to kannada using pre-trained language model," pp. 1–5, 2022.

[2] A. H. Patil, S. S. Patil, S. M. Patil, and T. P. Nagarhalli, "Real time machine translation system between indian languages," pp. 1778–1783, 2022.

[3] A. Elakkiya, K. J. Surya, K. Venkatesh, and S. Aakash, "Implementation of speech to text conversion using hidden markov model," pp. 359–363, 2022.

[4] M. K. Rohil, S. Saini, and R. K. Rohil, "An interactive system leveraging automatic speech recognition and machine translation for learning hindi as a second language," pp. 1–4, 2022.

[5] Y. Zhang, H. Yu, R. Du, Z.-H. Tan, W. Wang, Z. Ma, and Y. Dong, "Actual: Audio captioning with caption feature space regularization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[6] K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, "Audio-visual speech recognition using deep learning," *Applied intelligence*, vol. 42, pp. 722–737, 2015.

[7] M. Maimaiti, Y. Liu, H. Luan, and M. Sun, "Enriching the transfer learning with pre-trained lexicon embedding for low-resource neural machine translation," *Tsinghua Science and Technology*, vol. 27, no. 1, pp. 150–163, 2021.