Development of a Language Translation & Language Understanding Model UsingMachine Learning

Mohammed. Juneduddin[1,*], Ansari Awais Sarosh [1], Shaikh Sohail Kaleem [1],
Wagh Gaurav Bharat [1], Lohar Kaushal Jayant [1]
[1]Department of Mechanical Engineering, SVKM's Institute of Technology, Dhule, Maharashtra, India

Corresponding Author Email : nooralam2011@gmail.com

Abstract

In our increasingly globalized world, language serves as a bridge for human interaction but also poses significant obstacles. The diversity of global languages, particularly in regions like India, challenges digital literacy due to uneven access. This research introduces an innovative approach to foster digital inclusion, aligning with the "Digital India" initiative. Our primary goal is to leverage Machine Learning for advanced Language Translation and Understanding Models. Translation, as the process of encoding and decoding information, becomes particularly challenging for less explored language pairs such as Marathi and English or Hindi and English. While achieving perfect translation remains elusive, Machine Translation aims to provide comprehensible interpretations. Our focus lies in developing Machine Translation systems tailored for low-resource language pairs, specifically the English-Marathi language pair, in both translation directions. In the initial phase, we utilize provided parallel training data within specific constraints. In the subsequent phase, we expand our horizons by incorporating parallel corpora from various sources to address English-Marathi and English-Hindi translations, maximizing our Machine Translation capabilities. Sitting at the intersection of Machine Translation and Natural Language Processing, this research offers a transformative path to bridge linguistic divides, foster digital inclusion, and create a brighter future for India's diverse linguistic landscape.

Keywords: Machine Learning, Machine Translation, Natural Language Processing (NLP), HTML, Tailwind CSS, React.Js, Python, Flask, Jinja2, Azure Translator & Speech API, Azure CognitiveServices, NLTK (Natural Language Toolkit) , Microsoft Azure, Prisma, TensorFlow.

1. INTRODUCTION

2. In our increasingly interconnected world, language stands as the foremost conduit for human interaction, reflecting the sheer diversity of human expression and posing complex challenges [3]. With an intricate tapestry of languages spoken and written across the globe, the barrier to digital literacy becomes pronounced, especially in regions like India, marked by linguistic diversity and uneven access to digital resources [1]. This research endeavors to introduce an innovative solution that transcends language barriers and aligns with the "Digital India" initiative. Translation serves as the essential bridge between people speaking different languages, a process that encodes information from one language and decodes it into another, adhering to the target language's rules. While attempts have been made to automate this process for various language pairs, challenges persist, and the accuracy, particularly for regional languages like Marathi and Hindi, remains a subject of exploration [2]. Machine Translation strives to provide comprehensible interpretations, even if they may occasionally exhibit slight imperfections. This research project focuses on the development of Machine Translation systems, specifically tailored for low-resource language pairs, with a primary focus on the English-Marathi language pair in both translation directions. The study unfolds in two phases: one constrained, leveraging provided parallel training data, and another unconstrained, utilizing parallel corpora from diverse sources for English-Marathi and English-Hindi translations.

Positioned at the intersection of Machine Translation and Natural Language Processing (NLP), this research contributes to bridging linguistic divides, fostering digital inclusion, and supporting the vibrant linguistic diversity of India [3]. The challenges in translating Marathi to English are distinctive, encompassing both syntactic and morphological differences, underlining the necessity for a rule-based machine translation system [4]. This work seeks to address these challenges while promoting broader accessibility to the Marathi language, thereby enhancing global communication and knowledge sharing. In a world where English

is spoken as a first language by approximately 375 million people and Marathi represents the mother tongue of more than 80% of Maharashtra's population, automated translation from Marathi to English is crucial for effective communication. Existing tools like Google Translator rely on Statistical Machine Translation, and this research explores advanced techniques like Transformer RNN based Neural Machine Translation and sub-word segmentation with byte pair encoding (BPE) to enhance translation quality [4].

The aims and objectives of this research encompass creating a unique combination of services: first, facilitating language translation to enable inter-translatability of news summaries into multiple Indian languages, and second, enhancing language understanding to provide relevant news summaries and video suggestions based on keyword results [5, 6]. This multifaceted research venture seeks to revolutionize language translation and understanding, breaking down linguistic barriers for a brighter and more connected future [7].

2.METHODOLOGY

1.1 System Architecture

1.1.1 Categorization of needs in 3 components.

1.1.1.1 The Frontend

1.1.1.2 The Backend.

1.1.1.3 And the Logic

1.1.2 The Logic is component which works as connection between the Frontend and the Backend and perform operations on it.

1.1.3 It is a Machine Learning based app so it will also have the ML and DL models and Algorithms in itto perform operations on the data.

1.1.4 Here, the frontend is handled by the popular JS framework React.js and for Tailwind CSS willbe usedfor custom styles due to its modularity.

1.1.5 The backend will be mostly handled by Flask, a very popular python web framework library.With help from some tools from Microsoft Azure Cloud.

1.1.6 The logic operations primarily the DL operations will be performed by using Tensor Flow python library and GPT3.

1.1.7 GPT-3 stands for Generative Pre-trained Transformer 3 it is an autoregressive language modelthat uses deep learning to produce human-like text.

1.1.8 It is the third-generation language prediction model in the GPT-n series created by OpenAI, aSan Francisco-based artificial intelligence research laboratory in contribution with Tom Brown &Benjamin Mann of John Hopkins University.

1.1.9 We have used Azure Translator Service, therefore our app will have and use same underlying concept for Language Translation and Understanding as Azure Services.

1.1.10 And with this all the needs of our proposed projects are fulfilled. This concludes that our proposed project is technically and operationally feasible.

1.1.11 Next is system architecture. Here is the simplified version of the backend architecture of our proposed system. It is divided in 3 sections. The Algorithm Section, The Programming Section, TheAPI Section.

1.1.12 In this architecture the query requests from frontend are sent to the API Section. It handles thescraping of language input from user.

1.1.13 This Input is then processed through our NLP algorithm and APIs from Azure Cognitive Servicesfor getting the Text/Audio type to machine readout format.

1.1.14 Input is now process through the Azure Translator Algorithm.

1.1.15 The algorithm will be developed by Microsoft containing the Technologies mentioned in referencesfrom Above mentioned

research papers with the help of GPT-3 from OpenAI and the ML models will be trained using RNN, Text2Vector Model, Seq2Seq Machine Translation algorithms etc.

1.1.16 The DB storage will be handled by itself as we are using Azure Cloud Service.

1.1.17 After processing the input, generated output in target language is sent to backend of system.

1.1.18 The Frontend then displays the query output from the DB.

1.1.19

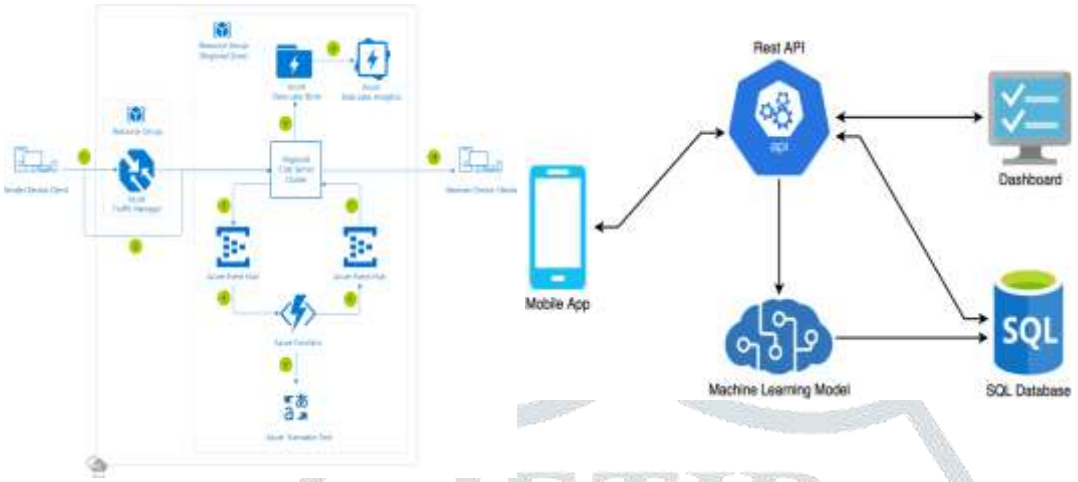

Figure 1.System Architecture & Flow Diagram

1.2. Seq2Seq Modelling

Sequence to sequence modelling is the basis of all summarization abstractive solutions, in which an encoder decoder model is formed using LSTMs. This approach was not very promising, because we knew that starting a model from scratch will require intensive optimization and experimentation but wedecided to go on with it to have hands-on-experience with basic abstractive models.

First step was to apply data processing and change sentences to sequences, and pad them, becausethe model will take vectors of numerical sequences of fixed size.

The process went as follows

1. For the text examples we:
   a. Removed unnecessary spacing.
   b. Changed characters to lowercase
   c. Removed Unwanted punctuation
   d. Removed Stop words
2. For test samples:
   a. Same processing was done without removing stop words
   b. Start and end tokens were added
3. We used Keras Tokenizer to tokenize and change texts to sequences
4. We padded sequences to have a fixed sized input

The second step was to create the encoder-decoder model, and train it on the datasetEncoder Layer:

1. Encoder Embedding Layer
2. 3 Encoder LSTM Layers

Decoder Layer: Embedding Layer, 1 LTSM layer, dense layer

After Training the model we had to divide it into two separate models:

1. **Encoder Model**: containing the encoding layers
2. **Decoder Model**: containing the decoding layers

This separation is because RNNs generate each instance given the last ones. So, the decoder needstoget the sequences decoded before as an input with each prediction. We finally generated the summarization function

2.3. System Architecture design:

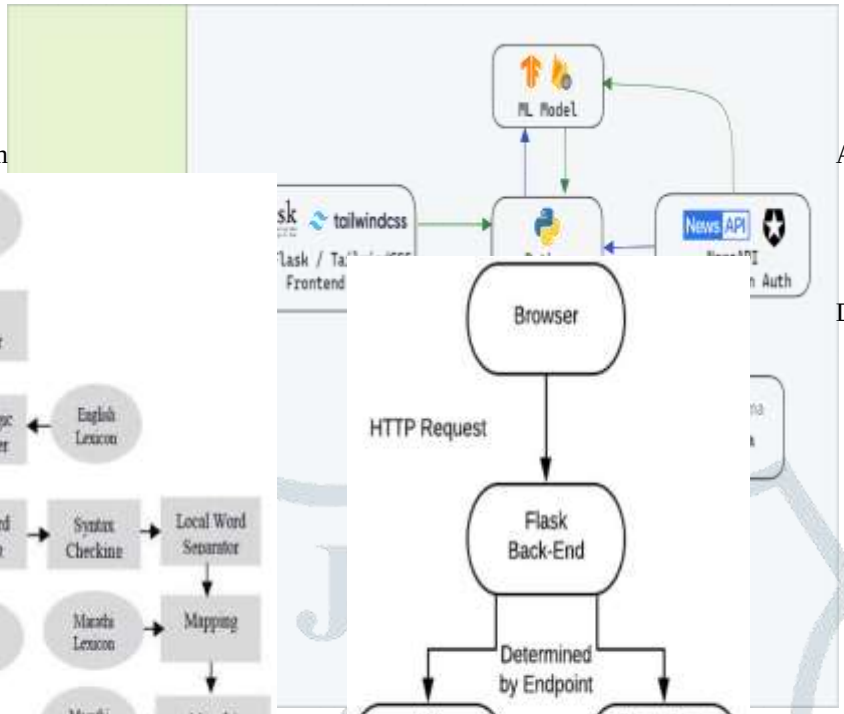

Fig 2: System Architecture

Fig 3: Data Flow Diagram

1.3. Technical Feasibility

1. For our project there are 3 main technical requirements.

a. A Good Frontend UI/UX

b. A Good Backend API

c. A fast and accurate ML algorithm which is feasible by all aspects

2. The use of frameworks like React.js and Tailwind resolves the requirement of good Frontendand provide a good user experience (UX).

3. For news site most important thing is legitimate reputable news sources and that need is resolvedusing Azure API. The database storage need is handled by Azure DB Services hosted on Azure

Cloud for all other storage needs.

4. And for ML operations, the available resources both hardware and software available today are capable and allow as well as encourage the handling, development, and deployment of MachineLearning algorithm for any purposes. The companies like Google and Facebook have released TensorFlow and Keras respectively for development of Machine Learning algorithms and models which can be deployed on the vast cloud Infrastructures like Azure, AWS, GCP for very economical prices. The NLP (Neural Language Processing) libraries like GPT2/GPT3by OpenAIare very helpful in training machine to understand the text and Google Cloud Speech API is verygood for text to speech processing.

1.4. Behavioral Feasibility

1. It is a practical preposition with execution of every mentioned feature possible using AIalgorithms

2. Similar technology exists with just 10% features similar to our plans

3. The provided solution can be applied to solve the current lengthy, scattered and unverified news problems.

4. The NLP (Neural Language Processing) libraries like GPT2/GPT3 by OpenAI have the capacityto technically handle the solution

5. Flexible and easy navigations with simple interfaces.

6. Feasible to develop the ecosystem with current ML algorithms

7. All the mentioned algorithms can be upgraded and modified for each feature of our new developed system

8. Cost effective news and information services to the consumers

9. Accuracy of news display with source verification algorithms for every news.

10. All in one ecosystem eliminating the need for third party platforms to search and share news

11. Simultaneous collection of audio, videos and related images for a specific news.

1.5. Technologies Used

1. HTML: HTML is the standard markup language for creating Web pages. HTML stands for Hypertext Markup Language. HTML describes the structure of Web pages using markup. HTMLelements are the building blocks of HTML pages. HTML elements are represented bytags.

2. Tailwind CSS: Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all thebuilding blocks you need to build bespoke designs without any annoying opinionated styles you must fight to override. The beauty of this thing called tailwind is it doesn't impose design

specification or how your site should look like, you simply bring tiny components together to construct a user interface that is unique. What Tailwind simply does is take a 'raw'CSS file, processes this CSS file over a configuration file, and produces an output.

3. **React.Js:** React is a library for building compostable user interfaces. It encourages the creation ofreusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance.

4. **Python:** Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, includingstructured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle-detecting garbage collection, referencecounting, and Unicode support. Python 3.0, released in 2008, was a majorrevision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version

2.7.18 in 2020. Python consistently ranks as one of the most popular programming languages.

5. **Python Flask:** Flask is a micro web framework written in Python. It is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party librariesprovide common functions. However, Flask supports extensions that can add application features

as if they were implemented in Flask itself. Extensions exist for object- relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Applications that use the Flask framework include Pinterest and LinkedIn.

6. **Jinja:** Jinja is a web template engine for the Python programming language. It was created by Armin Ronacher and is licensed under a BSD License. Jinja is similar to the Django template engine but provides Python-like expressions while ensuring that the templates are evaluated ina sandbox. It is a text-based template language and thus can be used to generate any markup as wellas source code. The Jinja template engine allows customization of tags, filters, tests, and global. Also, unlike the Django template engine, Jinja allows the template designer to call functions witharguments on objects. Jinja is Flask's default template engine and it is also used by Ansible, Trac, and Salt.

7. **TensorFlow:** TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training andinference of deep neural networks. TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.TensorFlow can be used in a wide variety of programming languages, most notably Python, as well as JavaScript, C++, and Java. This flexibility lends itself to a range of applications in many different sectors.

8. **Microsoft Azure:** Microsoft Azure is a cloud computing service from Microsoft. Azure offers a range of software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) options for deploying applications and services on Microsoft-managed data center infrastructure.

9. **Azure API:** Azure API is a simple JSON-based REST API primarily used to provide a central interface to create, provision and manage API for web and cloud applications and services. With Azure API Management user can; Monitor the health of APIs, identifying errors, configure throttling, rate limits and more on each API.

10. **Prisma (ORM):** Prisma is a next-generation object–relational mapper (ORM) that claims to helpdevelopers build faster and make fewer errors. Prisma takes a different approach to ORMscomparedto traditional ORMs. It uses a custom Schema Definition Language (SDL) that automatically writesmigrations and generates type-safe code.

3. RESULTS AND ISCUSSION

Final results and outcomes in terms of the developed user interface are as follows:

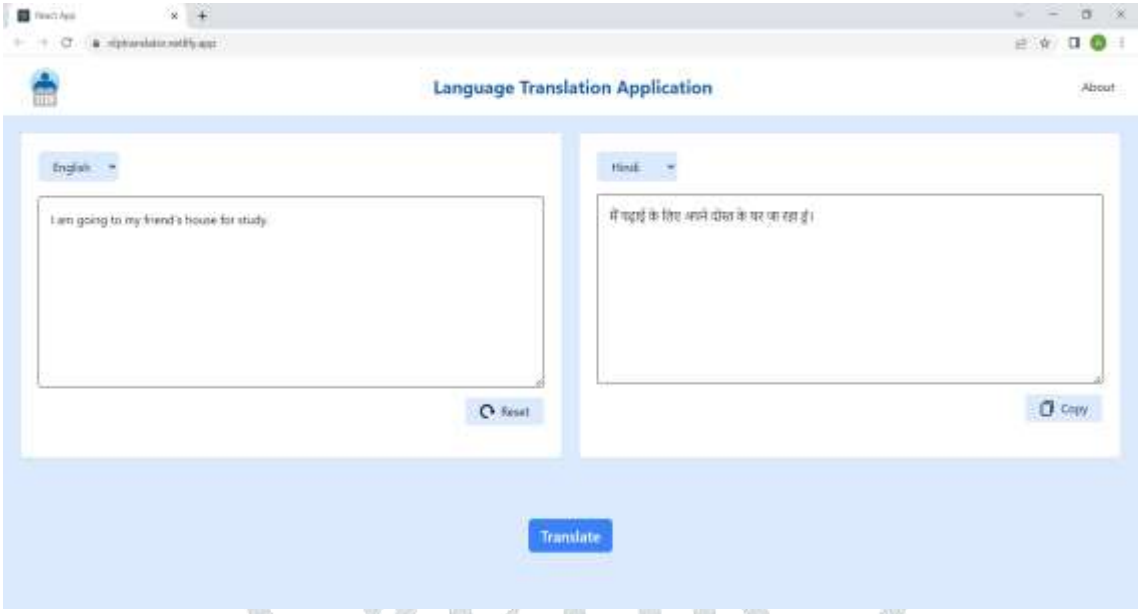

Fig4: User Interface of the developed application-English to hindi translation

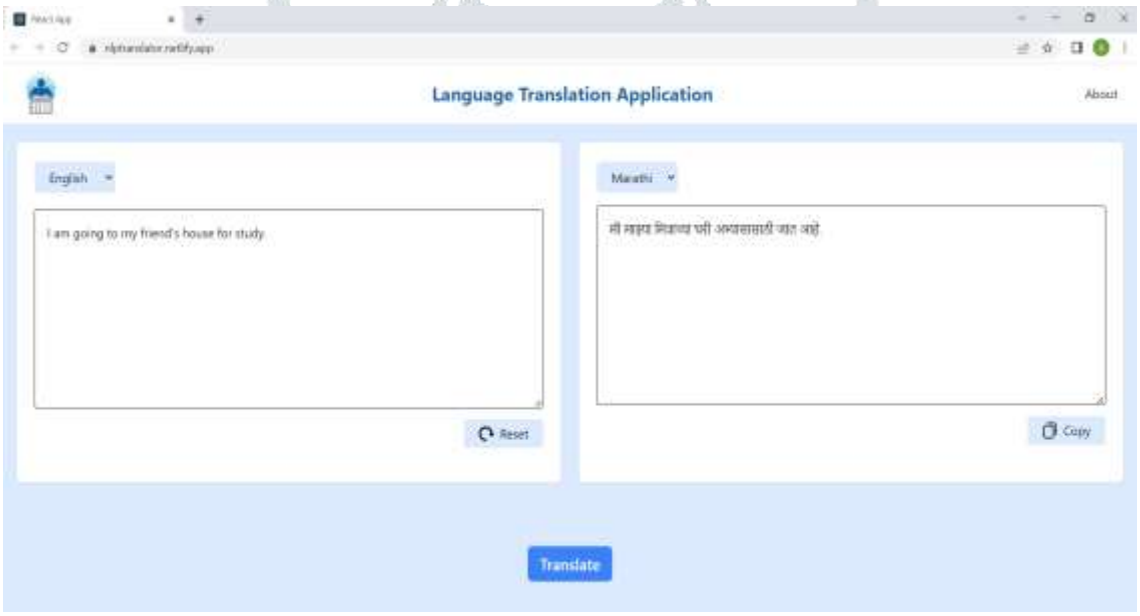

Fig5: User Interface of the developed application-English to Marathi Translation

Applying Language Understanding to Translation:

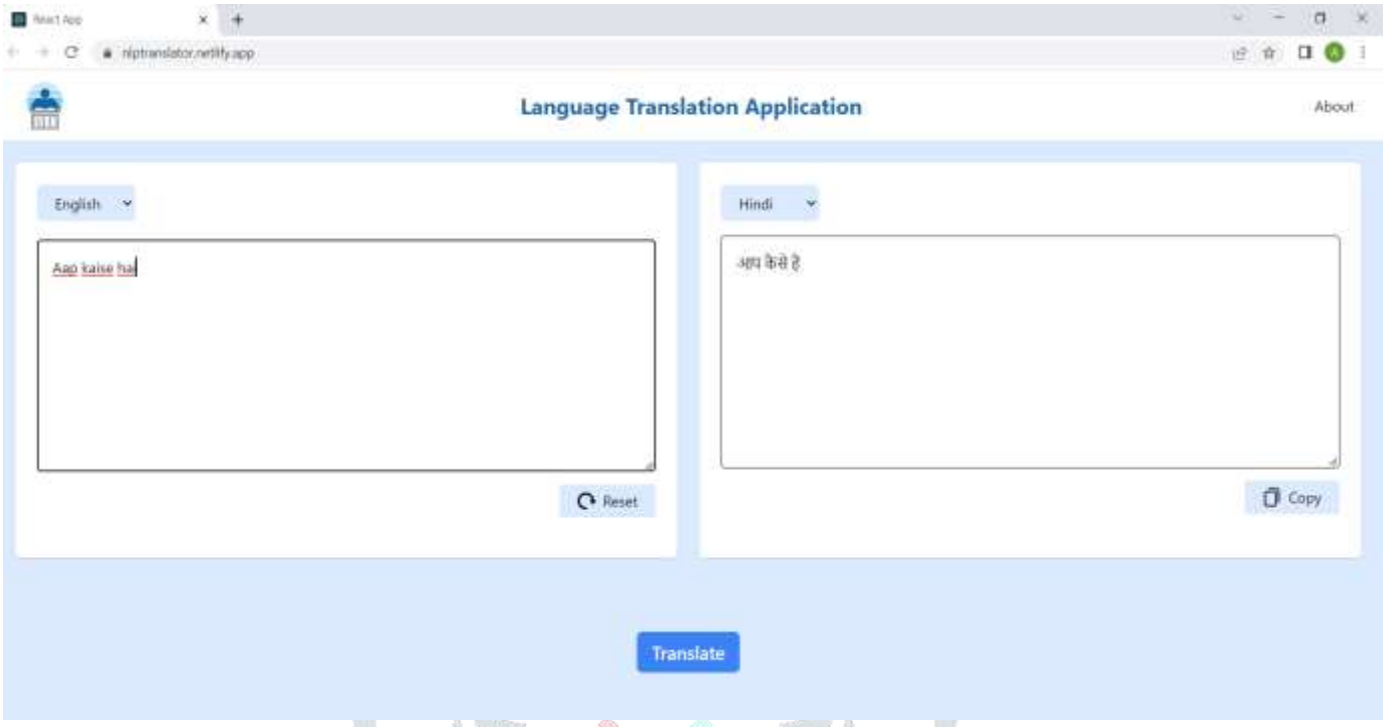

Fig6: User Interface of the developed application-Language understanding and translation

Speech to Speech Translation:

As it is an Audio formed Speech input, we can see mic below the input box and after pressing that audioinput button, at the top window we can see that red dot which indicates that system is taking voice input.

English To Hindi Speech Translation:

As it is an Audio file, we can't show or play that here that's why we are showing Screenshot of result.

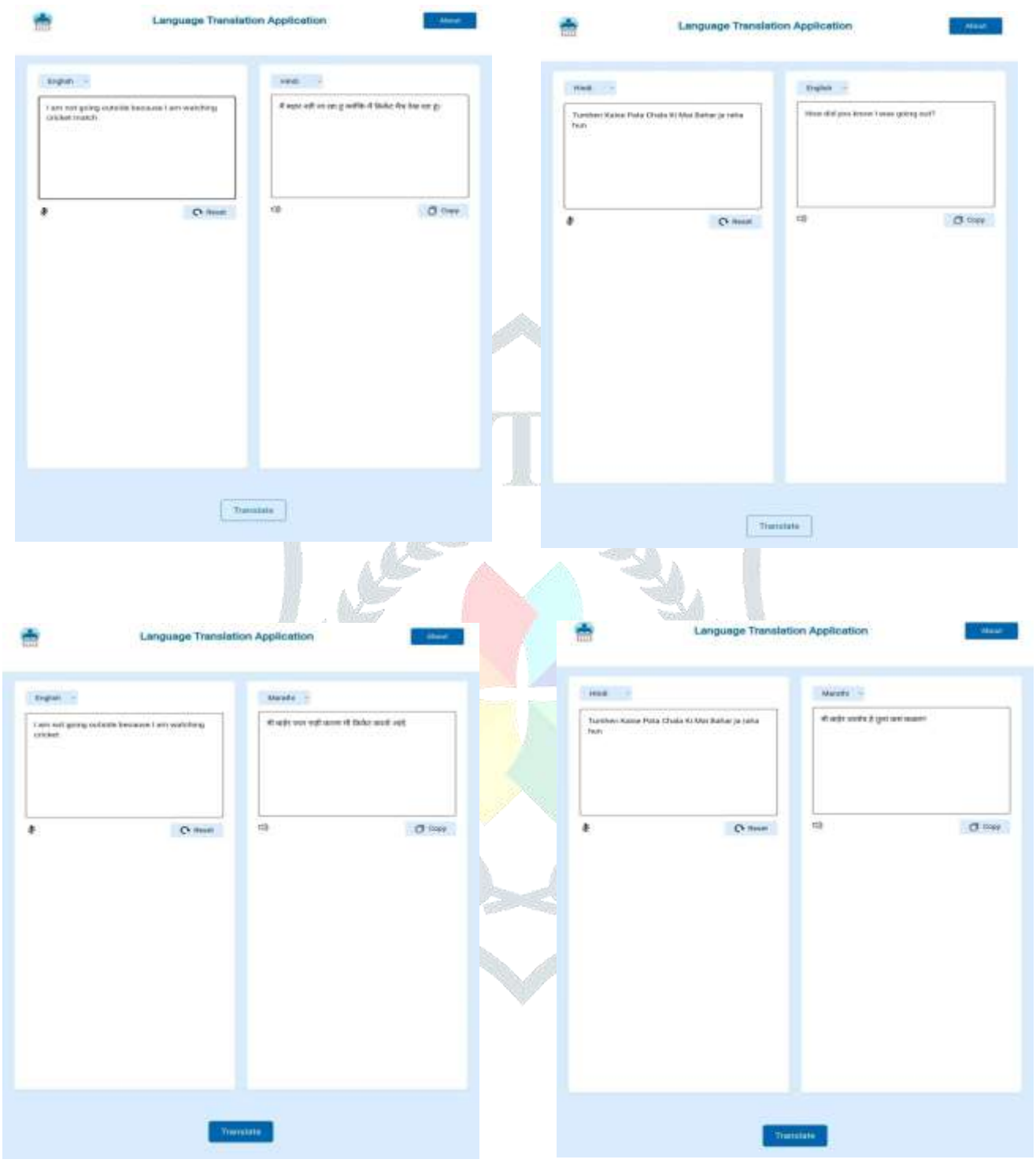

Fig10: User Interface of the developed application: Text to text translation between different languages

English To Marathi Speech Translation:

As it is an Audio file, we can't show or play that here that's why we are showing Screenshot of result.

Hindi to English Speech Translation:

As it is an Audio file, we can't show or play that here that's why we are showing Screenshot of result.

Fig12: User Interface of the developed application-Translations between different languagesHindi To Marathi Speech Translation:

As it is an Audio file, we can't show or play that here that's why we are showing Screenshot of result.

4.CONCLUSION

In this study, we have successfully developed a robust deep neural network using Azure cognitive services for a comprehensive machine translation pipeline. This system efficiently converts English text into Hindi-Marathi translations, outperforming previous architectures with lower validation loss and a test accuracy of 76.71%. We ensured reproducibility through measures like fixing random seeds, using 3-fold cross-validation, and thorough model compilation and evaluation. Our cloud-based translation system for English-Hindi-Marathi, designed for bilingual tasks, is founded on bilingual dictionaries for query translation. Additionally, our model extracts proper names and transliterations, making it versatile for languages with varying sound systems, even without pronunciation dictionaries. This work is unique in its extensive reliance on Azure Services, contributing to language translation research, particularly for English to Marathi, which is underexplored in India. The field of machine translation offers ample opportunities for further exploration and customization due to the dynamic nature of Natural Language Processing and Human Language Technology's continual evolution. Our study underscores the promise of a linguistic feature-driven Neural Machine Translation approach, especially for low-resource languages. Incorporating similar-language training data significantly improves performance, as our results demonstrate. We advocate for transformer network-based Machine Translation models in low-resource settings, emphasizing that linguistic information like morphological and part-of-speech features can enhance translation quality. We also propose using morph-based segmentation alongside byte pair encoding for languages rich in morphology.

By applying neural machine translation techniques to various Indian language pairs and rigorously evaluating them, we've laid the groundwork for cost-effective cloud-based applications, offering real-time translations. As we optimize our architectural design, the potential arises for deploying these models on embedded devices, enabling offline translation, particularly in regions with limited internet connectivity like rural India. Furthermore, the concept of a real-time speech-to-speech translation system that transcribes and vocalizes spoken language holds promise, especially for improving customer service interactions in India. This technology can aid companies in better catering to their customer base, particularly in regions with limited English proficiency.

REFERENCES

1. Nilesh Shirsath, Aniruddha Velankar, Ranjeet Patil, Dr.Shilpa Shinde, "Various Approaches of Machine Translation for Marathi to English Language", International Conference on Automation, Computing and Communication 2021. DOI: https://doi.org/10.1051/itmconf/20214003026

2. Vandan Mujadia, Dipti Misra Sharma, "English-Marathi Neural Machine Translation for LoResMT 2021".DOI: https://aclanthology.org/2021.mtsummit-loresmt.16.pdf

3. Mallamma V Reddy, Dr. M. Hanumanthappa, "NLP Challenges For Machine Translation FromEnglish To Indian Languages", International Journal Of Computer Science And Informatics: Vol. 4. DOI: 10.47893/IJCSI.2014.1168

4. Namrata G Kharate , Dr. Varsha H. Patil, "Challenges In Rule Based Machine Translation From Marathi To English", ITCSS– 2019. DOI: 10.5121/Csit.2019.91005

5. S. B. Kulkarni, P. D. Deshmukh and K. V. Kale, "Syntactic and Structural Divergence in English- toMarathi Machine Translation", IEEE 2013 International Symposium on Computational and Business Intelligence, August 24-26, 2013, New Delhi,

pp. 191-194,

DOI: 10.1109/ISCBI.2013.46

6. H. Yu, "Summarization for Internet News Based on Clustering Algorithm," 2009 International Conference on Computational Intelligence and Natural Computing, 2009, pp. 34-37,

DOI: 10.1109/CINC.2009.194.

7. K. Chen et al., "Extractive Broadcast News Summarization Leveraging Recurrent Neural Network Language Modeling Techniques," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 1322-1334, Aug. 2015, DOI: 10.1109/TASLP.2015.2432578.

8. Karthik Revanuru, Kaushik Turlapaty, Shrisha Rao, “Neural Machine Translation of IndianLanguages”, 10th Annual ACM India Compute Conference, November 2017.
DOI: https://doi.org/10.1145/3140107.3140111

9. H. Tauseef and M. Asfand-E-Yar, "Extractive Summarization of Text Using Supervised and Unsupervised Techniques," 2021 3rd International Conference on Natural Language Processing (ICNLP), 2021, pp. 37-41, DOI: 10.1109/ICNLP52887.2021.00012.

10. Rahul, S. Adhikari and Monika, "NLP based Machine Learning Approaches for Text Summarization," 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC),2020, pp. 535-538, DOI: 10.1109/ICCMC48092.2020.ICCMC- 00099.