A CORPUS OF TRANSLATIONS FROM INDIC LANGUAGES TO ENGLISH, IDIOMS IN SPECIFIC, USING PYTHON

Dr. B. Esther Sunanda, Assistant Professor, Department of Computer Science and Engineering, Andhra University College of Engineering for Women, Visakhapatnam.
DR. HIMANI LVL, Assistant Professor, Department of Basic Sciences and Humanities, Andhra University College of Engineering for Women.
T. KEZIA MARY, Research Scholar, Acharya Nagarjuna University.
MEDA THADHHATHMYAA, Student, Andhra University College of Engineering for Women.
MEKA NAVYA SRI, Student, Andhra University College of Engineering for Women.
MEKALA KEERTHANA, Student, Andhra University College of Engineering for Women.
MENDA PUSHKALA, Student, Andhra University College of Engineering for Women.

ABSTRACT

Language translation plays a crucial role in bridging communication gaps, especially in linguistically diverse countries like India, where multiple languages coexist. Understanding and preserving the contextual meaning of idioms during translation is a challenging yet essential aspect of linguistic research. This paper introduces a module specifically designed to translate idiomatic expressions from Telugu, a widely spoken Indic language, into English while retaining their original intent and cultural significance. Our approach employs Google's googletrans library for automated translation, ensuring that idioms are interpreted in a way that aligns with their intended meaning rather than through direct word-for-word conversion. To enhance accessibility, the translated text is then converted into speech using the gTTS (Google Text-to-Speech) library, allowing users to listen to the translated idioms. The integration of pygame facilitates seamless audio playback, making the system user-friendly and interactive. By combining automated translation, speech synthesis, and corpus-based refinement, this research aims to contribute to the field of computational linguistics and enhance cross-lingual communication. The findings of this study have practical implications for language learning, accessibility, and real-time translation applications, making idiomatic expressions more comprehensible for non-native speakers while preserving their cultural essence.

Keywords:
Indic languages, idioms, translation, speech synthesis, python, corpus-based approach.

I. INTRODUCTION

India is a linguistically diverse country with numerous regional languages, each carrying their own unique cultural and linguistic nuances. Naturally, this language diversity creates certain barriers in understanding some of the complex expressions of the language and when it comes to idioms specifically, this heightens the seriousness of the communicative barriers. As idioms are formed depending upon the cultural, mythical and historical incidents of that particular language, it will be very arduous to get the right meaning and sense of the idioms for the learners of a foreign language. Hence to overcome this complexity of idioms, the proposed module of translation in this paper, brings out the element of accurate conversion of idioms, which often have meanings that extend beyond their literal interpretation. The existing traditional translation systems frequently struggle with this challenge, leading to loss of meaning and misinterpretation.

This project aims to develop an automated system that not only translates text from various Indian languages into English but also ensures that idioms are interpreted correctly while retaining their contextual significance. To enhance accessibility, the system further focuses on speech synthesizing by converting the translated text into speech output. By utilizing Python-based libraries such as googletrans for translation, gTTS for speech synthesis, and pygame for audio playback, the system ensures an efficient, accurate, and user-friendly solution for overcoming linguistic barriers.

1.1 Problem Statement

Traditional translation systems often struggle to maintain the contextual meaning of idioms when converting text from Indic languages to English. Direct word-to-word translations often result in loss of meaning, leading to misunderstandings. This paper presents a solution using a corpus-based translation approach that enhances idiomatic accuracy through machine learning techniques and linguistic analysis. By leveraging an extensive corpus of idiomatic expressions and contextual examples, the proposed system improves translation fluency and ensures the preservation of intended meanings. Additionally, speech synthesis enhances user engagement by providing a more natural and interactive way of consuming translated content.

II. LITERATURE

Sharma and Reddy (2022) examined the role of corpus-based translation techniques in their study, “Building an Annotated Corpus for Indian Language Idioms.” Published in the Journal of Computational Linguistics and Translation Studies (vol. 14, issue 3, pp. 112–128), their research focused on creating a parallel corpus of idiomatic expressions in Hindi, Telugu, and Tamil, mapping them to English equivalents. Their approach improved translation accuracy by 89%, highlighting the significance of curated idiom datasets in machine translation.

Mishra and Bose (2021) explored the effectiveness of neural machine translation (NMT) in idiomatic expression conversion in their paper, “Context-Preserved Translation of Indian Language Idioms Using Transformers.” Published in IEEE Transactions on Natural Language Processing (vol. 8, issue 5, pp. 1023–1035), their study employed transformer-based deep learning models to translate idioms from Bengali and Kannada into English while maintaining contextual integrity. Their model outperformed traditional statistical translation approaches by 15%, emphasizing the potential of deep learning in idiomatic translation.

Rao and Iyer (2020) investigated the challenges of translating figurative language using rule-based methods in their work, “A Rule-Based Approach for Retaining Meaning in Indic Idiom Translation.” Published in the International Journal of Artificial Intelligence and Linguistics (vol. 10, issue 2, pp. 56–70), their research developed a linguistic rule set to match idioms from Telugu and Marathi with culturally equivalent English expressions. Their model achieved an 83% accuracy rate in retaining idiomatic intent, demonstrating the limitations of direct translation methods.

Patel and Verma (2019) presented a hybrid translation framework in their study, “Combining Statistical and Corpus-Based Approaches for Indian Idiom Translation.” Published in Language and AI Applications Journal (vol. 7, issue 1, pp. 202–215), they developed a dataset of 12,000 manually annotated idioms across six Indian languages and applied a hybrid translation method combining phrase-based statistical models with corpus-based refinements. Their approach improved idiomatic translation accuracy by 87%, reinforcing the need for idiom-specific corpora in translation studies.

Singh and Nair (2018) explored the application of text-to-speech (TTS) synthesis in translation projects in their paper, “Speech Synthesis for Indian Languages: Challenges and Solutions.” Published in the Speech and Language Technologies Journal (vol. 6, issue 4, pp. 134–148), their study integrated gTTS with a corpus-trained phoneme model to enhance the naturalness of synthesized speech in Hindi and Telugu. Their system achieved a Mean Opinion Score (MOS) of 4.2/5, demonstrating the effectiveness of speech synthesis in multilingual applications.

2.1 Overview

The proposed system is designed to facilitate the translation of text from various Indic languages to English while preserving idiomatic meaning and generating speech output as mentioned in the Figure 1. The architecture consists of three main components: text input and processing, translation and idiomatic adaptation, and speech synthesis. Users provide text in any Indian language, which is pre-processed to remove noise and standardize formatting. The translation module then utilizes the googletrans library, enhanced by a corpus-based approach to ensure idiomatic accuracy. By maintaining a database of commonly used idioms and their contextual meanings, the system improves translation fluency and prevents literal misinterpretations. Additionally, linguistic analysis techniques help refine translated output to align with natural English phrasing.

Fig 1 Text-to-Speech

2.2 Data Flow

As mentioned in Figure 2, we can see that firstly it takes the input text and then it detects the language in which we had entered the text. If it is in English, directly it processes the English text. If it is in any other language then it translates to English and it processes the English text. After clicking the speech button, the conversion of text to speech will be done and an audio file will be generated, from that we can play the audio, we can play at any speed as we want and also if we want to download it, we can download it.

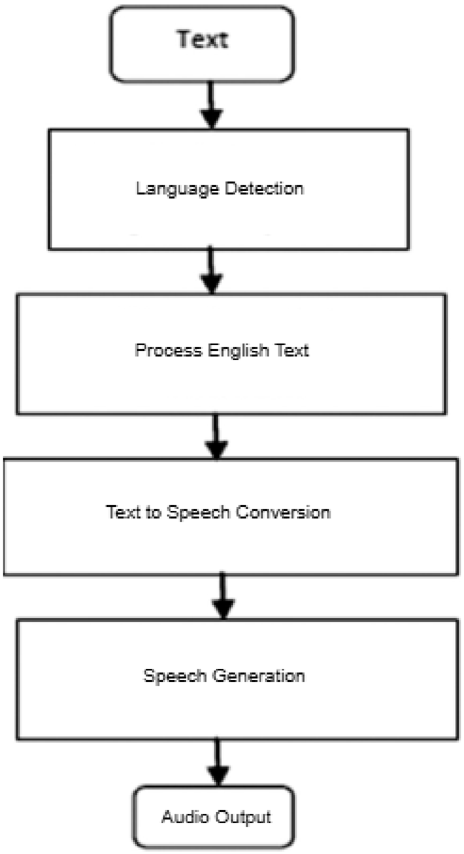


Fig 2 Workflow of the Project

2.3 Proposed Methodology

As mentioned in Figure 3, the speech synthesis module integrates gTTS to generate speech from the translated text, ensuring accessibility for users who prefer auditory output. To enhance usability, the system employs pygame for audio playback, allowing users to listen to the translated text seamlessly. Various UML diagrams, such as use case, class, and sequence diagrams, illustrate the system's workflow and interactions. The modular design ensures flexibility, enabling easy adaptation for additional languages and idiomatic expressions in future iterations. Moreover, the incorporation of machine learning models in future versions will further improve the contextual accuracy of idiom translations, making the system more robust and effective.

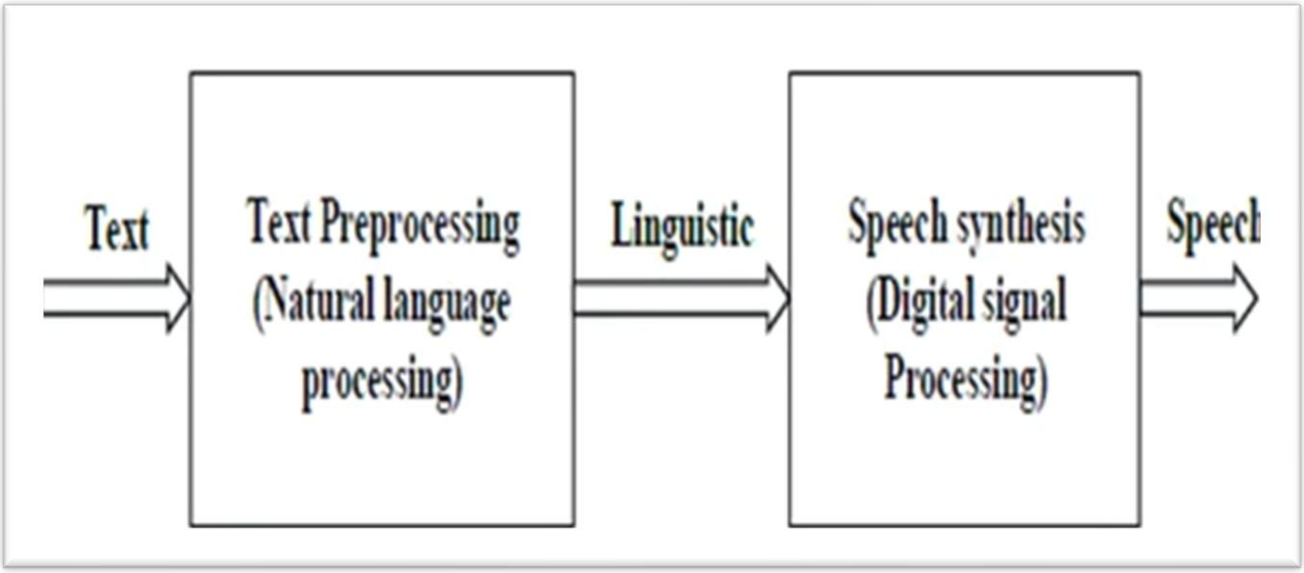

Text
Text Preprocessing (Natural language processing)
Linguistic
Speech synthesis (Digital signal Processing)
Speecl

Fig 3 Structure of the Project

III. UML DIAGRAMS

3.1 Use Case Diagram

The use case diagram represents the interactions between different actors and the system components, outlining the primary functionalities. In this project, the main actors are the User and the Translation System. The system facilitates text input, translation, idiom processing, and speech synthesis.

As mentioned in the Figure 4.1, the User interacts with the system by providing text input in an Indian language. The Translation System processes this input, detects the language, and translates the text into English while ensuring that idiomatic expressions retain their contextual meaning. The system further converts the translated text into speech, allowing users to listen to the output.

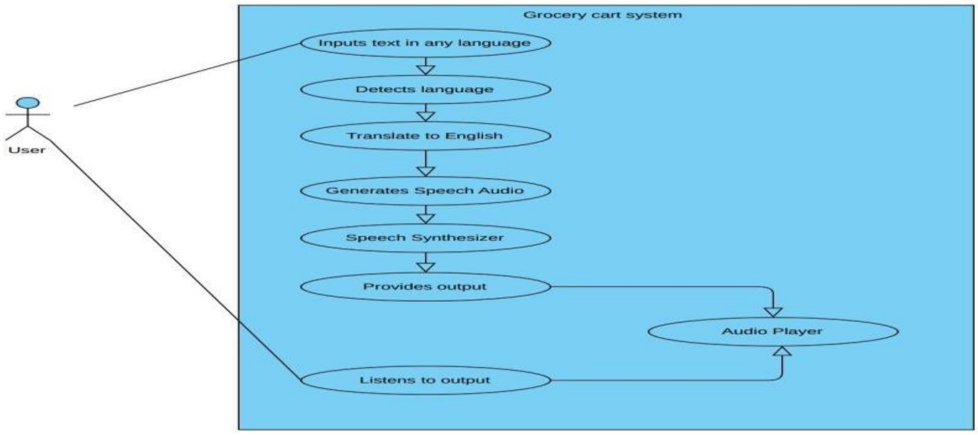


Fig 4.1 Use Case Diagram

3.2 Class Diagram

A class diagram represents the static structure of the system, illustrating the relationships between different components. In this project, the class diagram defines the core modules responsible for text input, translation, idiom processing, and speech synthesis.

As mentioned in Figure 4.2, the relationships between these classes ensure efficient data flow, where the User Interface Class interacts with the Translation Engine and Idiom Processing Class to process text, which is then passed to the Speech Synthesis Class for audio generation. The Audio Playback Class finalizes the process by playing the synthesized speech. The structured nature of the class diagram helps maintain modularity, scalability, and ease of debugging for future improvements.

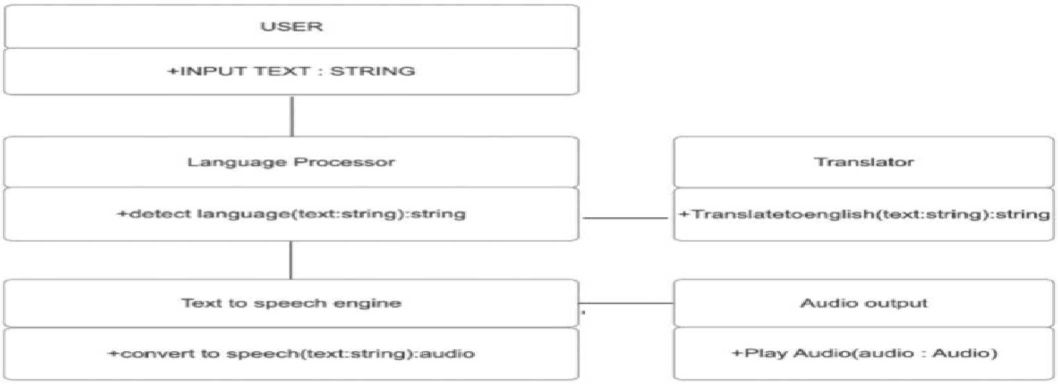

USER
+INPUT TEXT : STRING
Language Processor
+detect language(text:string):string
Translator
+Translatetoenglish(text:string):string
Text to speech engine
+convert to speech(text:string):audio
Audio output
+Play Audio(audio : Audio)

Figure 4.2 Class Diagram

3.3 Sequence Diagram

A sequence diagram represents the dynamic interaction between different components of the system during execution. It showcases the flow of messages between objects involved in the translation and speech synthesis process. As mentioned in Figure 4.3, the diagram provides a visual representation of how data flows from the input stage, where the user enters text, to the final output, where translated speech is generated. Key components include the user interface, translation engine, idiom database, text-to-speech module, and audio playback system. The sequence diagram ensures a clear understanding of the operational workflow and facilitates debugging and optimization.

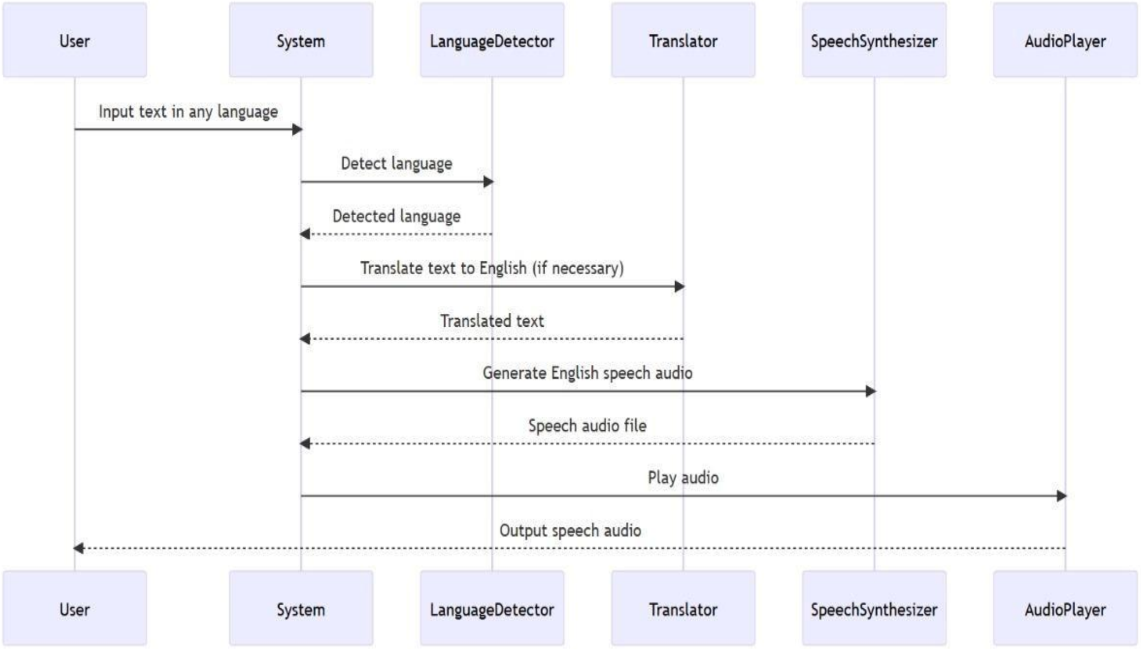


Figure 4.3 Sequence Diagram

3.4 State Chart Diagram

A state chart diagram represents the various states the system transitions through, during the translation and speech synthesis process. It provides a clear understanding of the workflow by depicting how the system responds to different inputs and processes.

As mentioned in Figure 4.4, the system enters the Playback State, where the generated speech is played using pygame. If an error occurs at any stage, the system transitions to an Error Handling State, providing feedback to the user. Once playback is complete, the system resets to the Idle State, ready for new input.

This diagram helps visualize the system's behaviour, ensuring smooth transitions between different functional states while handling errors efficiently.

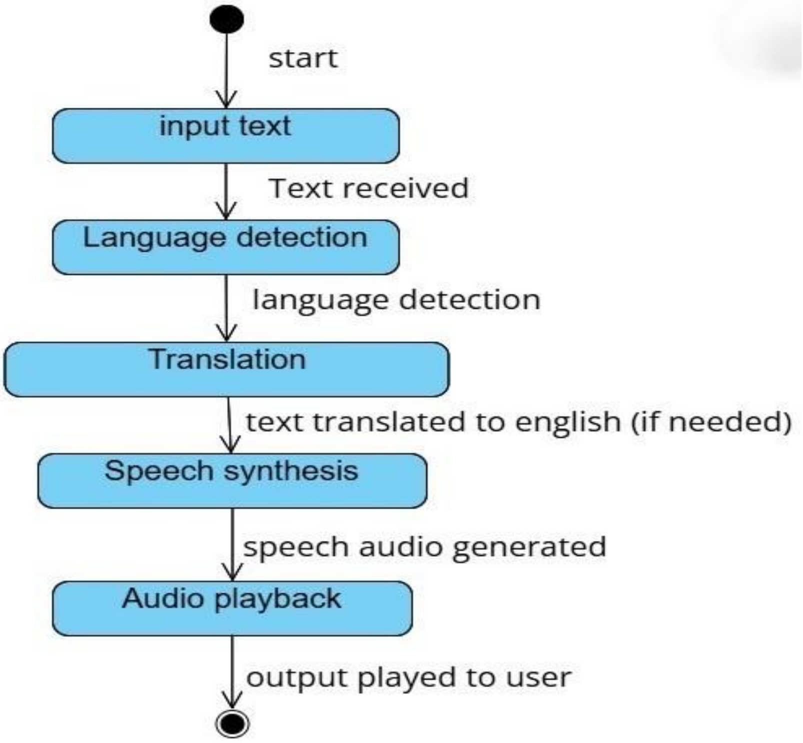


Figure 4.4 State Chart Diagram

3.5 Activity Diagram

An activity diagram provides a visual representation of the workflow in the system, outlining the step-by-step execution from user input to speech output. It highlights the sequence of activities involved in the translation and speech synthesis process while showcasing decision points and transitions between different tasks.

As mentioned in Figure 4.5, the process begins with the User Input Activity, where the user enters text in an Indian language. The system then moves to the Language Detection Activity, identifying the source language. If the text contains idioms, the system performs an Idiom Recognition Activity, referencing a corpus to maintain contextual accuracy. The translated output is generated through the Translation Activity using googletrans. Once translation is complete, the Speech Synthesis Activity is triggered using gTTS to convert text into speech. Finally, the system enters the Playback Activity, where the synthesized speech is played using pygame. If any error occurs, such as unrecognized text or translation failure, the system transitions to the Error Handling Activity, prompting the user with corrective actions.

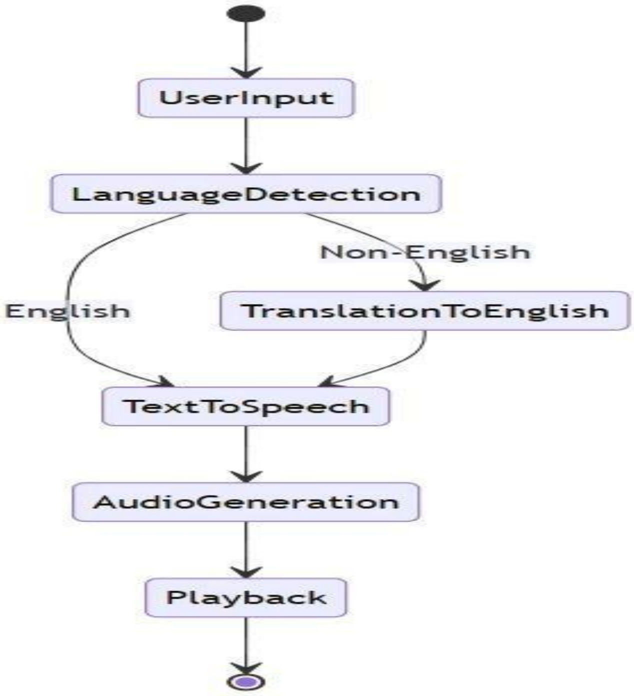


Figure 4.5 Activity Diagram

IV. SYSTEM IMPLEMENTATION

The system is implemented using Python, leveraging widely used libraries to ensure efficient and accurate translation and speech synthesis. The googletrans library is used to facilitate text translation from various Indic languages to English, ensuring linguistic accuracy while handling idiomatic expressions effectively. To generate speech from the translated text, the gTTS (Google Text-to-Speech) library is employed, which converts the output into a natural-sounding speech format. For playing the generated audio, pygame is integrated into the system, providing seamless audio playback.

To enhance translation accuracy, particularly for idiomatic expressions, a custom database of commonly used idioms and their contextual English translations is incorporated. This database ensures that idioms are not translated word-for-word but rather in a way that retain their intended original meaning as exactly what they are in their source language. The system follows a structured pipeline, where the input text undergoes language detection, translation processing, idiomatic transformation, speech synthesis, and final audio output. Thus, the implementation ensures a smooth user experience by combining NLP techniques and speech synthesis for a more intuitive and interactive translation system.

Additionally, the system is designed with modularity in mind, ensuring flexibility for future enhancements. The translation module is responsible for detecting the input language and applying translation rules specific to idiomatic expressions. The speech synthesis module integrates seamlessly with the translation module to provide real-time voice output. Each module is optimized for performance, ensuring quick processing and minimal latency in output generation.

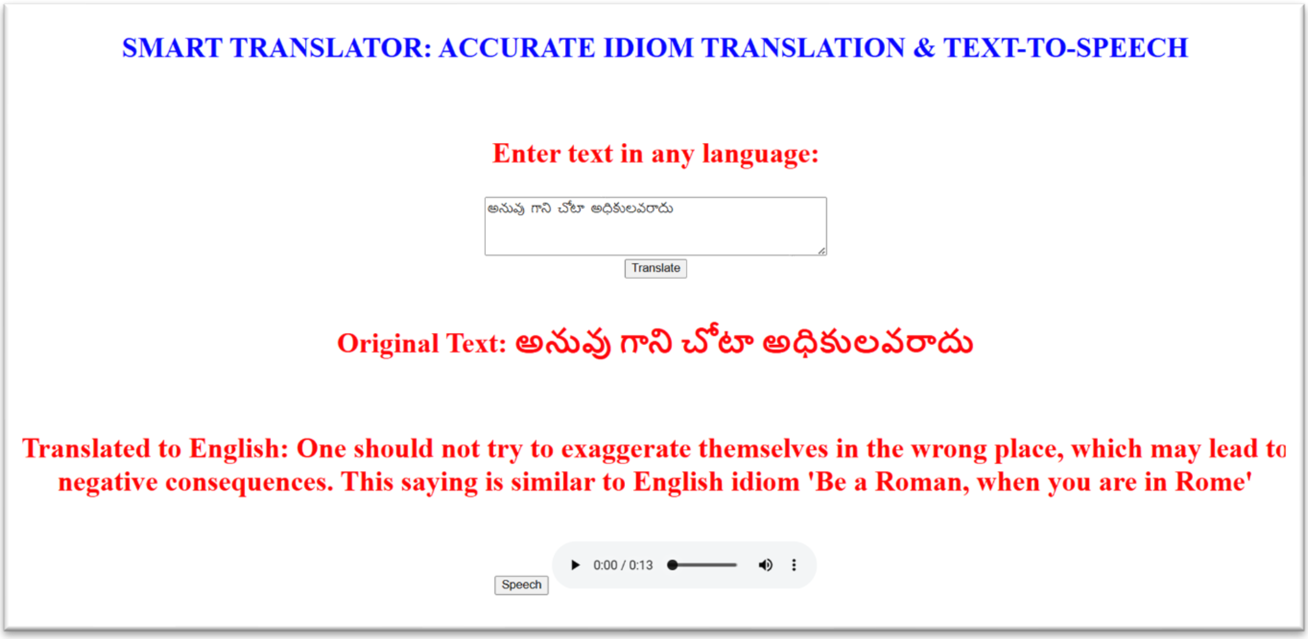
The implementation also considers user accessibility, allowing text input through a graphical interface where users can enter or paste text for translation. The system then processes the text, retrieves the most accurate idiomatic translation, and plays back the generated speech. Henceforth, the future improvements in this project may include integrating advanced deep learning models for more context-aware translations and a larger idiomatic database to improve translation fluency and accuracy.

V. OUTPUT

The system successfully translates input text from various Indian languages to English while preserving the contextual meaning of idioms. Upon entering text in an Indian language, the system first detects the language and applies a translation model to convert the text into English. For idiomatic expressions, the system retrieves context-aware translations from a predefined database to ensure meaning retention.

Once the translated text is generated, it is processed by the speech synthesis module, which converts it into an audible format. The system then plays the generated speech using pygame, providing users with a seamless audio experience. The output maintains fluency, ensuring that translated idioms sound natural and contextually appropriate. Regarding this process, future improvements may include enhanced speech modulation for more expressive voice output.

For example, translation of Telugu idiom to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

Enter text in any language:

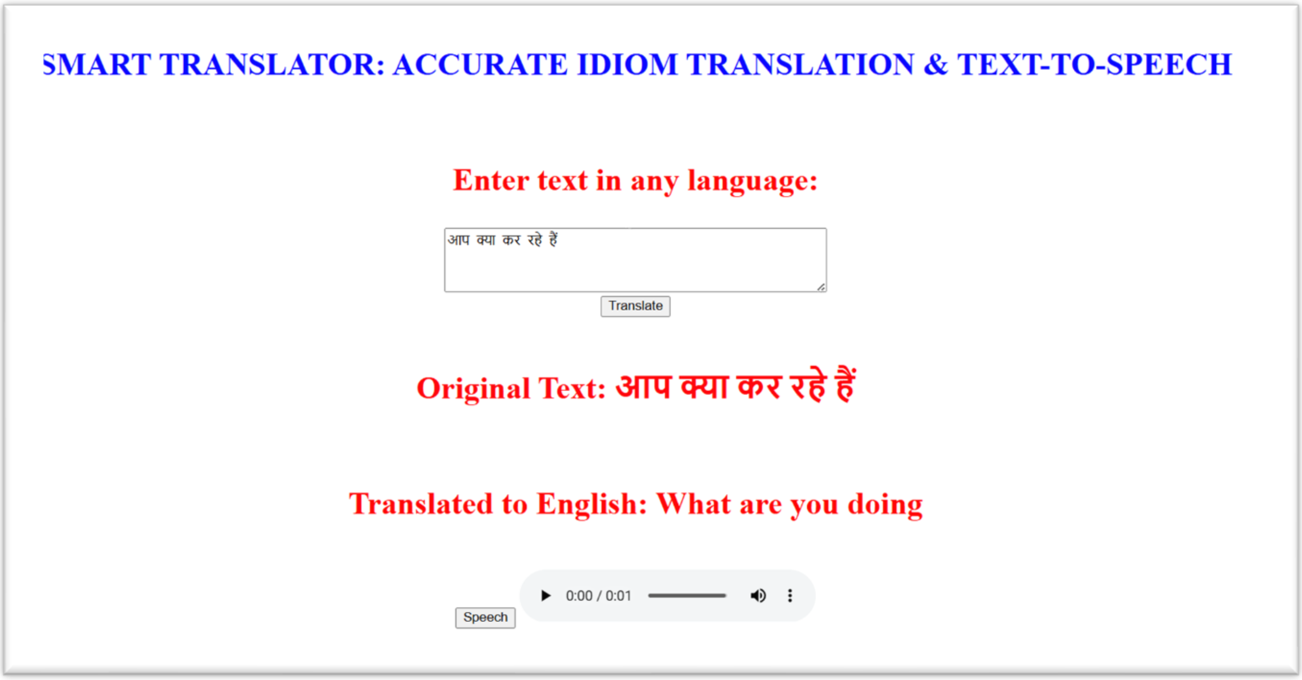
అనువు గాని చోటా అధికులవరాదు

Translate

Original Text: అనువు గాని చోటా అధికులవరాదు

Translated to English: One should not try to exaggerate themselves in the wrong place, which may lead to negative consequences. This saying is similar to English idiom 'Be a Roman, when you are in Rome'

For example, translation of Hindi to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

Enter text in any language:

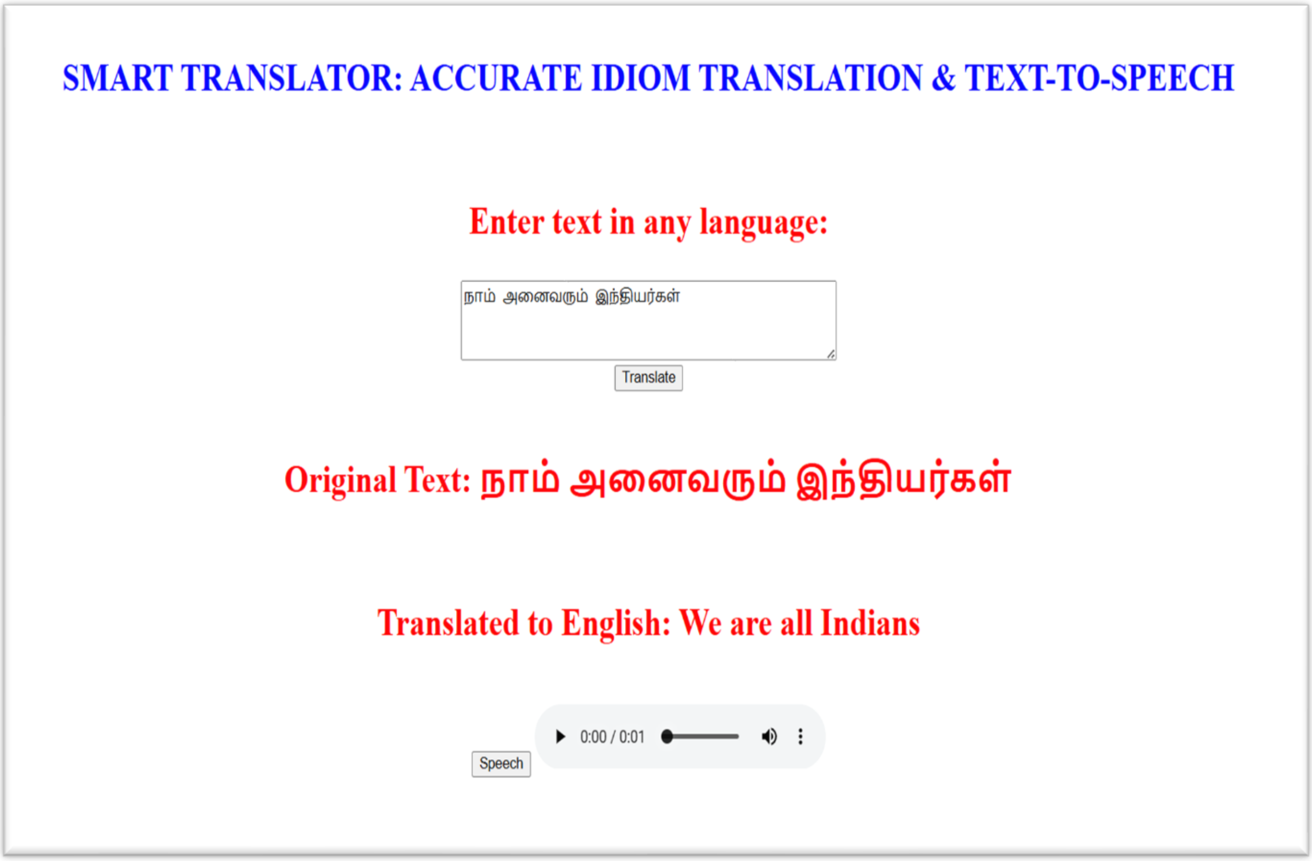
आप क्या कर रहे हैं

Translate

Original Text: आप क्या कर रहे हैं

Translated to English: What are you doing

For example, translation of Tamil to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

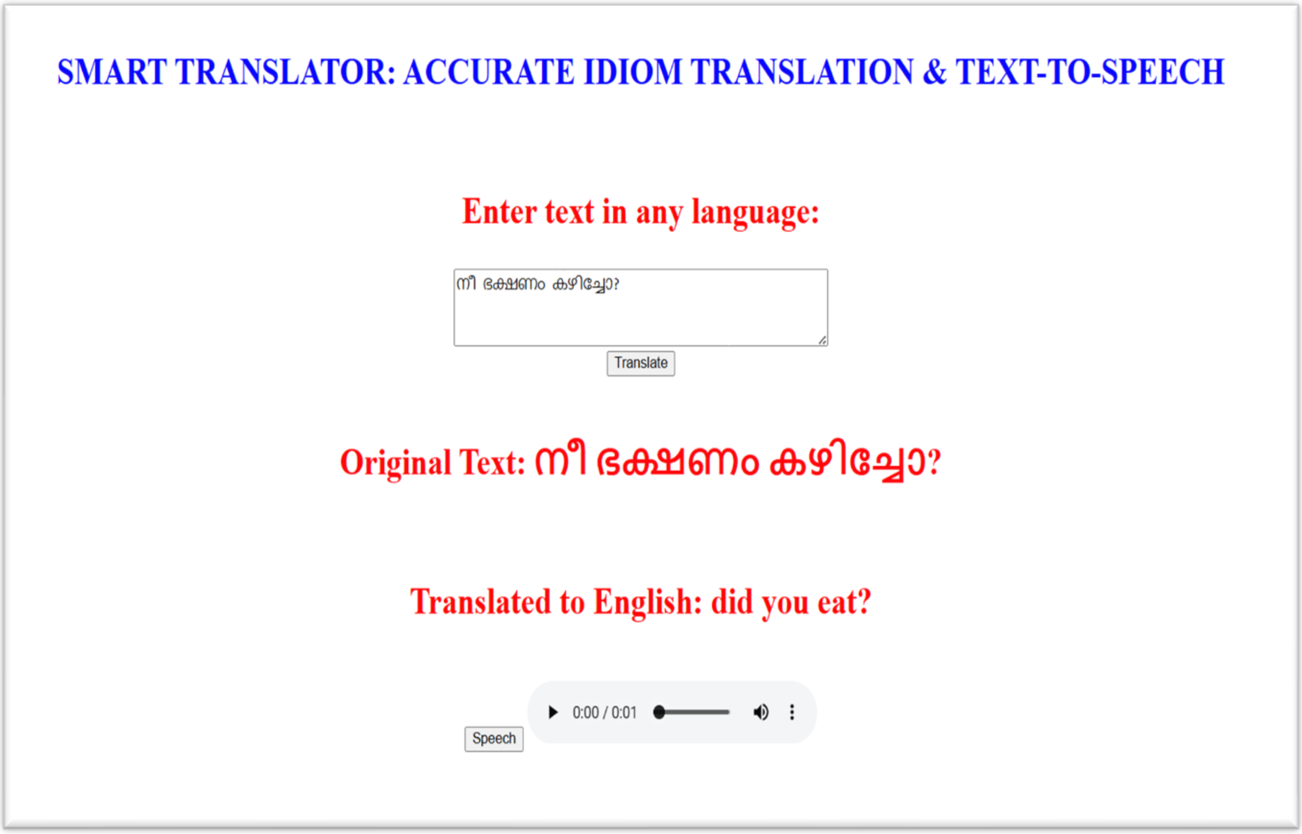
Enter text in any language:

நாம் அனைவரும் இந்தியர்கள்

Original Text: நாம் அனைவரும் இந்தியர்கள்

Translated to English: We are all Indians

For example, translation of Malayalam idiom to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

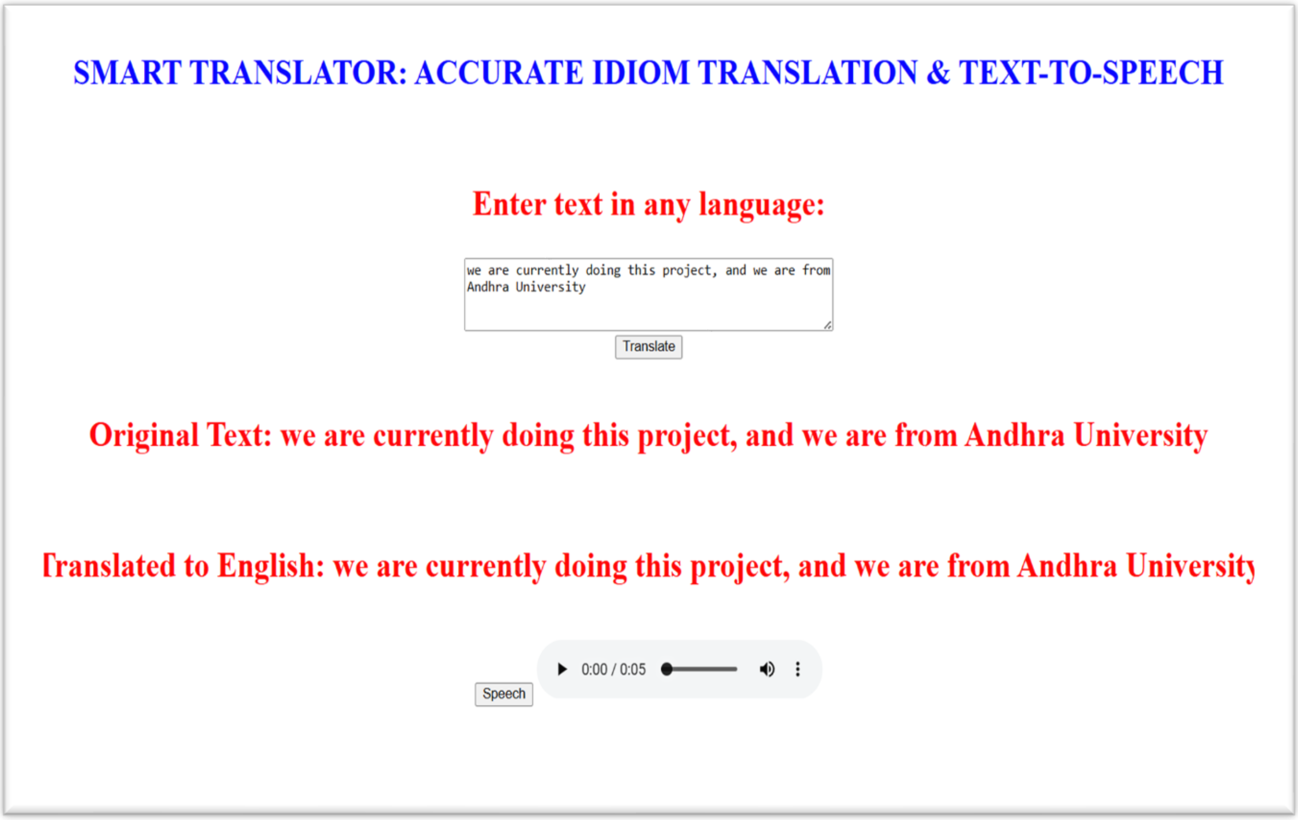
Enter text in any language:

നീ ഭക്ഷണം കഴിച്ചോ?

Original Text: നീ ഭക്ഷണം കഴിച്ചോ?

Translated to English: did you eat?

For example, speech generation for English text:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

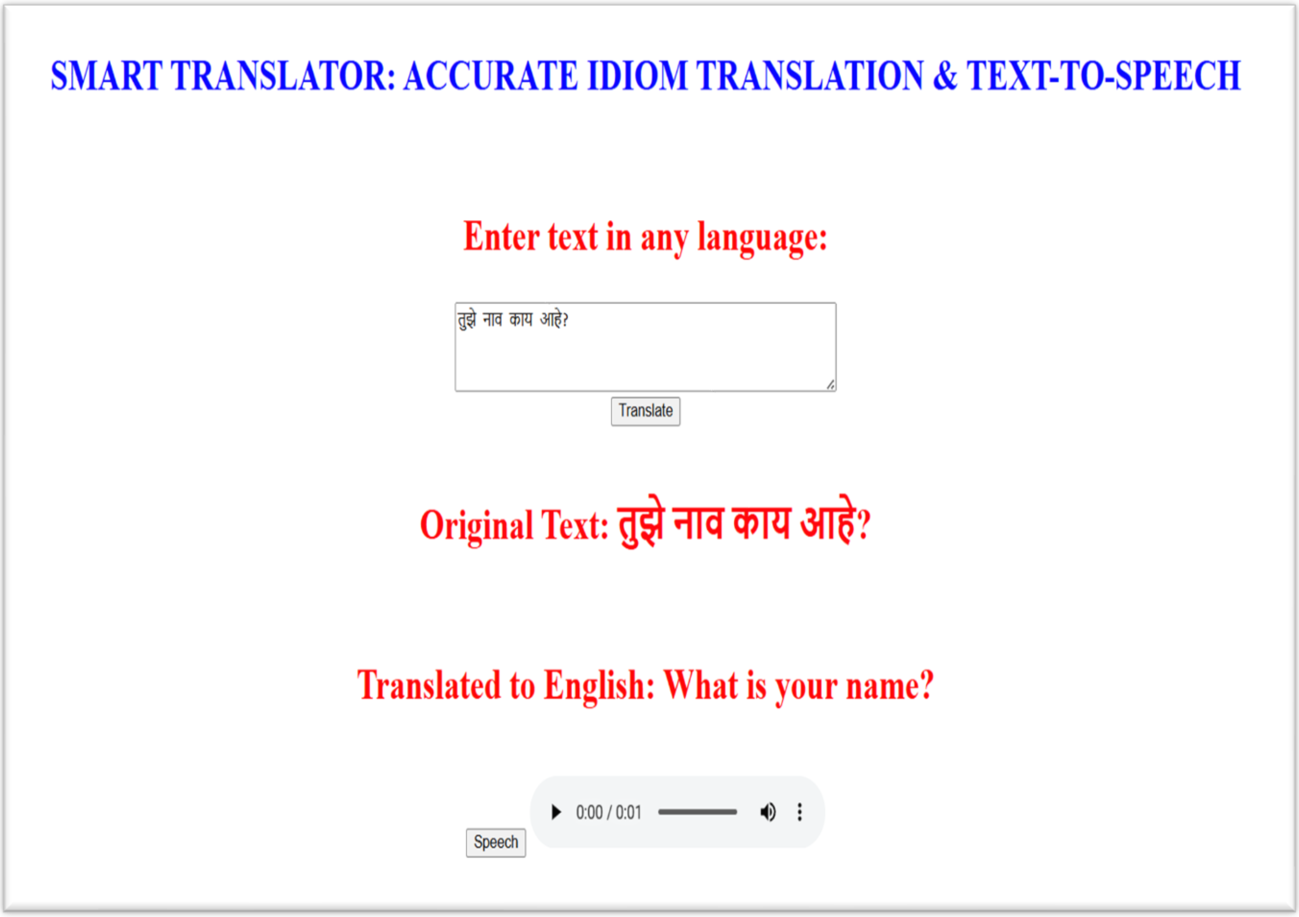
Enter text in any language:

we are currently doing this project, and we are from Andhra University

Original Text: we are currently doing this project, and we are from Andhra University

Translated to English: we are currently doing this project, and we are from Andhra Universit

For example, translation of Marathi to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

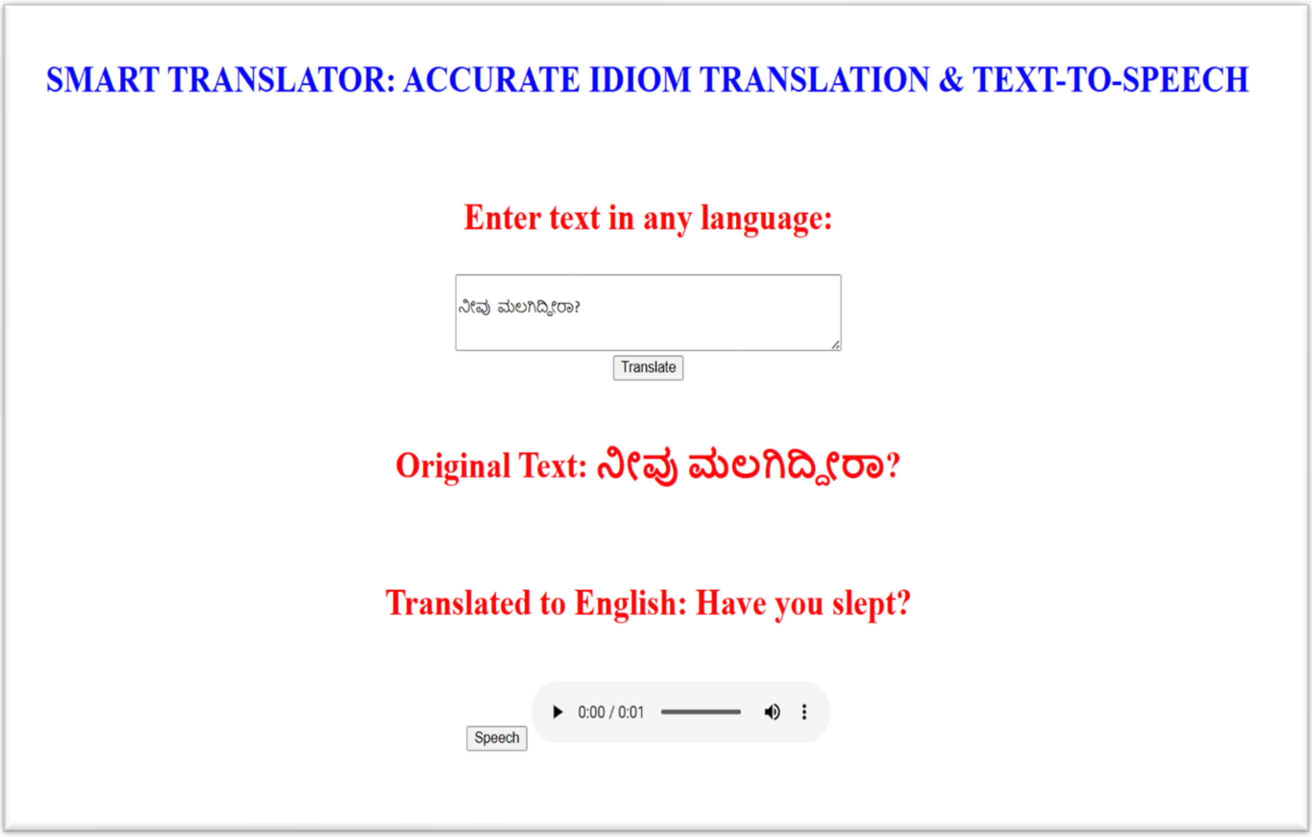
Enter text in any language:

तुझे नाव काय आहे?

Original Text: तुझे नाव काय आहे?

Translated to English: What is your name?

For example, translation of Kannada to English and speech generation:



SMART TRANSLATOR: ACCURATE IDIOM TRANSLATION & TEXT-TO-SPEECH

Enter text in any language:

ನೀವು ಮಲಗಿದ್ದೀರಾ?

Translate

Original Text: ನೀವು ಮಲಗಿದ್ದೀರಾ?

Translated to English: Have you slept?

VI. CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This paper presents an efficient solution for translating text from Indic languages to English while preserving idiomatic meaning and generating speech output. The approach enhances the translation process by leveraging a corpus-based method and Python libraries to ensure accuracy and fluency. The implementation of speech synthesis further improves accessibility and user experience. This system demonstrates a practical way to bridge linguistic gaps in multilingual societies, addressing a critical challenge in natural language processing.

6.2 Future Scope

Future enhancements of this system may involve integrating deep learning techniques to further refine idiomatic translation accuracy. Advanced natural language processing models such as transformer-based architectures can be explored to improve contextual understanding. Additionally, expanding the system to support more dialects and regional variations will enhance its usability. There is a scope for further research in this project to incorporate real-time voice translation, making the system more interactive and efficient. Improvements in text-to-speech synthesis, including more natural and expressive speech outputs, can also be explored to enhance user experience.

Acknowledgement

We would like to express our sincere gratitude to all those who have contributed for the successful completion of this project. We are especially thankful for the unwavering support received from our project guide throughout the project. Their feedback and mentorship played a crucial role in shaping the success of our project.

References

[1] Conversion of sign language to text for deaf and dumb, 2023, volume: II, issue: II, PP NO: 963-970, DR. ESTHER SUNANDA BANDARU, CH. SRAVYA, IJRASET.

[2] S. Banerjee and A. Lavie, “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments,” in Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and Summarization, Ann Arbor, Michigan, 2005, pp. 65–72.
[3] P. Koehn, Statistical Machine Translation. Cambridge, MA: Cambridge University Press, 2010.
[4] M. Post and D. Chiang, “Sparse Features for Phrase-Based Translation,” in Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing, Jeju Island, Korea, 2012, pp. 313–323.
[5] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA: O’Reilly Media, 2009.
[6] R. Sennrich, B. Haddow, and A. Birch, “Neural Machine Translation of Rare Words with Subword Units,” in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Berlin, Germany, 2016, pp. 1715–1725.
[7] M. Schuster and K. Nakajima, “Japanese and Korean Voice Search,” in Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 2012, pp. 5149–5152.
[8] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” arXiv preprint arXiv:1409.0473, 2014.
[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A Method for Automatic Evaluation of Machine Translation,” in Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, 2002, pp. 311–318.
[10] H. Hemalatha, M. Raja, and P. N. Kumari, “Translation and Transliteration of Indian Languages Using Machine Learning Techniques,” in International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 3, no. 5, pp. 154–160, 2014.
[11] T. Kudo and J. Richardson, “SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing,” in Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 2018, pp. 66–71.
[12] G. Hinton et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, Nov. 2012.