AMRITA
VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

19EAC386 Open Lab
Final Report

Submitted by

BL.EN.U4EAC22025	K. Tappan Chengappa
BL.EN.U4EAC22026	Krishna P. T.
BL.EN.U4EAC22028	Krithik Ramesh Iyer
Guided by:	
Dr. Sreeja Kochuvila	
Examiner 1	**Examiner 2**
Dr. Pooja Kenchetty	Dr. Lalitha S.

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMPUTER ENGINEERING

AMRITA SCHOOL OF ENGINEERING, BANGALORE

AMRITA VISHWA VIDYAPEETHAM

BENGALURU-560035

Contents

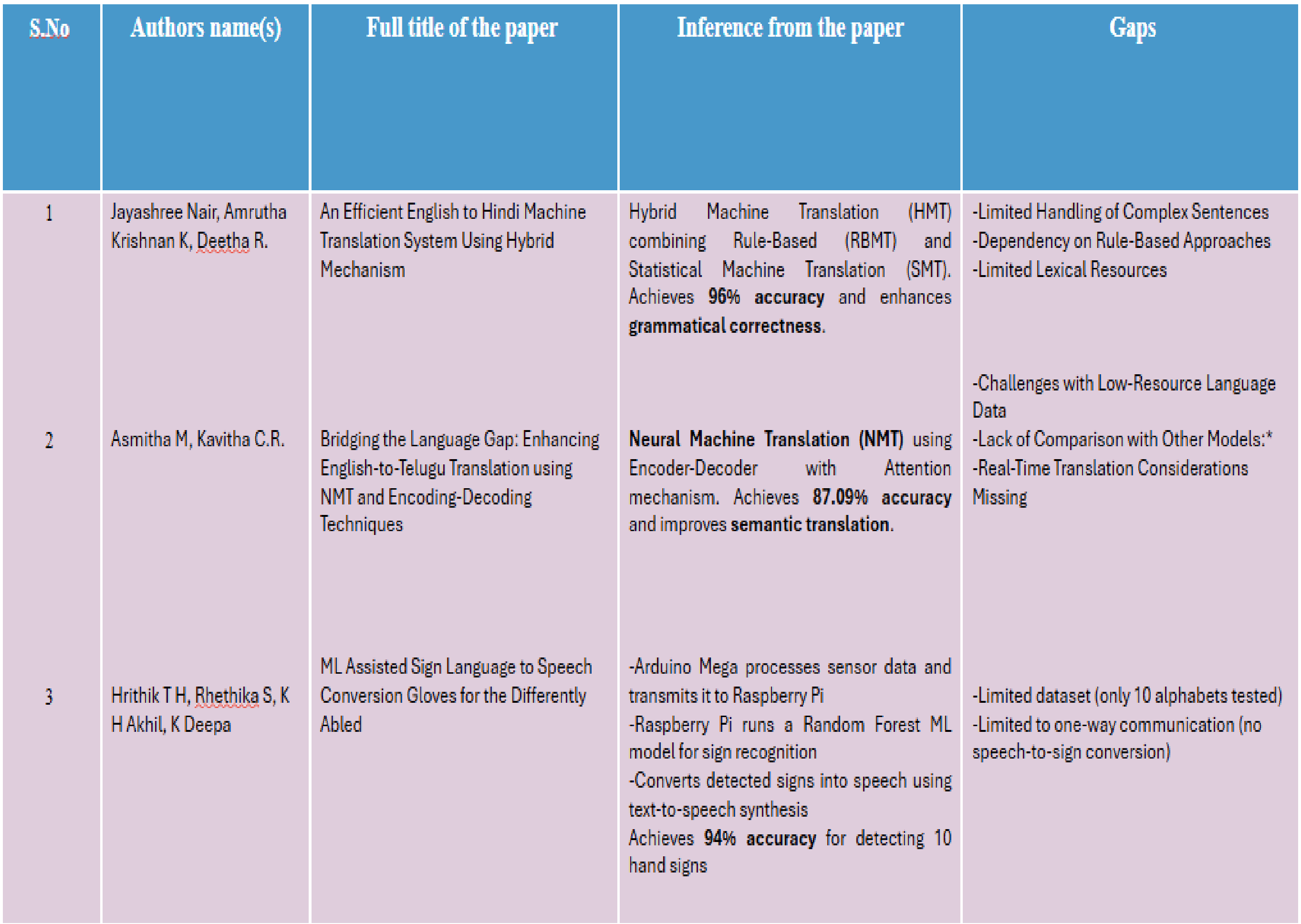
Introduction

India has many languages, and it is difficult to communicate with everyone by learning their language, especially in remote places with a lack of internet. It is also difficult to use Google Translate. Moreover, the current translators need the internet and don’t have a proper audio output for cheap prices, just text.

So, the solution aims to bridge the language gap between mainstream languages and remote languages (tribal languages).

This project aims to bridge language barriers and connect people especially remote people. The project aligns with SDG 10 which is to reduce inequalities, SDG 16 which is peace, justice and strong institutions.

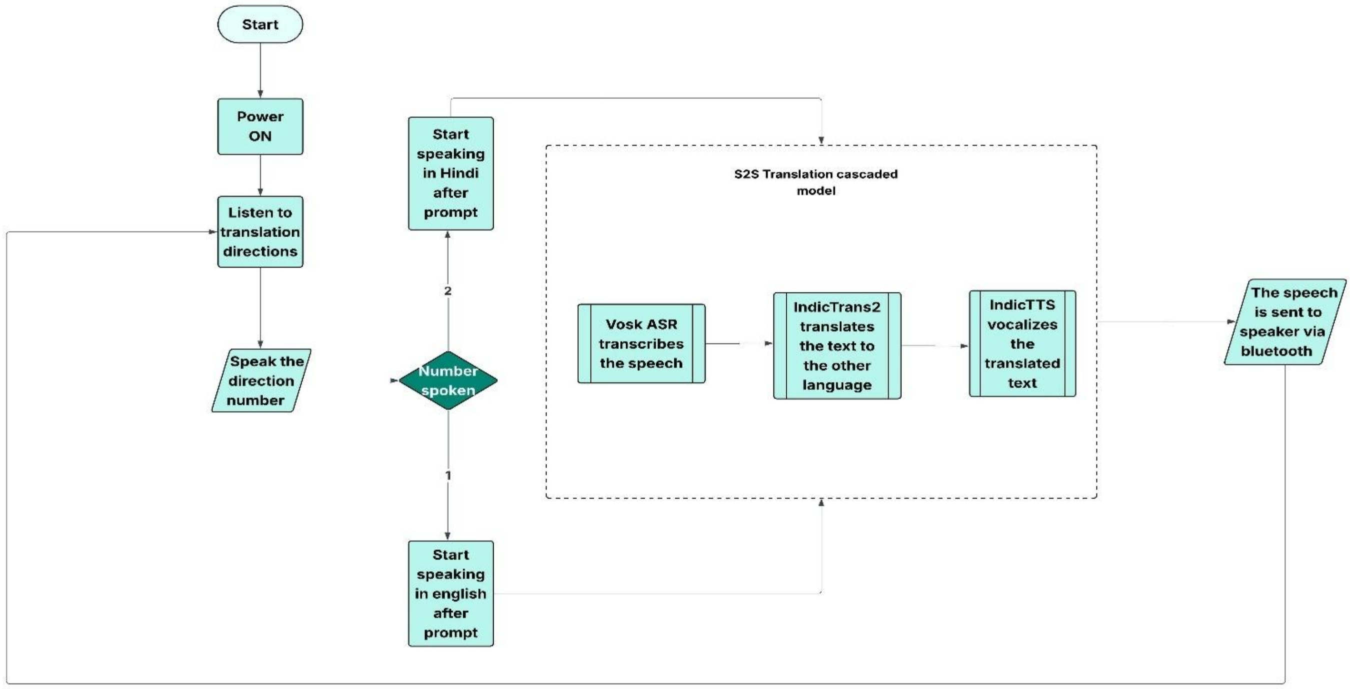
Literature Survey



S.No	Authors name(s)	Full title of the paper	Inference from the paper	Gaps
1	Jayashree Nair, Amrutha Krishnan K, Deetha R.	An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism	Hybrid Machine Translation (HMT) combining Rule-Based (RBMT) and Statistical Machine Translation (SMT). Achieves **96% accuracy** and enhances **grammatical correctness**.	-Limited Handling of Complex Sentences -Dependency on Rule-Based Approaches -Limited Lexical Resources
2	Asmitha M, Kavitha C.R.	Bridging the Language Gap: Enhancing English-to-Telugu Translation using NMT and Encoding-Decoding Techniques	**Neural Machine Translation (NMT)** using Encoder-Decoder with Attention mechanism. Achieves **87.09% accuracy** and improves **semantic translation**.	-Challenges with Low-Resource Language Data -Lack of Comparison with Other Models:* -Real-Time Translation Considerations Missing
3	Hrithik T H, Rhethika S, K H Akhil, K Deepa	ML Assisted Sign Language to Speech Conversion Gloves for the Differently Abled	-Arduino Mega processes sensor data and transmits it to Raspberry Pi -Raspberry Pi runs a Random Forest ML model for sign recognition -Converts detected signs into speech using text-to-speech synthesis Achieves **94% accuracy** for detecting 10 hand signs	-Limited dataset (only 10 alphabets tested) -Limited to one-way communication (no speech-to-sign conversion)

S.No	Authors name(s)	Full title of the paper	Inference from the paper	Gaps
4	H D Nandish, Rahul R, T K Luchingba, Neelima N	A Novel Implementation of a Cohesive Regional Language Tweet Translator	Covers **comparative analysis of TTS and STT models**, focusing on **speech-based translation** and evaluation using **BLEU, MOS, and MSE**.	Limited Language Coverage: The system primarily focuses on Indian regional languages and English, potentially excluding other languages and dialects that are not covered by the Indic-Transliteration library.
5	Surabhi Sudhan, Parvathy P Nair, MG Thushara	Text-to-Speech and Speech-to-Text Models: A Systematic Examination of Diverse Approaches	Reviews **TTS/STT models** for **assistive tech and language translation**, comparing **deep learning vs. rule-based approaches**.	-Limited Language Coverage -Many models are primarily evaluated on English datasets, limiting their applicability to other languages, especially low-resource languages.
6	L. Wijerathna, W.L.S.L. Somaweera, S.L. Kaduruwana, Y.V. Wijesinghe, D.I. De Silva, K. Pulasinghe, S. Thellijjagoda	A Translator from Sinhala to English and English to Sinhala (SEES)	Rule-based machine translation system Tokenization for **breaking sentences** into components. Rule-based translation engine using a knowledge base. **Parallel processing** with threading for performance optimization. Handles active and passive voice. Uses **C#.NET**, **SQL** 2005	-Cannot handle ambiguous words perfectly without human input . - Limited vocabulary and grammar support. -Focused only on Sinhala and English

Design


Start
Power ON
Listen to translation directions
Speak the direction number
Number spoken
2
1
Start speaking in Hindi after prompt
Start speaking in english after prompt
S2S Translation cascaded model
Vosk ASR transcribes the speech
IndicTrans2 translates the text to the other language
IndicTTS vocalizes the translated text
The speech is sent to speaker via bluetooth

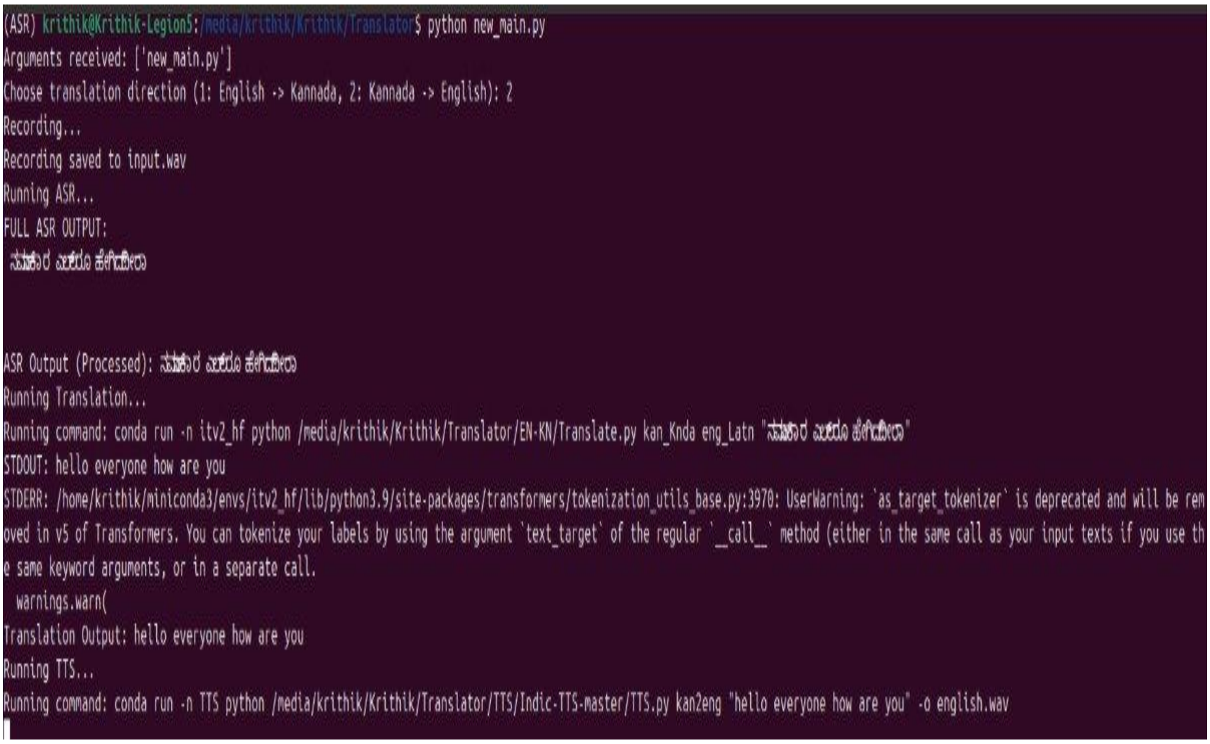
Feasibility: No network connectivity required and can be held in hand hence works in any

Novelty: The entire setup works offline and does translation locally on the SBC itself, moreover the setup is portable.

Simulation

The simulation consists of a program which cascades ASR, NMT and TTS and does the translation of languages. The environment used was python within Ubuntu 24.04

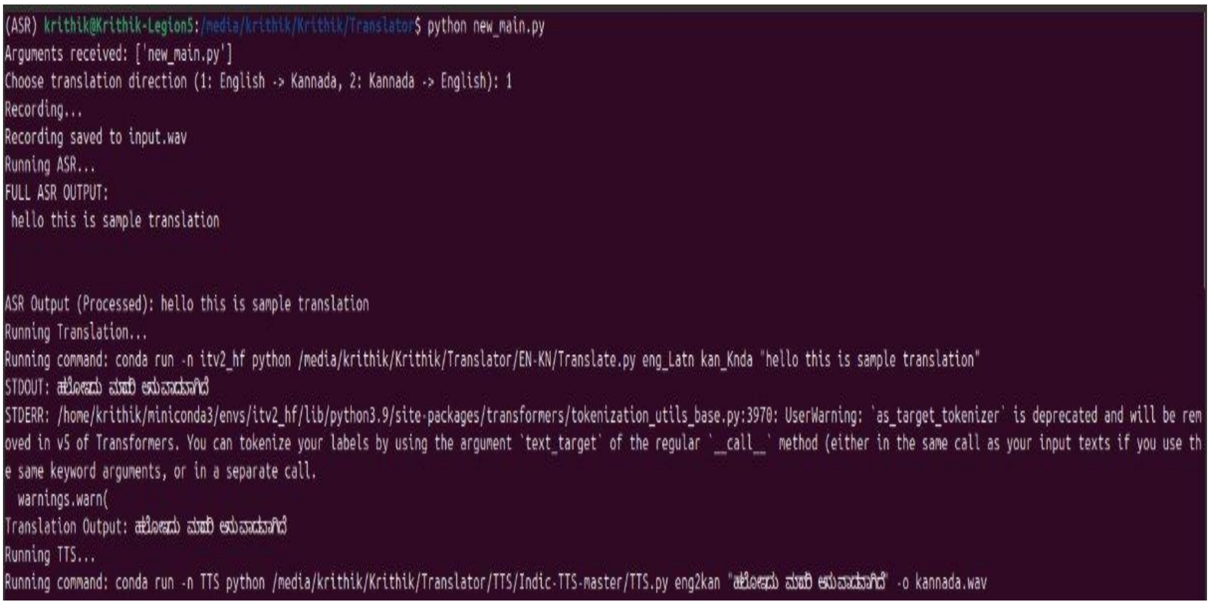
Case 1: English to Kannada



```
(ASR) krithik@Krithik-Legion5:/media/krithik/Krithik/Translator$ python new_main.py
Arguments received: ['new_main.py']
Choose translation direction (1: English -> Kannada, 2: Kannada -> English): 2
Recording...
Recording saved to input.wav
Running ASR...
FULL ASR OUTPUT:
 ನಮಸ್ಕಾರ ಎಲ್ಲರೂ ಹೇಗಿದ್ದೀರಾ


ASR Output (Processed): ನಮಸ್ಕಾರ ಎಲ್ಲರೂ ಹೇಗಿದ್ದೀರಾ
Running Translation...
Running command: conda run -n itv2_hf python /media/krithik/Krithik/Translator/EN-KN/Translate.py kan_Knda eng_Latn "ನಮಸ್ಕಾರ ಎಲ್ಲರೂ ಹೇಗಿದ್ದೀರಾ"
STDOUT: hello everyone how are you
STDERR: /home/krithik/miniconda3/envs/itv2_hf/lib/python3.9/site-packages/transformers/tokenization_utils_base.py:3970: UserWarning: `as_target_tokenizer` is deprecated and will be rem
oved in v5 of Transformers. You can tokenize your labels by using the argument `text_target` of the regular `__call__` method (either in the same call as your input texts if you use th
e same keyword arguments, or in a separate call.
  warnings.warn(
Translation Output: hello everyone how are you
Running TTS...
Running command: conda run -n TTS python /media/krithik/Krithik/Translator/TTS/Indic-TTS-master/TTS.py kan2eng "hello everyone how are you" -o english.wav
```

Case 2: Kannada to English



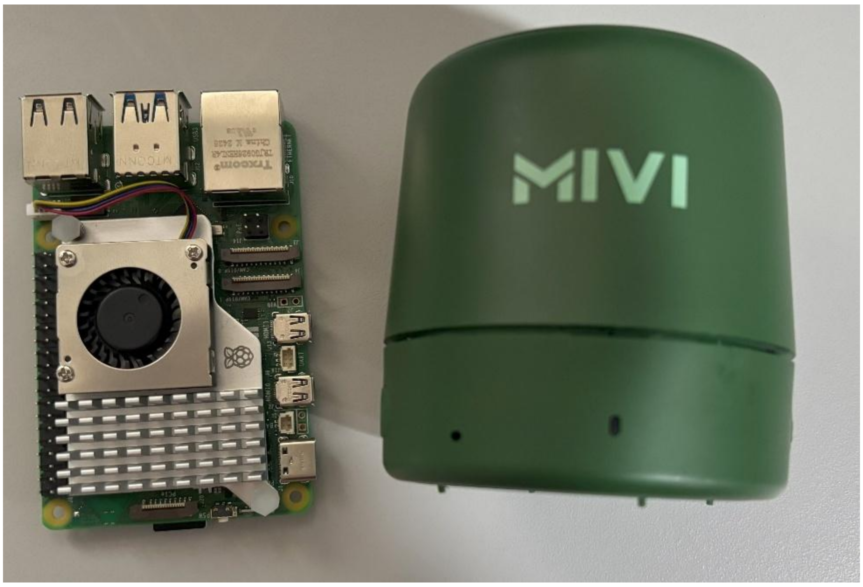
```
(ASR) krithik@Krithik-Legion5:/media/krithik/Krithik/Translator$ python new_main.py
Arguments received: ['new_main.py']
Choose translation direction (1: English -> Kannada, 2: Kannada -> English): 1
Recording...
Recording saved to input.wav
Running ASR...
FULL ASR OUTPUT:
 hello this is sample translation


ASR Output (Processed): hello this is sample translation
Running Translation...
Running command: conda run -n itv2_hf python /media/krithik/Krithik/Translator/EN-KN/Translate.py eng_Latn kan_Knda "hello this is sample translation"
STDOUT: ಹಲೋಇದು ಮಾದರಿ ಅನುವಾದವಾಗಿದೆ
STDERR: /home/krithik/miniconda3/envs/itv2_hf/lib/python3.9/site-packages/transformers/tokenization_utils_base.py:3970: UserWarning: `as_target_tokenizer` is deprecated and will be rem
oved in v5 of Transformers. You can tokenize your labels by using the argument `text_target` of the regular `__call__` method (either in the same call as your input texts if you use th
e same keyword arguments, or in a separate call.
  warnings.warn(
Translation Output: ಹಲೋಇದು ಮಾದರಿ ಅನುವಾದವಾಗಿದೆ
Running TTS...
Running command: conda run -n TTS python /media/krithik/Krithik/Translator/TTS/Indic-TTS-master/TTS.py eng2kan "ಹಲೋಇದು ಮಾದರಿ ಅನುವಾದವಾಗಿದೆ" -o kannada.wav
```

Hardware Implementation

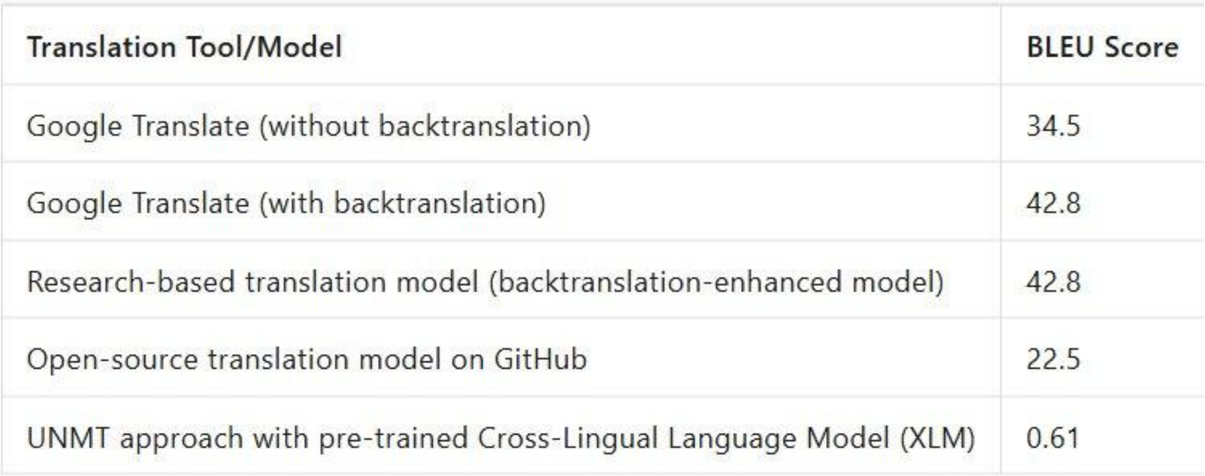
- Raspberry Pi 5 8GB and its cooling fan
- HDMI to micro–HDMI Cable
- Bluetooth Speaker
- SD Card
- Raspberry Pi power supply

Overall Cost: 11.4K INR



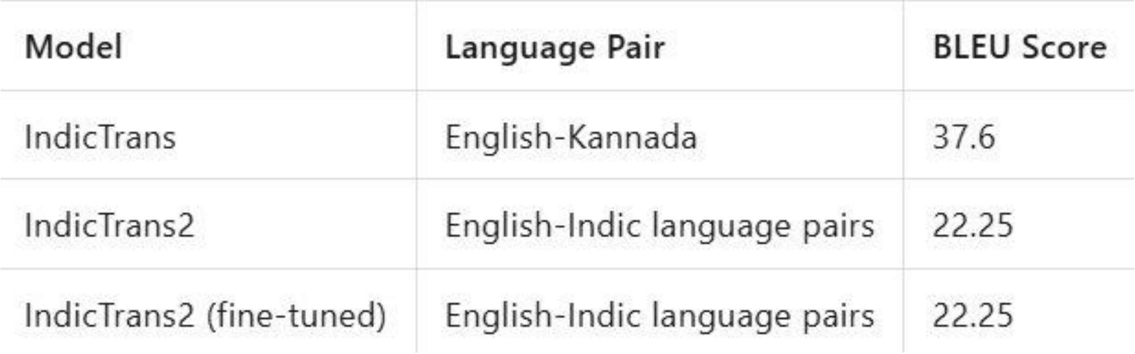
Results

Comparison of other translator models using BLEU score:



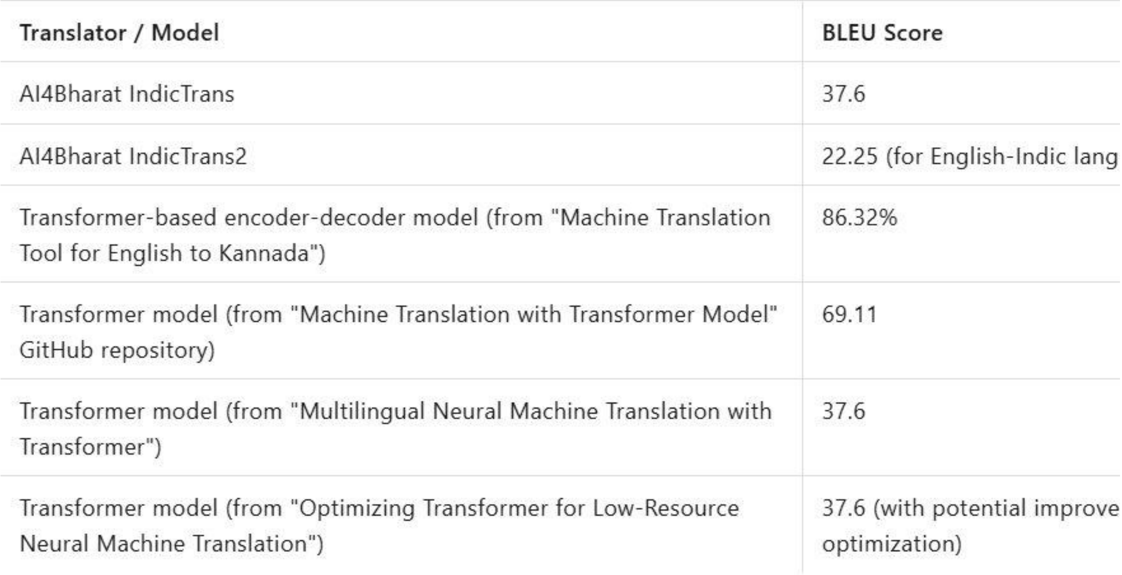
Translation Tool/Model	BLEU Score
Google Translate (without backtranslation)	34.5
Google Translate (with backtranslation)	42.8
Research-based translation model (backtranslation-enhanced model)	42.8
Open-source translation model on GitHub	22.5
UNMT approach with pre-trained Cross-Lingual Language Model (XLM)	0.61

BLEU scores of translation models built by Ai4Bharat



Model	Language Pair	BLEU Score
IndicTrans	English-Kannada	37.6
IndicTrans2	English-Indic language pairs	22.25
IndicTrans2 (fine-tuned)	English-Indic language pairs	22.25

Comparisons of BLEU Scores of other types of translation models with that of our model



Translator / Model	BLEU Score
AI4Bharat IndicTrans	37.6
AI4Bharat IndicTrans2	22.25 (for English-Indic lang
Transformer-based encoder-decoder model (from "Machine Translation Tool for English to Kannada")	86.32%
Transformer model (from "Machine Translation with Transformer Model" GitHub repository)	69.11
Transformer model (from "Multilingual Neural Machine Translation with Transformer")	37.6
Transformer model (from "Optimizing Transformer for Low-Resource Neural Machine Translation")	37.6 (with potential improve optimization)

Analysis:

- **Top Performers**:
  - *Google Translate (with backtranslation)* and a research model both achieve **42.8**, the highest credible scores, highlighting backtranslation's effectiveness.
  - *IndicTrans* (**37.6**) matches optimized Transformers, outperforming Google's base model (**34.5**).

- **Outliers & Issues**:
    - A *Transformer model* claims **86.32%**, likely due to overfitting, narrow test data, or metric misuse.
    - *IndicTrans2* (**22.25**) underperforms, suggesting multi-language scalability challenges; fine-tuning adds no improvement.
- **Low-Resource & Unsupervised**:
    - Optimized Transformers (e.g., *low-resource NMT*) hit **37.6**, showing viability for constrained settings.
    - *UNMT XLM* scores **0.61**, underscoring the difficulty of unsupervised methods without parallel data.

Key Takeaway: Backtranslation and domain-specific optimizations yield the best results, while multi-language models and unsupervised approaches struggle.

Conclusion

- Successfully integrated and tested the models
- IndicTrans is overall a good fit for using on a raspberry pi
- Hardware prototype works properly
- **Challenges & Roadblocks:**
    - The delay is high.
    - Better models do not work on the raspberry pi due to dependencies not developed for ARM architecture
- **Next Steps & Roadblocks:**
    - Delay must be improved
    - Need to add support for more languages and add seamless selection of direction