Team18 - Voice based metro ticket booking system

Aishwarya Bahirat, Kapil Soni, Mahendra Mahajan, Sourav Kumar Tripathy, T Senthilkumar, Tanzimur Rahman

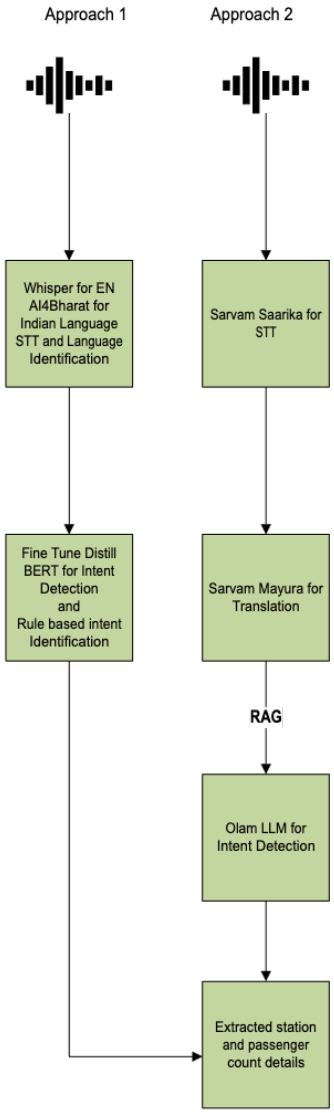
Abstract. Current metro ticket booking processes over platforms like WhatsApp require multiple steps and interactions with a chatbot, such as selecting the language, specifying the source station, selecting the destination station, and providing other necessary details. This multi-step interaction can be time-consuming, repetitive, and inconvenient for users, especially in fast-paced urban environments. To streamline this process, we aim to enable users to simply send a single voice message to the bot containing all the necessary booking details (e.g., language, source station, destination station, travel time, and ticket quantity). The bot will automatically detect the spoken language, transcribe the speech, extract the user's intent, and proceed directly to the booking step, significantly reducing the effort and time required for ticket booking. This approach not only enhances user experience but also makes the system more accessible for individuals who prefer voice-based interactions or are not comfortable with manual text-based inputs

1 Introduction

The use of artificial intelligence in public transport is making travel easier and more user-friendly. Metro systems, which are an important part of any city life, need booking systems that are simple to use and work well for people who speak different languages and may not be good in gadget usages. However, traditional systems often struggle with voice commands and don't handle multiple languages very well.

To solve this, we've built a smart system that brings together speech recognition, language translation, and advanced AI tools to better understand what users want. Our solution is designed for India's diverse language needs, supporting multiple regional languages, and ensures that booking metro tickets through voice commands is both accurate and easy to use.

2 Architecture



3 Implementation Details

3.1 STT Using Whisper/AI4Bharat and Finetuned BERT/Rules for intent detection

3.1.1 Data Collection & Preparation

- Created a dataset of metro booking voice commands in multiple Indian languages
- Annotated data with BIO tagging scheme for named entity recognition (B-FROM, I-FROM, B-TO, I-TO, B-NUM)
- Developed techniques to handle multi-word station names like "KR Market" and "Cubbon Park"
- Properly aligned tokens and labels in the training data for accurate entity extraction

3.1.2 Model Development

Implemented a modular pipeline with three core components:

- **Language Identification**: Detects the spoken language of the voice message
- **Speech Recognition**: Transcribes audio to text using language-specific models. Whisper is used for detecting and transcribe English languages.
- **Intent Detection**: Extracts key booking entities using fine-tuned NER models
    - Fine-tuned BERT-based model DistilBert for entity extraction (source station, destination station, ticket count)
    - Rule based classification is used for AI4Bharat approach

3.1.3 Training & Optimization

- Fine-tuned NER models using custom BIO tagging scheme
- Addressed challenges in tokenization and subword handling for station names
- Implemented proper entity boundary detection for multi-token entities
- Adapted models for both word-level and subword-level tagging
- Pattern matching approaches are also explored.

3.1.4 Evaluation

Developed comprehensive evaluation pipeline with multiple metrics:

- **Entity Recognition Accuracy**: For source station, destination station, and ticket count
- **Overall Accuracy**: Percentage of bookings with all entities correctly identified
- **Confusion Matrices**: To identify commonly confused stations and systematic errors

3.1.5 Deployment & Integration

- Created architecture supporting multiple Indian languages
- Language is auto detected, station information is extracted, and relevant intent is identified.

3.2 Multilingual ticketing with Sarvam AI and RAG

This approach leverages Sarvam AI's multilingual models and a Retrieval-Augmented Generation (RAG) framework to enable voice-based metro ticket booking in multiple Indic languages. This approach, implemented as a Streamlit web application, takes user voice inputs through a audio process module. The audio processing module records 16kHz audio using sounddevice python module and saves it as WAV files via wavio python module. Sarvam's Saarika API transcribes the audio into text. Sarvam's Mayura translates the transcription to English and enabling intent processing.

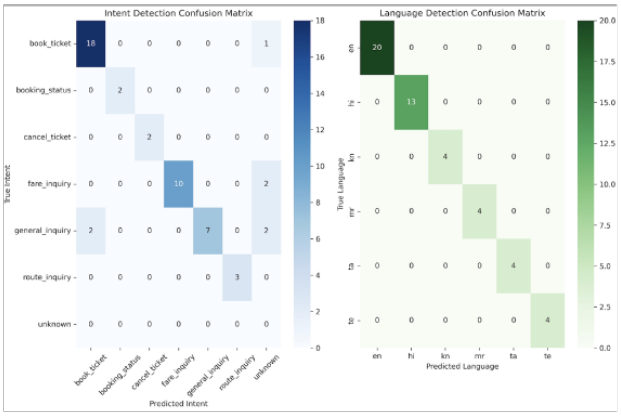
The RAG system which uses the sentence-transformers (all-MiniLM-L6-v2), retrieves relevant intent examples from a knowledge base using cosine similarity. These contexts, along with user input feed to a local Ollama LLM (llama3), achieve 90% intent recognition accuracy for actions like metro booking. Sarvam-M further refines intent classification. The Streamlit interface displays transcriptions, translations, and intents, making user-friendly interaction. This approach minimizes user effort by processing a single voice command, enhancing accessibility for non-English speakers in India's multilingual environments.

The evaluation of this approach system assesses the intent recognition performance. A test dataset of 26 English-language user queries, covering different intents was used to evaluate the system's classification accuracy. Processes queries through the RAG system with sentence-transformers (all-MiniLM-L6-v2) to retrieve intent contexts, followed by LLM-based classification. Performance metrics include precision, recall, and F1-score, computed using scikit-learn. The system achieved a weighted F1-score of 0.9612, indicating robust performance across intents. The overall accuracy was 0.96, These results demonstrate the system's effectiveness in handling multilingual intent recognition.

4 Challenges

Getting test data set for multiple Indian languages is challenging. Designing a system to handle a mix of multiple languages in the same voice message is also challenging. NER detection with fine tuning for long station names like Sri Satya Sai Hospital was not always accurate.

5 Evaluation Metric



Intent Type	Accuracy	Correct/Total	Performance Grade
book_ticket	93.3%	14/15	A
fare_inquiry	93.8%	15/16	A
cancel_ticket	80.0%	8/10	B
booking_status	75.0%	6/8	B-
route_inquiry	70.0%	7/10	C+
general_inquiry	71.4%	10/14	C+

Metric	Score	Percentage	Grade
Overall Accuracy	0.822	82.2%	B+
Macro F1 Score	0.856	85.6%	A-
Micro F1 Score	0.863	86.3%	A-
Weighted F1 Score	0.863	86.3%	A-

6 Future Scope

- Evaluate small models that can run on edge devices like mobiles
- Fine tune Indic BERT model for intent detection
- TTS
- Deployment as a WhatsApp Bot
- Add support to get feedback from user
- Support to get the monthly pass

7 References

- [1] Sarvam AI, "Saarika, Mayura, Bulbul, and Sarvam-M: Multilingual AI Models for Indian Languages," https://sarvam.ai, accessed October 2024.
- [2] Streamlit, "Streamlit: A Framework for Building Data Apps," https://streamlit.io, accessed October 2024.
- [3] Ollama, "Ollama: Run Large Language Models Locally," https://ollama.ai, accessed October 2024.
- Finetuning BERT for Custom NER https://medium.com/@maneyogesh065/fine-tuning-biobert-for-custom-named-entity-recognition-a-complete-guide-a05b124edda0

8 Github Link

https://github.com/senthiltk/DeepLearningProject

9 Annexure

9.1 Contributions

[1] AI4Bharat and Indian Language SST and Intent Detection – Aishwarya Bahirat and Sourav Kumar Tripathy
[2] Whisper and Fine tuning DistilBert for intent detection – T Senthilkumar and Tanzimur Rahman
[3] Sarvam evaluation for STT and intent detection – Kapil Soni and Mahendra Mahajan

10 Demos

https://github.com/senthiltk/DeepLearningProject/blob/main/book_ticket.mov
https://github.com/senthiltk/DeepLearningProject/blob/main/cancel_ticket.mov
https://github.com/senthiltk/DeepLearningProject/blob/main/fare_inquiry.mov
https://github.com/senthiltk/DeepLearningProject/blob/main/route_inquiry.mov