SpeakRite

English Improvement Coach for Indian Accented Tones

Enhancing Communication Skills for Indian Speakers

CORPORATE GURUKUL

Index

Project Outline

Problem Statement

- The problem statement revolves around creating an English Improvement Coach tailored for Indian accented tones. The goal is to leverage speech to text and text to speech models, along with a chat interface, to facilitate English language coaching specifically designed for individuals with diverse and native accents.

Aim

- The aim is to develop an English Improvement Coach tailored for Indian accented tones by effectively utilizing speech to text and text to speech models along with a chat based interface.

CORPORATE GURUKUL

The Team

CORPORATE GURUKUL

SpeakRite

Group 5

The Team - AIChemists

Anika Kamath

Data Preprocessing, Analysis, Model Development and Documentation

Santhosh Kumar Santhanam

Data Preprocessing, Analysis, Model Development, Comparing Optimal Model and Documentation

Kapilesh Neelakandan

Openai-GPT API Integration, gtts Integration, Capturing Speech Input and Front-End Development using gradio

Aryan Gupta

Openai-GPT API Integration, gtts Integration, Capturing Speech Input and Front-End Development using gradio

Bhavishya Sharma

Systems Modelling, Dataset Research, Grammar correction Models Implementation, Documentation

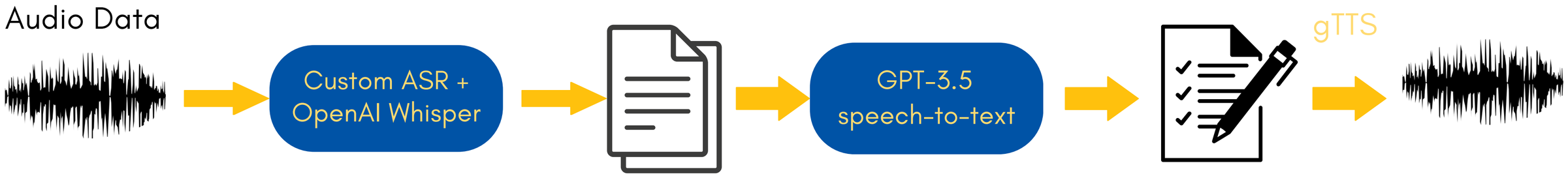
Adithya Murali

Dataset Research, Analysis, Comparing Optimal Model and Documentation

CORPORATE GURUKUL

SpeakRite

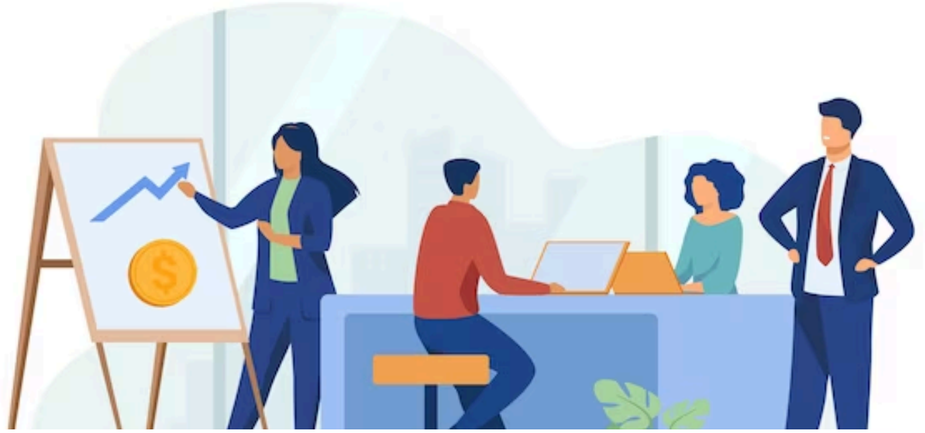
What are we trying to build?



Abstract

Acquiring Accented Indian English Audio Dataset	Svarah: Comprises diverse Indian accents in .wav file format, as a foundational resource for training and validating the speech to text and text to speech models.
Customizing and experimenting with various speech to text models such as OpenAI Whisper, Wav2Vec2 and HuBERT	For transcription and analysis of Indian accented speech, addressing the nuances of different regional accents.
Building to Speech to Speech chat interface using Gradio	Provides feedback, and language improvement, addressing the specific needs and nuances of Indian speakers.

Business Use Cases



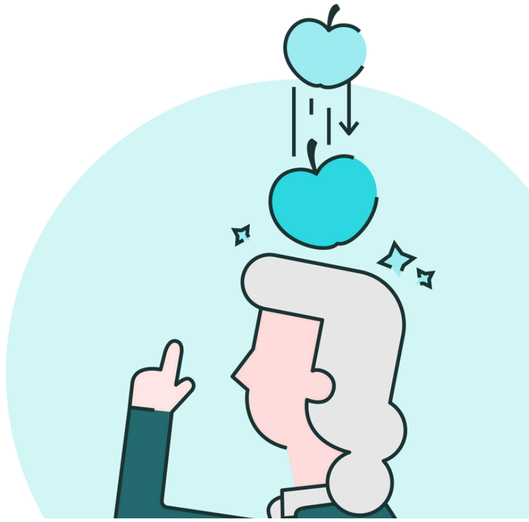
1 Language Training Institutes: Implementing the English Improvement Coach in language training institutes to offer specialized English language coaching for individuals with Indian accents

2 Corporate Training Programs: Integrating the system into corporate training programs, especially in organizations with a significant number of employees from India or individuals with Indian accents.

3 English Language Coaching Services: Incorporating the system into existing English language coaching services, particularly those providing personalized coaching to individuals with specific language requirements, including Indian accent considerations

4 In Customer Support Services: International companies in the technology, finance, and e-commerce sectors, outsource their customer service operations to call centers in India, tools like these can help in briding the communication gap caused due to accented intonation.

Important Use Case



Enhancing Public Speaking Skills with
Integrated 3D Models and Text-to-Speech Technology

Innovative Integration: Combining Text-to-Speech technology with 3D models creates a platform to refine public speaking skills through text correction and facial expression simulation.

Real-time Guidance: Analysis of final audio enables 3D models to provide visual feedback, aiding users in refining speech delivery instantly.

User-Centric Environment: This integrated solution offers pronunciation assistance and visual guidance, fostering confidence in public speaking scenarios. This system elevates traditional training, enhancing user performance in public speaking with comprehensive feedback.

Concepts Used

- Data Analysis and Visualization
- Training a Custom ASR using Transformers and OpenAI Whisper
- Integration of OpenAI APIs for text improvement
- Gradio - For building User Interface
- Fine-tuning of Regional Data (Filtering Regions as per Accent)
- Pandas Profiling - EDA
- Incorporating Google Text to Speech Engine for Audio Outputs in different accents.
- GitHub - Version Control and Development

Workflow at a Glance

Data Acquisition and Preprocessing

Carrying out research and acquiring datasets consisting of a wide range of Indian Accents

Performing EDA and carrying out filtering of demographic data as per accent type.

Using selected regional data (TN) to train a custom ASR

Model Training and Experimentation

Fine-tuned the audio parameters to build a Custom ASR on various models such as Wav2Vec, HuBERT and OpenAI Whisper.

As per the model and integration constraints, we chose to proceed with the OpenAI Whisper (base) model.

Converting the speech to text for improvement

Converting the transcribed data to text input for the GPT-3 API to process and handle requests via a python script.

This script takes in gramatically incorrect text input and processes it to generate corrected and improved text output.

Providing a text to speech output of improved speech

Using Google's text to speech engine, the improvised text is converted to an audio as well, which is available in different global accents.

Building the Front-end using Gradio

Using Python's in-built module, Gradio, to build a component - based user interface to host locally and deploy on a public domain using version control tools like GitHub.

Data Collection Process

Common Voices by Mozilla - WorldWide Accent Dataset

Svarah - An Indic accented English speech dataset

AccentDB by Tensorflow

Svarah - An Indic accented English speech dataset

Dataset which suits our purpose the most

Exploratory Data Analysis

Exploratory Data Analysis - Svarah

Dataset statistics

Number of variables	11
Number of observations	6656
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	4.8 MiB
Average record size in memory	754.1 B

Variable types

Text	3
Numeric	1
Categorical	7

Exploratory Data Analysis - Svarah

text

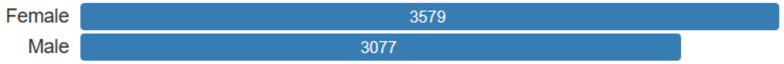
Text

Distinct	5585
Distinct (%)	83.9%
Missing	0
Missing (%)	0.0%
Memory size	755.5 KiB

gender

Categorical

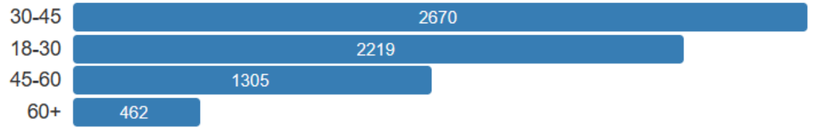
Distinct	2
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	403.6 KiB



age-group

Categorical

Distinct	4
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	402.2 KiB



Exploratory Data Analysis - Svarah

native_place_district

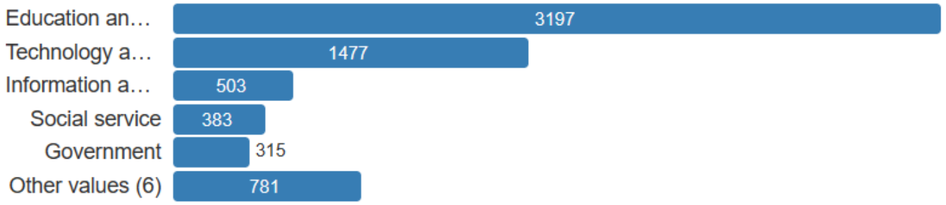
Text

Distinct	65
Distinct (%)	1.0%
Missing	0
Missing (%)	0.0%
Memory size	429.0 KiB

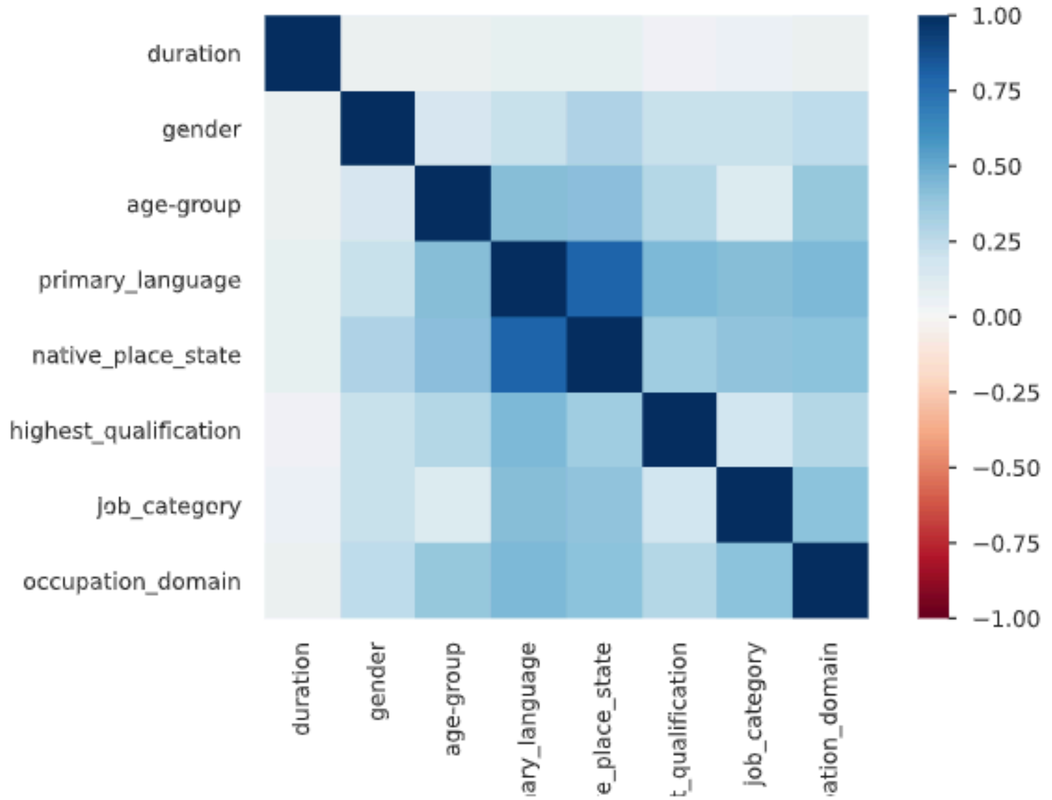
occupation_domain

Categorical

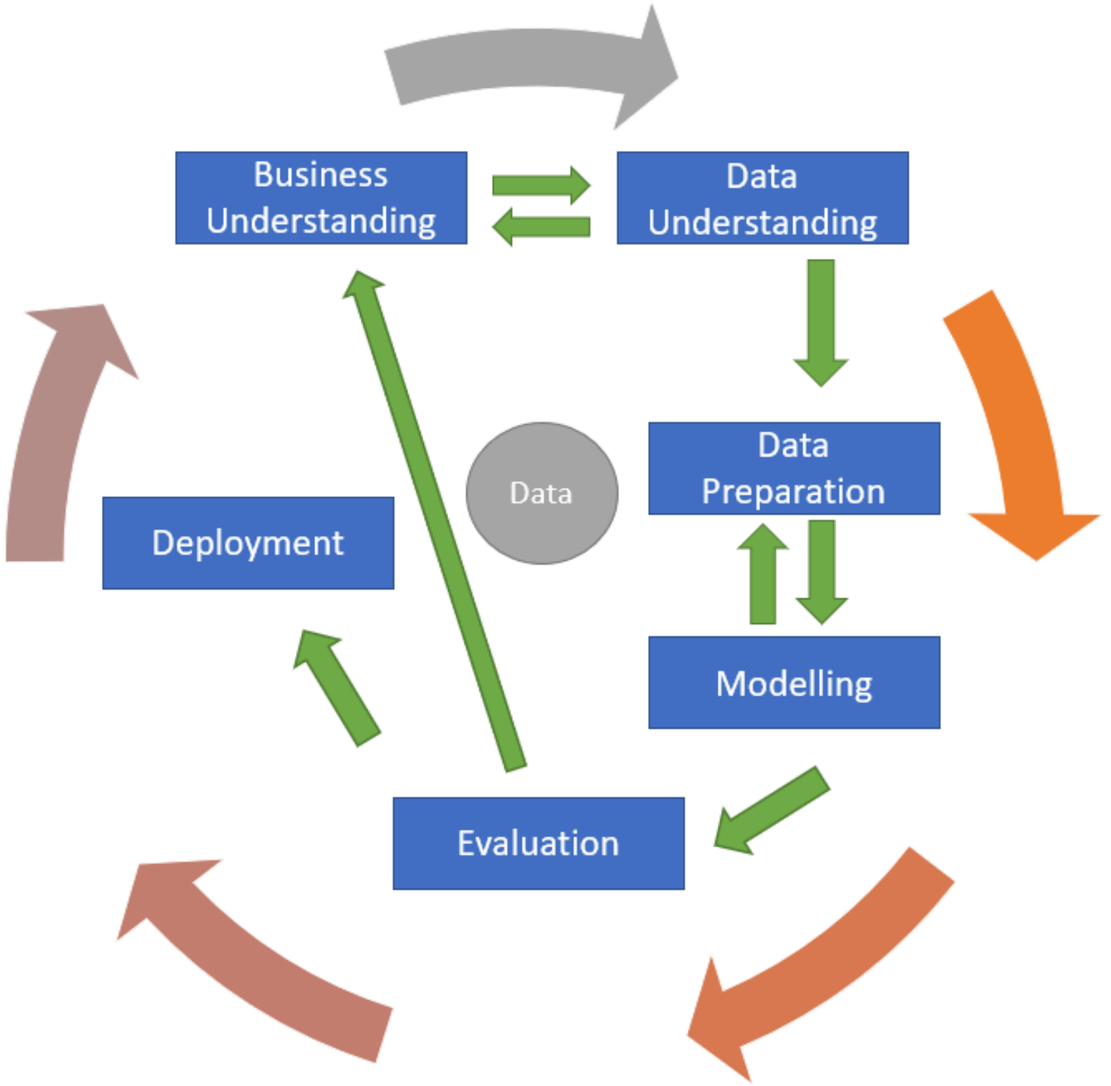
Distinct	11
Distinct (%)	0.2%
Missing	0
Missing (%)	0.0%
Memory size	512.6 KiB



Exploratory Data Analysis - Svarah



Implementation of CRISP - DM Methodology

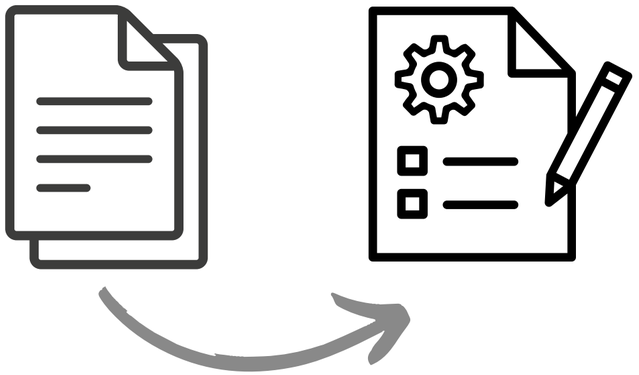


Algorithms (Models) Used

OpenAI Whisper

Input audio to Text

GPT3 Transformer

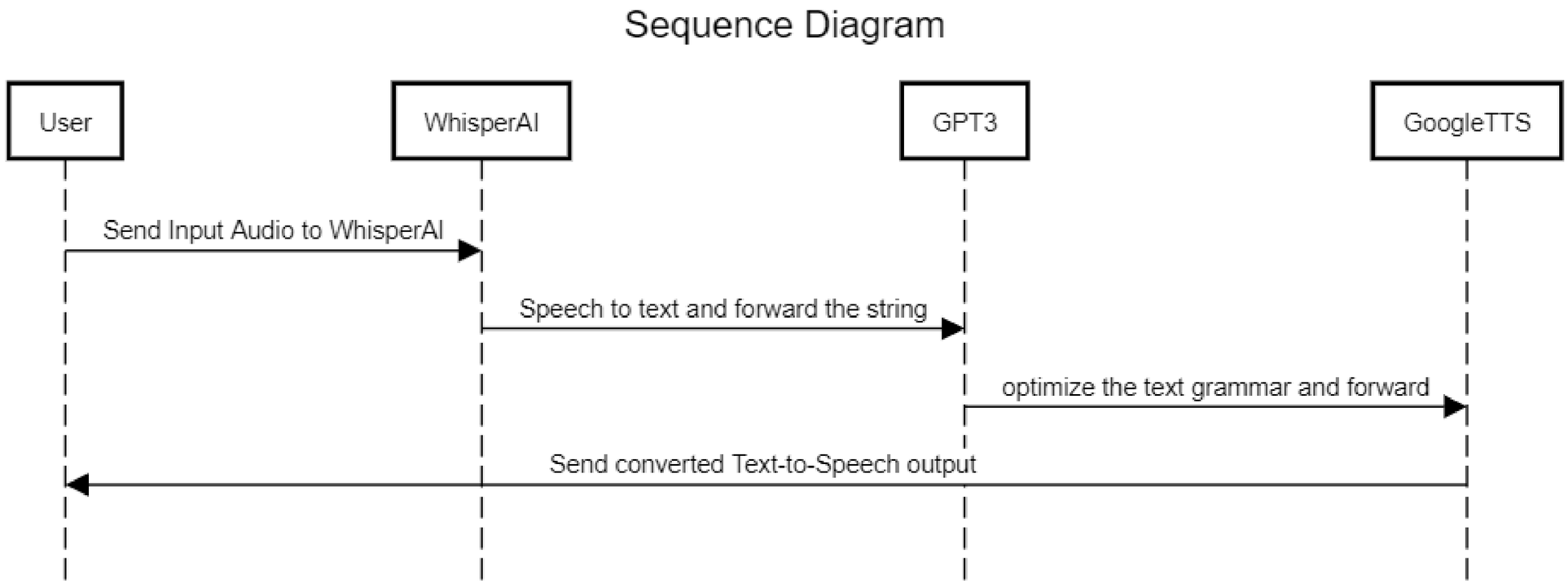


Text Optimization / Grammar correction

Google Text-to-Speech

Optimized text to Speech conversion

Sequence Diagram



Problems Faced in Training Previous Models

Wav2Vec

Computational resources: Training Wav2Vec models can be computationally expensive, requiring powerful hardware and significant training time.

HuBERT

Clustering Quality: HuBERT relies on k-means clustering to generate pseudo-labels for pre-training. Poor clustering quality can lead to inaccurate labels and hinder model performance, especially when dealing with accented datasets.

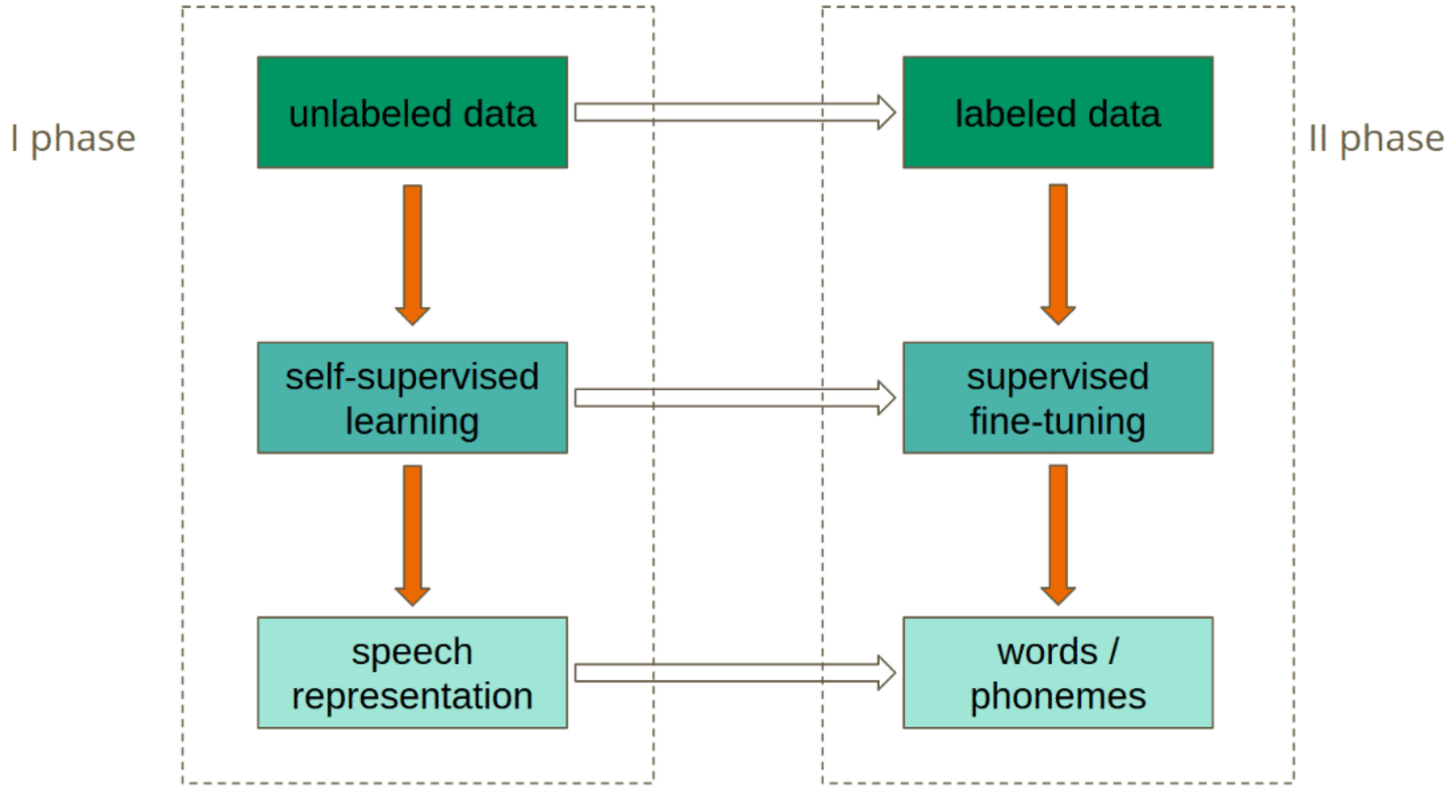
speech to text

LLaMa

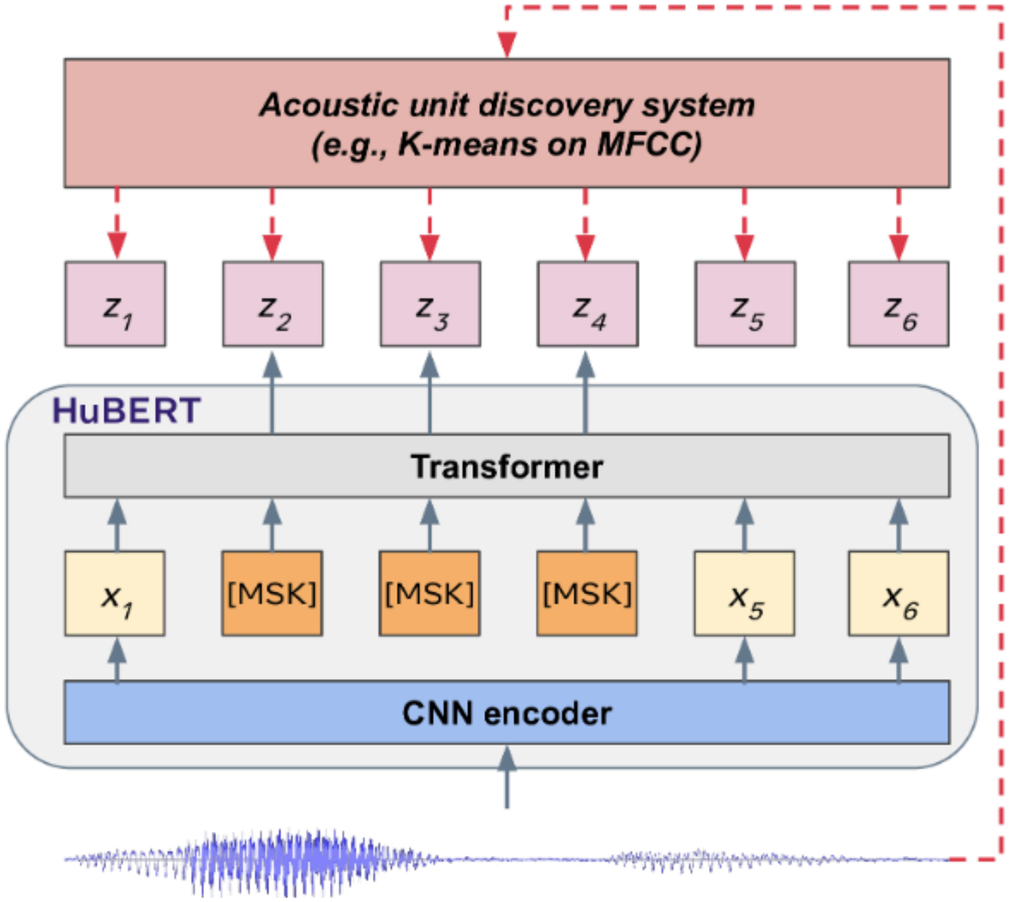
Formatting and Preprocessing: LLaMA has specific input format requirements. Properly format and preprocess your custom data to ensure compatibility with the model's architecture, which was challenging to integrate with the speech to text model

text to speech

Wav2Vec Model



Hubert Model



OpenAI Whisper

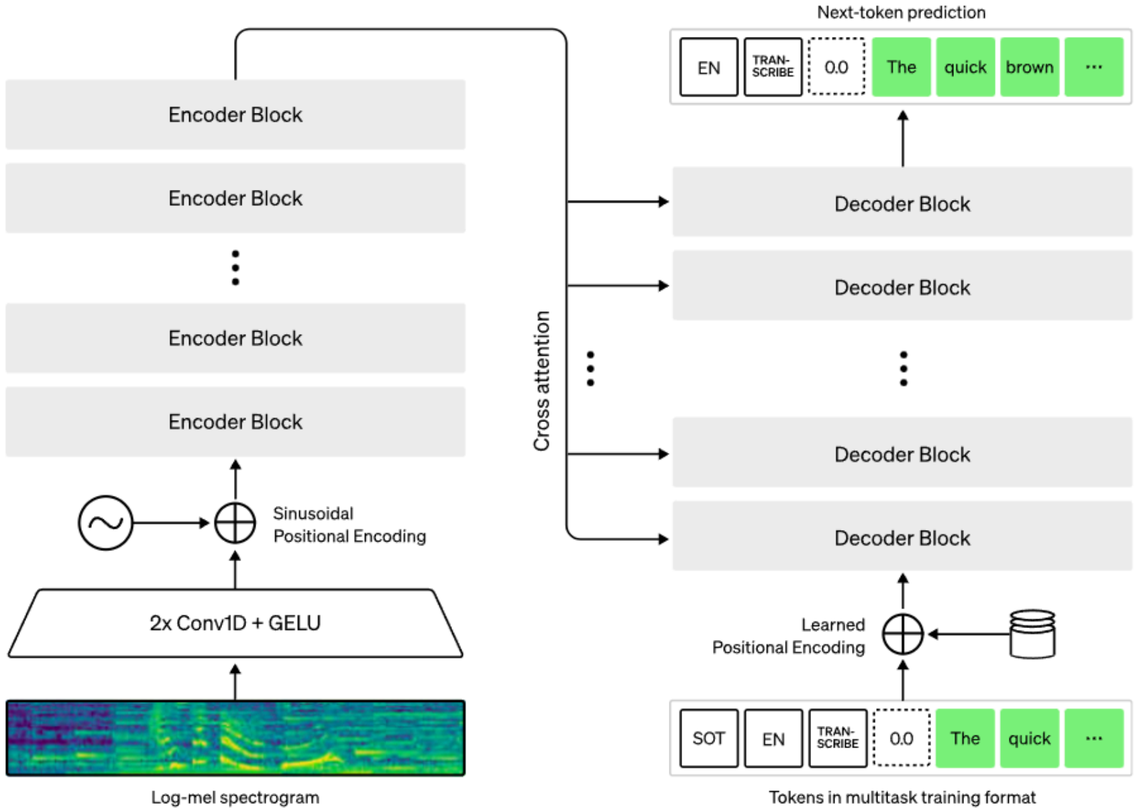
Whisper is a **Transformer** based encoder-decoder model, also referred to as a **sequence-to-sequence model.**

We have trained WhisperProcessor with our dataset to :

- Pre-process the audio inputs (converting them to log-Mel spectrograms for the model)
- Post-process the model outputs (converting them from tokens to text)

Models Used

OpenAI Whisper Model Architecture



CORPORATE GURUKUL

SpeakRite

Models Used

GPT - 3

GPT-3, or Generative **Pre-trained Transformer 3**, is a state-of-the-art language processing artificial intelligence model developed by OpenAI.

GPT 3 HAS BEEN TRAINED FOR-

Natural Language Understanding:

- Example: GPT-3 is trained to understand and generate human-like text, enabling it to perform tasks such as language translation, summarization, and question-answering.

Model Comparison

Model	Architecture	Pre-trained	Fine-tuning	Task	Languages	Performance	Availability
wav2vec	Transformer encoder-decoder	Yes (Wav2Vec 2.0 Large)	Required	Speech-to-text	Multilingual	High accuracy, good for noisy audio	Hugging Face Transformers library
whisper	Conformer-based encoder-decoder	Yes (Whisper Large)	Optional	Speech-to-text, speech translation	Multilingual	High accuracy, fast inference, good for noisy audio	Hugging Face Transformers library
Hubert	Transformer encoder	Yes (HuBERT Large)	Required	Speech recognition, speaker diarization	Multilingual	Good accuracy, efficient for large datasets	Fairseq toolkit

Model Comparison (Metric : Accuracy)

Wav2Vec	Whisper	HuBERT
62 %	83.34 %	67.62 %

```
flattened_list_1 = [sentence[0].lower() for sentence in model_predictions]
flattened_list_2 = [sentence.lower() for sentence in ground_truth_transcriptions]

all_sentences = flattened_list_1 + flattened_list_2
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(all_sentences)
similarity_matrix = cosine_similarity(tfidf_matrix[:len(flattened_list_1)],
tfidf_matrix[len(flattened_list_1):])

similarities = np.diag(similarity_matrix)

# Display the cosine similarities
tt = 0
for idx, sim_score in enumerate(similarities):
    print(sim_score)
    tt+=sim_score

total_samples = len(ground_truth_transcriptions)
accuracy = tt/ total_samples * 100
```

```
print("Accuracy of Whisper Model:",round(accuracy,2))
```

```
Accuracy of Whisper Model: 83.34
```

CORPORATE GURUKUL

SpeakRite

Code Snippet: Pre-Processing

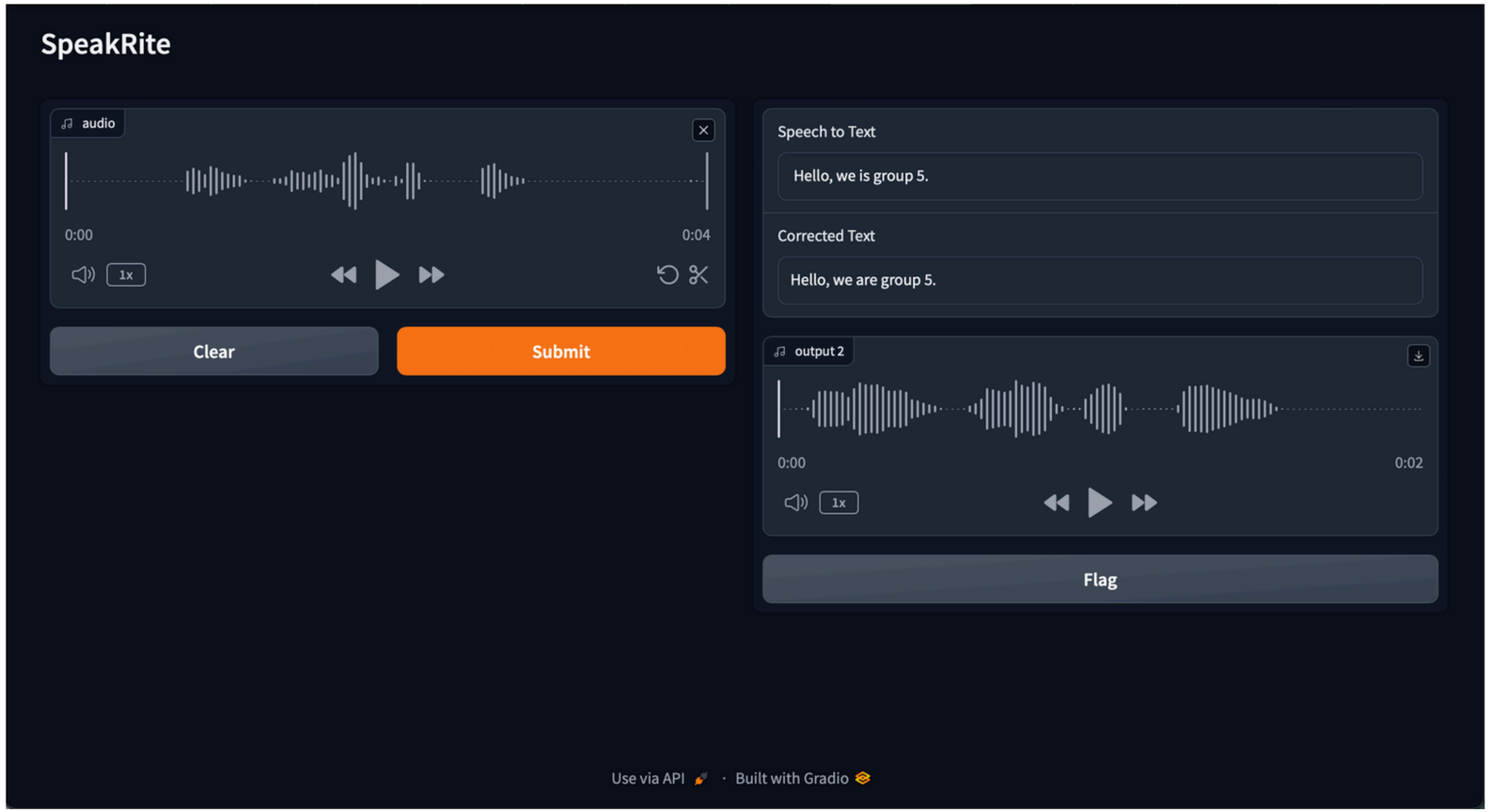
```
json_data = []
with open('svarah_manifest.json', 'r') as json_file:
    for i, line in enumerate(json_file):
        json_data.append(json.loads(line))

df = pd.DataFrame(json_data)
df = df[df['primary_language'] == 'tamil']
df1 = pd.read_csv('meta_speaker_stats.csv')
combined_df = pd.merge(df[['audio_filepath', 'text']],
 df1[['audio_filepath']], on="audio_filepath")

combined_df = combined_df.rename(columns={'audio_filepath': 'file_name', 'text': 'transcription'})
combined_df['file_name'] = combined_df['file_name'].str.replace('audio/', '')
print(df.columns)
combined_df.to_csv('/content/drive/My Drive/Indian_accent/all_languages.csv', index=False)
```

Code Snippet : Model Integration

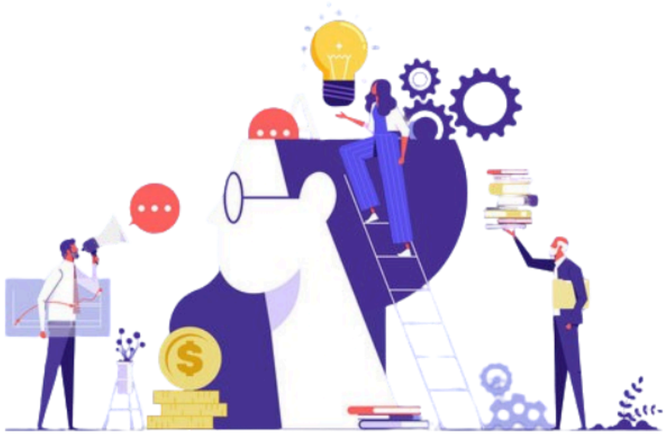
Integrating the Custom ASR and the GPT-3 using Gradio Interface

```
p = pipeline("automatic-speech-recognition", model="openai/whisper-base.en")

transcriber = pipeline("automatic-speech-recognition", model="openai/whisper-base.en")

def transcribe(audio):
    sr, y = audio
    y = y.astype(np.float32)
    y /= np.max(np.abs(y))
    input_text = transcriber({"sampling_rate": sr, "raw": y})["text"]
    messages = [
    {"role": "system", "content": "You are a helpful assistant."}]

    if input_text:
        messages.append(
            {"role": "user", "content": '''correct the english
            in the following sentences.'''+input_text},
        )
        chat_completion = openai.ChatCompletion.create(
            model="gpt-3.5-turbo", messages=messages
        )
    reply = chat_completion.choices[0].message.content
    tts = gTTS(text=reply, lang='en')
    tts.save("output.wav")

    return(input_text,reply, 'output.wav')
```

User Interface



Our Learnings



Structure of a Project

We learnt about the proper and formal way of presenting a Project and how the breakdown of tasks is to be carried out.

Neural Networks

Learning about diverse, pre-existing Speech and Text synthesis models , comparing them and integrating them with our project.

Problem Research

Dive into modules designed to address specific pronunciation challenges associated with Indian accents. Practice words and phrases commonly mispronounced, and receive personalized feedback.

Identifying Problem Statement

Identify and overcome common grammatical challenges faced by English learners with Indian accents. Engage in interactive lessons that focus on improving grammar skills in real-world scenarios.

Challenges Faced

Diverse Accent Variations:

Indian English encompasses a wide range of accents and dialects, which can pose a challenge when developing speech-to-text and text-to-speech models.

Model Training and Adaptation:

Training speech-to-text and text-to-speech models to effectively recognize and generate Indian accented English requires specialized adaptation and fine-tuning

Data Quality and Availability:

Ensuring the availability of high-quality, diverse Indian accented speech data for training the models can be a significant challenge.

Overcoming Noise and Disturbances:

The GitHub dataset may have variability in quality and background noise, which can impact model training.

Future Scope

Augmenting Public Speaking with Real-time Feedback and 3D-TTS Synergy:

This innovative synergy blends real-time speech analysis for Indian accented English with integrated 3D models and Text-to-Speech technology. It offers instant pronunciation corrections, tailored language tips, and facial expression simulation. Empowering users in refining their speech, it's a transformative approach revolutionizing public speaking training.

Expansion to Other Regional Accents:

Extending the English Improvement Coach's adaptability to encompass a broader range of regional accents within India, addressing the diverse linguistic landscape and catering to the varied needs of English language learners across the country.

Gamification and Engagement:

Incorporating gamified elements, progress tracking, and rewards systems to maintain user motivation and engagement in the language learning process, tapping into cultural preferences and learning styles prevalent in India.

Personalized Learning Paths:

Developing adaptive learning algorithms that tailor course content and exercises to the individual user's proficiency level, learning pace, and specific areas of improvement, taking into account the nuances of Indian English accents.

Key Take-aways

1. The Importance of Tailored and Inclusive Solutions in Language Technology Advancements
2. The Value of Culturally Relevant Approaches in Enhancing Language Learning
3. The Importance of Adapting Technology to Meet Diverse Linguistic Needs
4. Challenges in Creating Specialized Models for Regional Accents

Team AIChemists

SpeakRite