**BACHELOR OF TECHNOLOGY**

**IN**

**“COMPUTER SCIENCE OF ENGINEERING”**

**SUBMITTED BY :**

**Namratha Akshaya BL.EN.U4CSE22134**

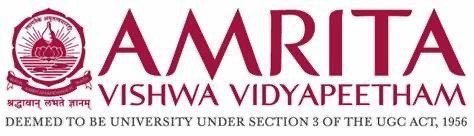
**Namrutha M BL.EN.U4CSE22137**

**Niharika Reddy Bl.EN.U4CSE22175**

**B U Naveen Raj BL.EN.U4CSE22209**

**19CSE303: MACHINE LEARNING**

**HACKATHON**

****

**NOVEMBER-2024**

**AMRITA SCHOOL OF ENGINEERING, BANGALORE**

**AMRITA VISHWA VIDYAPEETHAM**

**BENGALURU-560035**

**AUGUST-2024**

**PROBLEM STATEMENT:**

Develop an AI-based system to analyze audio input and classify it as noise or topic-relevant speech to effectively monitor and mitigate class noise pollution

**Solution:**

Step 1: It includes two functionalities; Audio Input Capture and Preprocessing

1. Microphone Setup and Preprocessing:

Setup a microphone doubles with signal preprocessing in order to receive clean sounds.

Filter and normalize speech signals by solving noise reduction problems with different approaches of ML like the Deep Speech Denoising or Wave-U-Net.

2. Segmentation for Real-Time Processing:

In order to prevent latencies, divide audio input into segments of equally sized small chunks of time, like 5 seconds. Step 2: Speech recognition is the process of converting audio to text although some authors prefer to refer to this as the transcription of audio to text.

1. Advanced Speech-to-Text Conversion with AI Models:

Select an extra-tertiary API for speech recognition which encompasses a variety of accents, noise levels, and speech patterns including Google Cloud Speech-to-Text, DeepSpeech or Whisper.

For real-time systems include a real-time ASR model which meets low-latency optimized for fast and high accuracy transcription of the audio into text.

2. Confidence Score Filtering:

Using a threshold level of confidence on ASR, exclude low confidence text transcription results out of all recognition outcomes that can henceforth be employed for subsequent processes.

Step 3: Text Translation with the help of MS learning models

1. Language Identification Using NLP Models:

To perform the language identification of the transcribed text in a swift and efficient manner, it is advised to employ a pre-trained language identification model such as, for instance, fastTex or langid.

Next if the detected language is English go to the text classification. If not, proceed to translation stages In either case, mostly all sub procedures are finally connected to this procedure.

2.Translation for Regional Languages:

Employ automatic translation approaches such as MT (Google, translate API for instance or mBART for multilingual translation) to translate non-English speech to English.

It also merits that the input domains consist of texts in English since this results in easy processing by the other tools.

Step 4: Here, expounding in details text preprocessing for the ML Models used in Classification

1. Tokenization and Text Cleaning:

Tokenizing text – Delete unnecessary symbols, convert all letters to lowercase, and use NLP libraries such as spaCy or Transformers by Hugging Face to tokenize the text and remove stop words Standardization -Result in standardized terms

Use stemming to reduce words into roots so that similar words contribute to similar classifications.

2. Feature Extraction Using Advanced NLP Technique:

Convert text into numerical features for the classification model:

TF-IDF: Pays proportional action to every word in relation to the situation.

Word Embeddings: Instead, they should perform the following crucial tasks: word embeddings (for example Word2Vec, GloVe) or BERT-based embeddings (including Sentence-BERT) to capture semantic context and relationships throughout the textual data to increase predictive potential.

Step 5: We present a text classification model for three tasks: noise high-/low-light detection, and relevance classification.

1. Labeling and Dataset Preparation:

Acquire and annotate training data in the form of noise, off topic, speech and signal which is on topic, technical speech.

Examples might include: Receive Noise (0) for general or gossip conversations, and Relevant Speech (1) for technical or academic conversations conducted in classrooms, examinations, interviews, etc.

2. Model Selection and Training:

Traditional Classifiers (Naive Bayes, SVM): The Rules are very useful with simple text and are not as effective identifying all text in complex text environments or in noisy scenes.

Deep Learning Model:

LSTM or BiLSTM: Recurrent neural networks also analyze the temporal dependencies in content of the speech and allow making more accurate classification.

Transformers (BERT, RoBERTa: Utilize language models tuned from the labelled dataset for proper distinction between technical and non-technical language.

Ensemble Learning: Use SVM with BERT model for better performance and better accuracy because models when combined produce better results when used in noisy environments.

3. Real-Time Classification Pipeline:

For each audio segment, convert text into embeddings and pass them through the classification model to label it as either:

Noise (0): Extraneous noise.

Relevant Speech (1): Specialized in the context of the discussions, sound.

4. Continuous Model Improvement:

It is also important to perform active learning to refine the result of the model over a subsequent set of iterations. Receive data from users that include instructors/examiners and update the unknown entities based on wrong classification instances.

Step 6: Output and Response Based on Classification

1. Real-Time Dashboard and Notifications:

Output on a user interface with real-time audio classification of the results.

To alert users, the system may blinking the indicator or making a sound when the program identifies that noise exists continuously and requires action.

2. Automated Actions for Specific Applications:

Exam Hall: Sound off loudspeakers to alert the invigilators if some students start talking on matters other than the concerned topic or if there is noise making.

Online Interviews: If non-relevant sounds continue, inform the interviewer, and offer a further course of action to refocus the meeting.

Classroom: Equip instructors with a live “noise vs. content” graph display so that they can deal with interference and regain order.

Step 7: Model Assessment and Calibration of the ML models

1. Evaluation Metrics:

Recalculate the accuracy, precision, recall and the F1-score of the constructed model on the real-world use cases based test data set.

It is suggested to use cross validation and confusion matrices in order to detect patterns of misclassification and improve the procedure in future.

2. Feedback Loop for Model Improvement:

Promote feedback correction where the users themselves can adjust the misclassifications if they want.

Fine tune the model on such examples with an aim of improving performance in the future, this feedback being continuous.

3. Model Deployment and Monitoring:

Put the model where it can be used to process data in close to real time- cloud server, edge nodes or similar.

Similarly supervise model performance in production and for this purpose, methods like concept drift detection are introduced which can deal with change in speech and noise pattern, over a period of time.

**AI-enabled Applications**

1. Exam Hall Monitoring:

The ability to identify low level, unrelated discussions effectively with a high degree of success is ideal since invigilators will be alerted before there is noise.

2. Online Interviews:

A proposed system includes relevance classification for maintaining a relevant conversation and using background noise suppression techniques to eliminate background noise.

3. Classroom Noise Management:

Transmit data on noise levels inside a classroom so that teachers can balance conversations with interences.

Apply adaptive noise models in accordance to the type of activities in the classroom to keep the children engaged.

Develop an AI-based system to analyze audio input and classify it as noise or topic-relevant speech to effectively monitor and mitigate class noise pollution.