ODESSA: HMM BASED AUTOMATIC SPEECH RECOGNITION SYSTEM

UNIVERSITY *of* WASHINGTON

Introduction

Automatic Speech Recognition (ASR) Systems

- Essential for modern applications like virtual assistants and transcription services
- Neural Network ASR (DNNs, CNNs): High accuracy but resource-intensive
- HMM-based ASR (like ODESSA): Efficient, low power, ideal for resource-limited environments

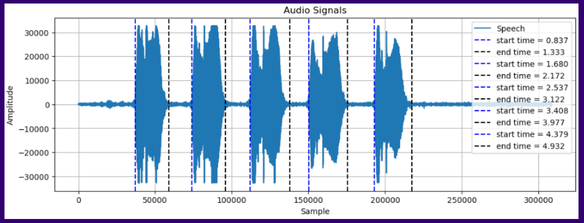
ODESSA's Edge

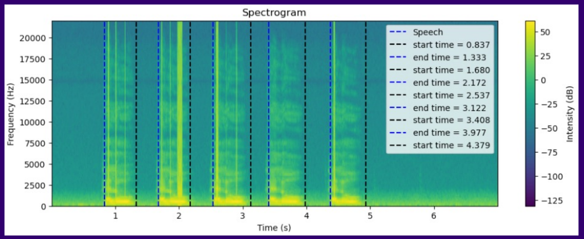
- Optimizes feature extraction with MFCCs
- Advanced HMM training for high-performance, low-energy ASR

Methodology

1. Speech Endpoint Detection

Algorithm: Rabiner and Sambur’s endpoint detection
Steps: Energy calculation, Zero-crossing rate detection and Thresholding





2. Audio Recording

Dataset: Six utterances, 20 samples each in various acoustic environments
Split: 80-20 split for training and validation

3. Feature Extraction (MFCCs)

Features: 26 MFCCs (13 static + 13 delta)

4. HMM Training

Parameters: Initial state distribution (π), State transition probabilities (A), Observation probabilities (B)
Utterances: Odessa, Turn ON the lights, Turn OFF the lights, What time is it, Play Music, Stop Music
Algorithm: Baum-Welch for parameter estimation

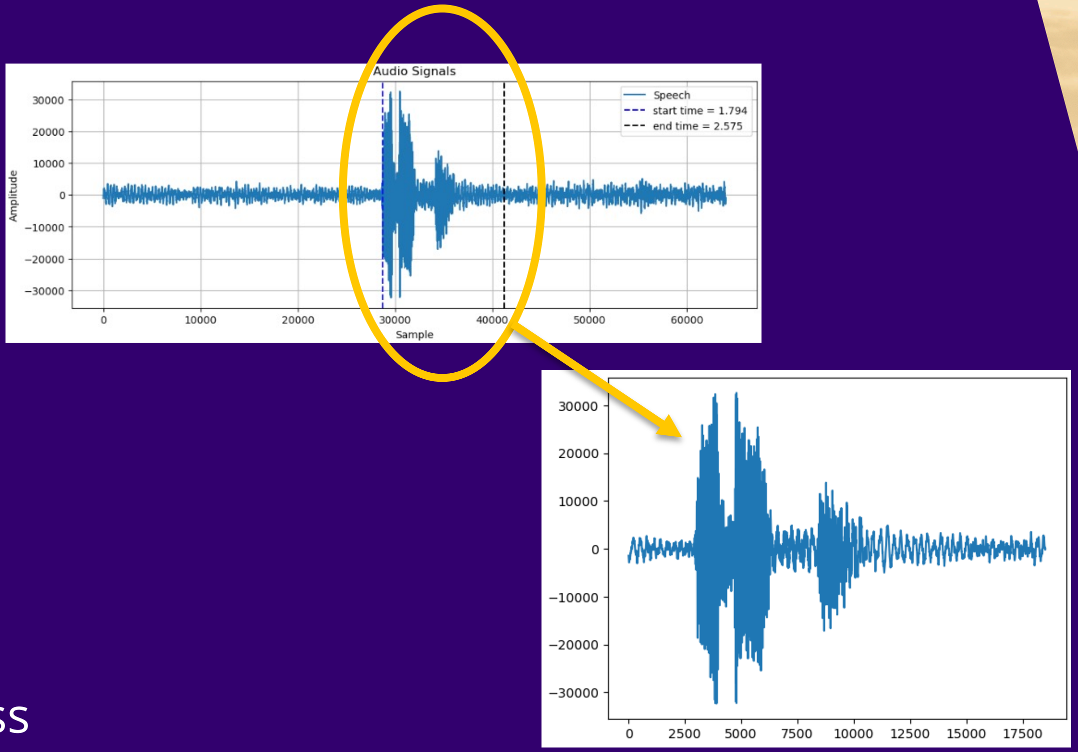
5. Real-Time Implementation

Process: Continuous monitoring, speech detection, feature extraction, HMM model comparison

Results

1. Speech Endpoint Detection

Accuracy: Detected start and end points of speech effectively



2. ASR Performance

1. **Evaluation Metrics**:
   1. Word Error Rate (WER)
   2. Log Likelihood Scores
   3. Viterbi Scores

   (There was not much difference in accuracy with either Viterbi or Loglikelihood. Therefore, Loglikelihood was finally considered)
2. **Results**: Consistently low Word Error Rate across different utterances

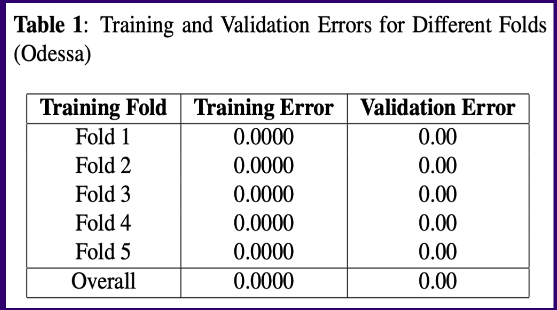


Table 1: Training and Validation Errors for Different Folds (Odessa)

Training Fold	Training Error	Validation Error
Fold 1	0.0000	0.00
Fold 2	0.0000	0.00
Fold 3	0.0000	0.00
Fold 4	0.0000	0.00
Fold 5	0.0000	0.00
Overall	0.0000	0.00

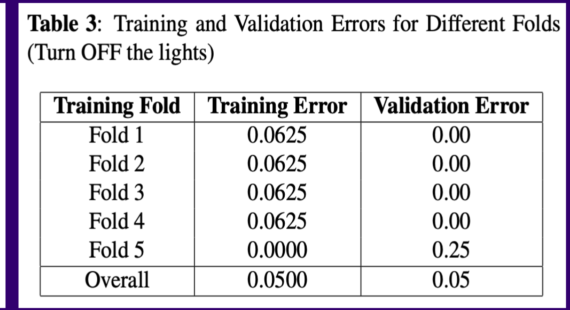


Table 3: Training and Validation Errors for Different Folds (Turn OFF the lights)

Training Fold	Training Error	Validation Error
Fold 1	0.0625	0.00
Fold 2	0.0625	0.00
Fold 3	0.0625	0.00
Fold 4	0.0625	0.00
Fold 5	0.0000	0.25
Overall	0.0500	0.05

3. Real-Time Implementation

Detected 6 utterances effectively

Challenges and Future Work

1. Challenges

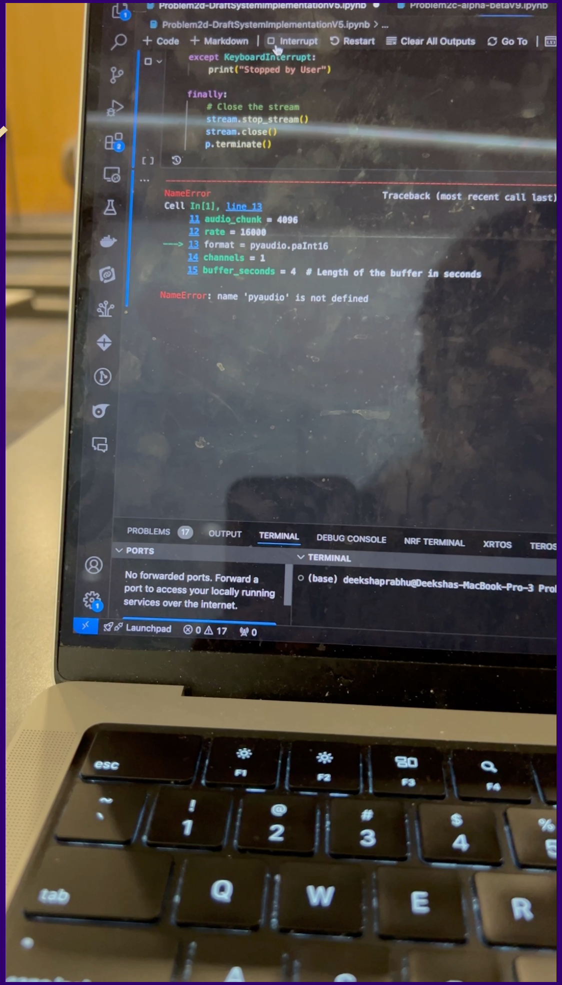
Model Development: Setting up and training the HMM model was complex and time-consuming

Real-Time Implementation: Achieving real-time processing was challenging, despite the low Word Error Rate (WER)

2. Future Work

1. Extended Vocabulary
2. Noise Robustness
3. Speaker Independent System

Video Demo



Problem2d-DraftSystemImplementationV5.ipynb
Problem2c-alpha-betaV9.ipynb
Problem2d-DraftSystemImplementationV5.ipynb > ...
+ Code
+ Markdown
Interrupt
Restart
Clear All Outputs
Go To

```
    except KeyboardInterrupt:
        print("Stopped by User")

    finally:
        # Close the stream
        stream.stop_stream()
        stream.close()
        p.terminate()
```

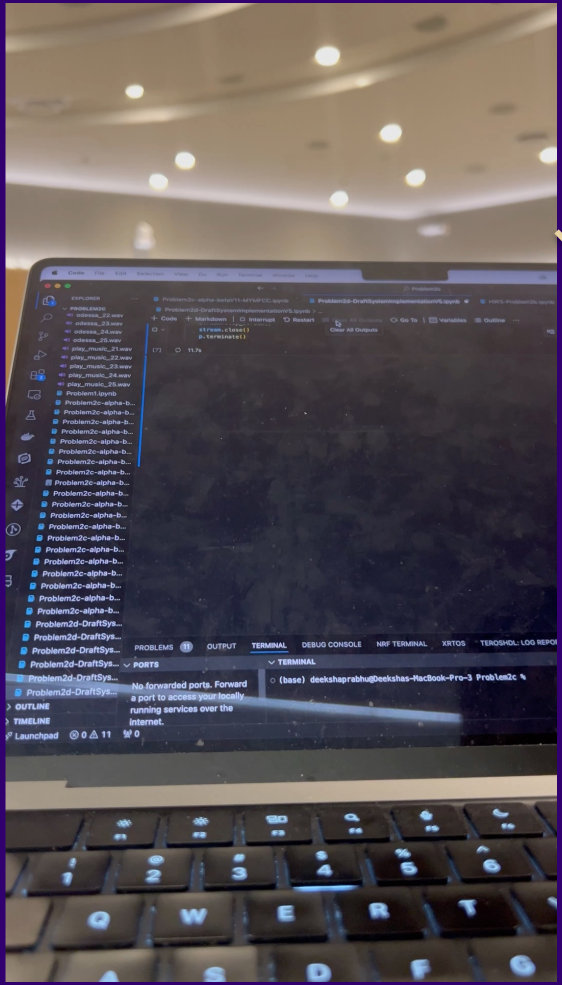
```
NameError                                 Traceback (most recent call last)
Cell In[1], line 13
     11 audio_chunk = 4096
     12 rate = 16000
---> 13 format = pyaudio.paInt16
     14 channels = 1
     15 buffer_seconds = 4  # Length of the buffer in seconds

NameError: name 'pyaudio' is not defined
```

No forwarded ports. Forward a port to access your locally running services over the internet.

```
○ (base) deekshaprabhu@Deekshas-MacBook-Pro-3 Prol
```

This video contains the below utterances and the transitions from Odessa
Odessa → Turn ON the lights
Odessa → Turn OFF the lights
Odessa -→ What time is it ?
Odessa → Play Music



```
stream.close()
p.terminate()
```

No forwarded ports. Forward a port to access your locally running services over the internet.

```
○ (base) deekshaprabhu@Deekshas-MacBook-Pro-3 Problem2c %
```

This video contains the below transition
Play Music → Odessa → Stop Music

DEMO

THANK YOU!