ODESSA: A HMM BASED AUTOMATIC SPEECH RECOGNITION SYSTEM

Deeksha Prabhu

University of Washington, Seattle, WA 98105, USA

ABSTRACT

The ODESSA system leverages Hidden Markov Models (HMMs) to achieve robust and efficient automatic speech recognition. By optimizing feature extraction through Mel-Frequency Cepstral Coefficients (MFCCs) and implementing HMM training techniques, ODESSA balances high accuracy with low computational complexity and energy consumption. This paper presents the design, implementation, and evaluation of ODESSA, demonstrating its effectiveness in real-time speech recognition tasks.

***Index Terms*—** Automatic Speech Recognition, Hidden Markov Model, Mel-Frequency Cepstral Coefficients, Real-Time Processing, Low-Power ASR

1. INTRODUCTION

Automatic Speech Recognition (ASR) systems are crucial for modern human-computer interaction, driving applications like virtual assistants and transcription services. The demand for accurate ASR is growing with advancements in algorithms and computational power. Hidden Markov Models (HMMs) are foundational in ASR, offering robust temporal sequence modeling and capturing speech dynamics. While neural network-based ASR systems, such as Deep Neural Networks (DNNs) and Convolutional Neural Networks (CNNs), provide high accuracy, they require significant computational resources, limiting their use in mobile and embedded systems. HMM-based systems, like ODESSA, offer a compelling alternative with lower computational complexity and power consumption, making them ideal for resource-constrained environments. ODESSA optimizes feature extraction with MFCCs and employs advanced HMM training for high-performance, low-energy ASR.

2. RELATED WORK

The field of ASR has seen substantial advancements, notably in HMM-based systems, neural network approaches, and feature extraction techniques. HMMs, introduced by Rabiner (1989) [1], have been crucial for modeling temporal sequences in ASR. Despite their robustness, traditional HMMs struggle with complex acoustic variations, often addressed by integrating Gaussian Mixture Models (GMMs).

Neural network approaches, including DNNs, CNNs, and RNNs, have advanced ASR by capturing complex acoustic patterns and dependencies, though their high computational requirements can be challenging for resource-limited settings. Feature extraction techniques, such as MFCCs, convert raw speech signals into representative features, mimicking the human auditory system.

Comparative studies show trade-offs between HMM-based and neural network-based ASR, with HMMs favored for computational efficiency and neural networks for accuracy. Recent research explores hybrid models and innovative feature extraction to balance accuracy and efficiency. ODESSA contributes by leveraging HMM robustness and optimizing feature extraction for a high-performance, low-power ASR solution.

3. METHODOLOGY

The development of ODESSA, an HMM-based automatic speech recognition (ASR) system, involves several stages: Speech Start and Endpoint Detection, Audio Recording, Feature Extraction, HMM Training, and Real-Time Implementation. Each stage is crucial for the system's performance and efficiency.

3.1. Speech Start and Endpoint Detection

Detecting the start and end points of speech within an audio signal is critical for ASR. This ensures that only relevant speech segments are analyzed, improving accuracy and efficiency.

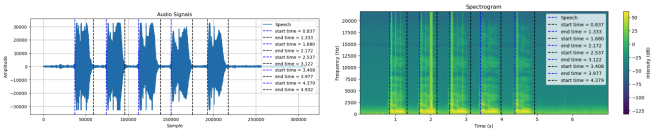
The algorithm by Rabiner and Sambur [2] is used for this purpose. It identifies the start and end of speech by analyzing the energy and zero-crossing rate of the signal. The steps are:

1. **Energy Calculation:** The short-term energy of the speech signal is computed. High energy levels typically correspond to speech activity.
2. **Zero-Crossing Rate:** The zero-crossing rate, which measures the number of times the signal crosses the zero amplitude axis, is calculated. This metric helps to identify the presence of voicing in the signal.

Special thanks to Prof. Jeff Bilmes for his excellent explanations of the course and guidance on the concepts essential to this project.

3. **Thresholding:** Adaptive thresholds are applied to the energy and zero-crossing rate to detect the start and end points of the speech. The thresholds are set based on the statistical properties of the signal, ensuring robust performance across different acoustic environments.

The endpoint detection process outputs a plot of the speech signal with marked start and end points and the corresponding spectrogram as shown in Figure 1.



(a) Speech Start and End Point Detection Plot

(b) Speech Start and End Point Detection Spectrogram

Fig. 1: Speech Start and End Point Detection

3.2. Audio Recording

ODESSA recognizes six utterances: "Odessa," "Turn ON the lights," "Turn OFF the lights," "What time is it," "Play Music," and "Stop Music." Each utterance was recorded 20 times in different acoustic environments, resulting in 120 audio samples. These samples were split 80-20 into training (96 samples) and validation (24 samples) sets to optimize and evaluate the HMM parameters.

3.3. Feature Extraction

Feature extraction is a critical step in ASR. In ODESSA, Mel-Frequency Cepstral Coefficients (MFCCs) are utilized for this purpose due to their ability to capture the essential characteristics of the speech signal.

3.3.1. Mel-Frequency Cepstral Coefficients (MFCCs)

The process of extracting MFCCs involves several steps:

1. **Framing:** Dividing the speech signal into 25ms frames.
2. **Windowing:** Applying a Hamming window to each frame.
3. **Fast Fourier Transform (FFT):** Converting each frame to the frequency domain.
4. **Mel-Scale Filter Bank:** Passing the FFT power spectrum through a mel-scale filter bank.
5. **Logarithm and Discrete Cosine Transform (DCT):** Computing the log of the mel-filtered spectral coefficients and applying DCT to decorrelate features and reduce dimensionality.

The first 13 MFCCs are retained. Additionally, first-order (delta) time derivatives of the MFCCs are computed, resulting in 26 MFCC features (13 static, 13 delta) for ASR input. The dimensionality is NxT, where $N = 26$ and T is the number of time frames.

3.4. HMM Training

The core of ODESSA's ASR system is the Hidden Markov Model (HMM), which models the temporal sequence of speech features. The training process involves estimating the HMM parameters that best represent the speech data.

Consider a discrete-time Markov chain with a finite set of states:

$$S = \{S_1, S_2, \ldots, S_N\}$$

An HMM is defined by the following compact notation to indicate the complete parameter set of the model $\lambda = (\pi, A, B)$, where π, A, B are the initial state distribution vector, the matrix of state transition probabilities, and the set of observation probability distributions in each state, respectively.

$$\Pi = [\pi_1, \pi_2, \ldots, \pi_N], \quad \pi_i = \Pr(q_1 = S_i), \tag{1}$$

$$A = \{a_{ij}\}, \quad a_{ij} = \Pr(q_{t+1} = S_j \mid q_t = S_i), \tag{2}$$

$$\text{for } 1 \leq i, j \leq N, \quad S_i, S_j \in S, \quad t \in [1, 2, \ldots, T]$$

The observation at time t, o_t, is a continuous variable in the case of Continuous HMMs. The observation matrix B is defined as $B = \{b_j(o_t)\}$, where $b_j(o_t)$ is the state conditional probability of the observation o_t defined by:

$$b_j(o_t) = \Pr(o_t = v_k \mid q_t = S_j), \tag{3}$$

for $1 \leq j \leq N, 1 \leq k \leq M$.

For a continuous observation, $b_j(o_t)$ is defined by a Gaussian probability density function (pdf), with state conditional observation mean vector μ_j and state conditional observation covariance matrix Σ_j.

$$\begin{aligned} p(o_t \mid y_t) = & \frac{1}{(2\pi)^{N/2}\sqrt{\prod_{i=1}^{N} \Sigma_{y_t}(i)}} \\ & \times \exp\left[-\frac{1}{2}\sum_{i=1}^{N} \frac{(o_t(i) - \mu_{y_t}(i))^2}{\Sigma_{y_t}(i)}\right] \end{aligned} \tag{4}$$

Thus, B may be defined as:

$$B = \{\mu_j, \Sigma_j\}, \quad i = 1, 2, \ldots, N \tag{5}$$

3.4.1. Model Initialization

In the initial phase of training the HMM, several parameters must be set to appropriate starting values for effective convergence:

1. **Global Mean and Variance Calculation:** For each utterance, the global mean and global variances of all samples are computed.

2. **Initial State Probabilities (π):** The initial state probabilities are uniformly distributed across all states. This uniform initialization ensures that the model does not initially favor any specific state.

3. **Transition Probabilities (A):** The transition probabilities are also initialized using a uniform distribution. This approach ensures an unbiased starting point for the transitions between states.

4. **Mean Vectors (μ):** The mean vectors are initially set to global means with added random noise from a Gaussian distribution (variance 1/8 of the average speech data variance) to promote better convergence.

5. **Covariance Values (σ):** The covariance values are initialized to the global variance of each individual MFCC element, with a single global variance vector for the diagonal covariance matrices to ensure consistency.

3.4.2. Baum-Welch Algorithm

The Baum-Welch algorithm [3], a special case of the Expectation-Maximization (EM) algorithm [4], is widely used for learning or parameter estimation in HMMs. For Gaussian continuous observations, it starts with an initial model $\lambda = (A, \mu_j, \Sigma_j, \Pi)$ and a sequence of observations $O = o_1, o_2, \ldots, o_T$. It estimates the transition matrix A and the observation matrix B as functions of the mean (μ_j) and covariance matrix (Σ_j) that maximize the likelihood of the given observations.

This algorithm involves two main steps:

1. **Expectation Step:** Compute the forward and backward probabilities to estimate the expected state occupancy and transition counts.

2. **Maximization Step:** Update the HMM parameters (initial state distribution, transition probabilities, and emission probabilities) to maximize the likelihood of the observed feature sequences given the current model.

The Baum-Welch algorithm for Gaussian continuous observations can be represented as follows:

$$\alpha_1(j) = \pi_j b_j(o_1), \quad 1 \le j \le N \tag{6}$$

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_t(i) a_{ij}, \quad 1 \le j \le N, 1 \le t \le T-1 \tag{7}$$

$$\Pr(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{8}$$

$$\beta_T(j) = 1, \quad 1 \le j \le N \tag{9}$$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}), \quad 1 \le i \le N, 1 \le t \le T-1 \tag{10}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\Pr(O|\lambda)}, \quad 1 \le i \le N, 1 \le t \le T \tag{11}$$

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} \beta_{t+1}(j) b_j(o_{t+1})}{\Pr(O|\lambda)}, \quad 1 \le i, j \le N, 1 \le t \le T-1 \tag{12}$$

$$\bar{\pi}_i = \gamma_1(i)\beta_1(i), \quad 1 \le i \le N \tag{13}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \le i, j \le N \tag{14}$$

$$\bar{\mu}_j = \frac{\sum_{t=1}^{T} \gamma_t(j) o_t}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N \tag{15}$$

$$\bar{\Sigma}_j = \frac{\sum_{t=1}^{T} \gamma_t(j)(o_t - \mu_j)(o_t - \mu_j)^\top}{\sum_{t=1}^{T} \gamma_t(j)}, \quad 1 \le j \le N \tag{16}$$

Log likelihood scores are used to check convergence. The model parameters (π, A, μ, σ) for each of the 6 utterances are saved in a JSON file.

3.4.3. Parameter Tuning

Hyperparameters such as the number of states M, and tolerance for convergence are adjusted to improve accuracy and minimize errors.

3.5. REAL TIME IMPLEMENTATION

For real-time processing, saved model parameters (π, A, μ, σ) stored in the JSON file are loaded. The PyAudio library continuously monitors the incoming speech signal. Upon detecting speech, the Speech Endpoint Detection algorithm isolates the speech segment. MFCCs are then computed for this segment. These features calculate log likelihood values against six HMM models, with the highest log likelihood model selected as the output.

4. EXPERIMENTAL SETUP AND RESULTS

This section details the experimental setup used to evaluate the ODESSA system, the results obtained from the experiments, and a discussion of these results. The evaluation focuses on the performance of the speech endpoint detection and the accuracy of the automatic speech recognition (ASR) system.

4.1. Experimental Setup

The experimental setup involves recording a dataset of isolated utterances, implementing the endpoint detection algorithm, and evaluating the ASR system's performance using the Hidden Markov Model (HMM) parameters.

4.1.1. Dataset

A dataset comprising six distinct utterances—"Odessa," "Turn ON the lights," "Turn OFF the lights," "What time is it," "Play Music," and "Stop Music" — was recorded in various acoustic environments. Each utterance was recorded 20 times, resulting in a total of 120 audio samples. The dataset was split into training and validation sets using an 80-20 split.

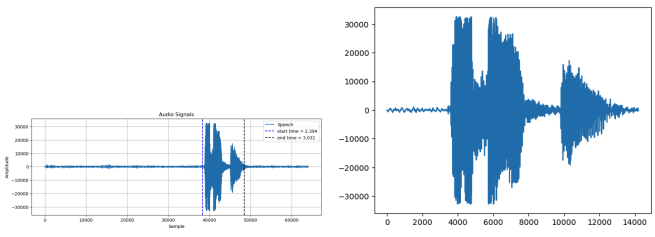
4.1.2. Evaluation Metrics

The performance of the endpoint detection algorithm was measured by observing the plots. The ASR system's performance was evaluated using the word error rate (WER) and the log likelihood scores.

4.2. Results

4.2.1. Speech Endpoint Detection

Figure (Figure 2) shows the speech utterance extracted from the audio buffer.



(a) Speech Buffer (b) Extracted Speech from the buffer

Fig. 2: Extraction of Speech from the audio buffer using Start and End Point Detection

4.2.2. ASR Performance

The ASR system's performance was evaluated using the HMM parameters trained on the dataset. The results, as shown in Tables 1, 2, 3, 4, 5, and 6, indicate the word error rate (WER) for each utterance.

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 2: Training and Validation Errors for Different Folds (Turn ON the lights)

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

4.2.3. Real Time Utterance Detection

Figures 3a and 3b illustrate the detection of the "Odessa" and "Play Music" utterances, respectively. The system successfully identifies and extracts these utterances from the continuous audio stream, demonstrating its real-time capabilities.

The ODESSA ASR system is designed to handle various utterances in real time. The performance is evaluated based on its ability to detect and correctly classify the utterances from a continuous audio stream. The system's robustness is highlighted by its consistent performance across different utterances and acoustic conditions, as shown in Figures 3a and 3b.

Table 4: Training and Validation Errors for Different Folds (What time is it)

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 5: Training and Validation Errors for Different Folds (Play Music)

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

5. CONCLUSION AND FUTURE WORK

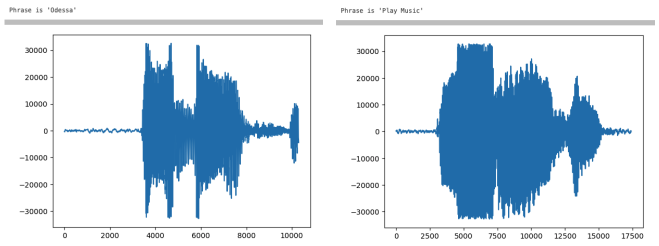
5.1. Conclusion

The ODESSA Automatic Speech Recognition (ASR) system, based on Hidden Markov Models (HMMs), effectively and accurately detects and recognizes speech utterances. Using the Rabiner and Sambur algorithm for speech segment detection ensures that only relevant speech data is processed, enhancing accuracy and efficiency.

Through rigorous training and evaluation, ODESSA achieved high accuracy and low word error rates (WER) in real-time recognition tasks.

5.2. Future Work

While the ODESSA system shows promising results, there are several areas for potential improvement and future research:



(a) Detection of Odessa Utterance (b) Detection of Play Music Utterance

Fig. 3: Detection of speech utterences by ODESSA ASR

Table 6: Training and Validation Errors for Different Folds (Stop Music)

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

- **Extended Vocabulary:** Expanding the system to handle a larger vocabulary and more complex utterances.
- **Noise Robustness:** Enhancing robustness to background noise and varying acoustic environments to improve recognition accuracy.
- **Speaker Independent System:** Developing the system to be speaker-independent by training on a diverse dataset and implementing advanced speaker normalization techniques to ensure accurate recognition regardless of voice, accent, or speaking style.

6. REFERENCES

[1] Lawrence R Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[2] Lawrence R Rabiner and Murray R Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell System Technical Journal*, vol. 54, no. 2, pp. 297–315, 1975.

[3] Leonard E Baum and Lloyd E Welch, "Maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[4] Arthur P Dempster, Nan M Laird, and Donald B Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.