Speech Recognition : Statistical Methods

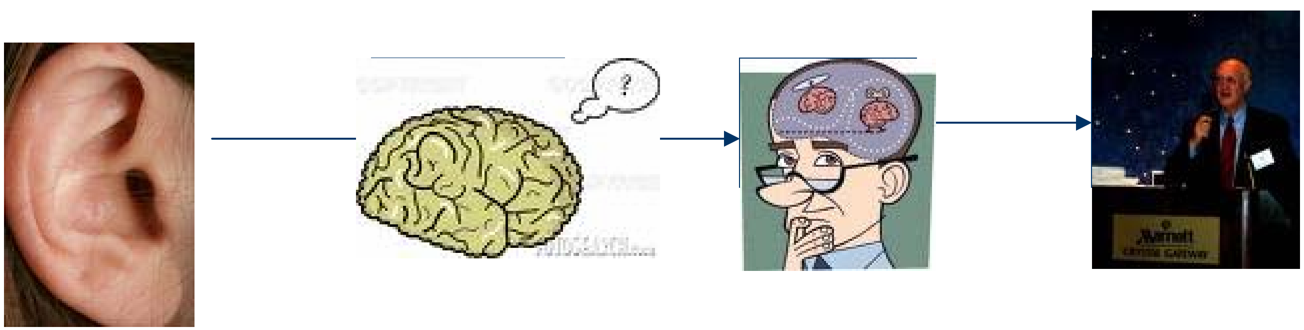
Pankaj Rajan

Agenda

- Motivation.
- Introduction
- Automatic Speech Recognition
- Summary and Discussion.

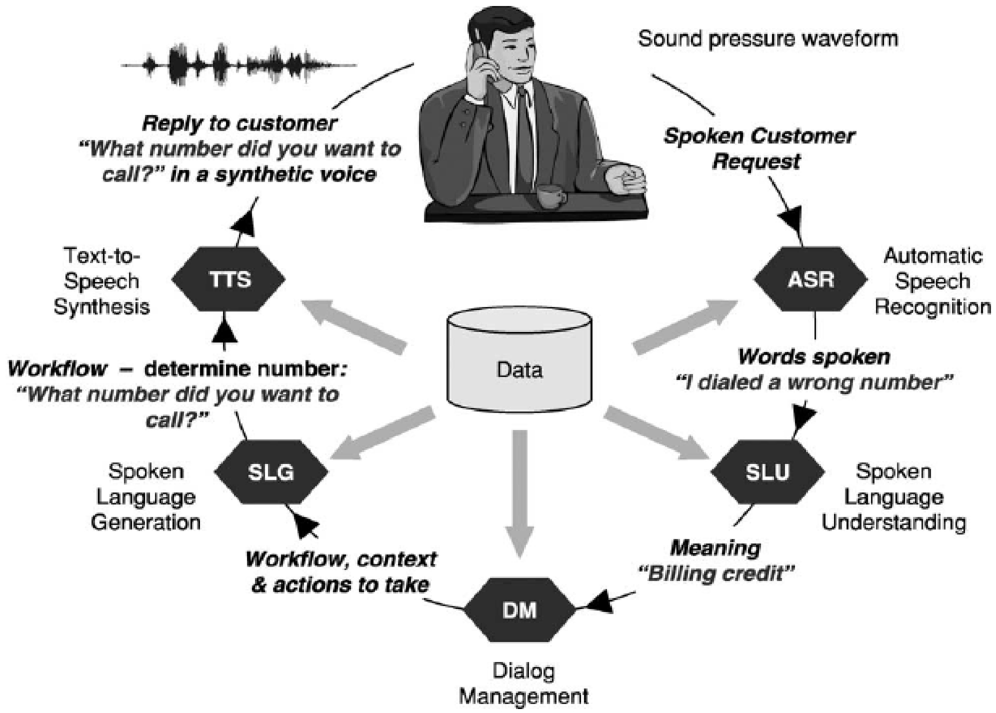
Introduction

- Human Speech Dialog



Hear Process Decide Act/Talk

Dialog With a Machine



What is ASR ?

- The process of recognizing the words in the speech is called *Automatic Speech Recognition or ASR*.

How it works?

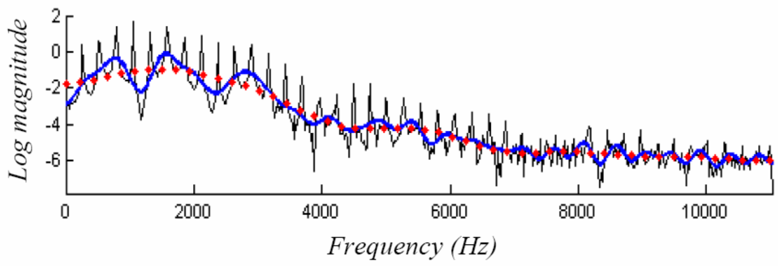
- ASR attempts to decode speech into best estimate of sentences using two steps:

1. Convert the Speech signal into Spectral feature vectors, which are usually measured in the span of 10 ms.
2. Use syntactic decoder to generate every possible valid sentence , and select the best sentence .

Step -1 Deriving the Features

- MFCC are chosen :

  Reason: They match certain characteristics of human perception.



- Feature vector:
  - 13 MFCC.
  - 13 MFCC First derivatives.
  - 13 MFCC second derivatives

Step-2 – Selection of the best sentence

- Mathematically: It can be presented as a MAP estimation problem.

$$W_{\max} = \arg\max_{w} P(W \mid X)$$

Where W is the word string

- Since by Baye’s Rule

$$P(W \mid X) = \frac{P(X \mid W)P(W)}{P(X)}$$

Since P(X) doesn't depend on W :

$$W_{\max} = \arg\max_{w} P(W \mid X) \longrightarrow W_{\max} = \arg\max_{w} P(X \mid W) P(W)$$

Acoustic Model

Language Model

Acoustic Modeling: *Probability Measures*

- Acoustic modeling uses probability measures to characterize sound realization using statistical models.
- Statistical Model called HMM is used for the solution.

  HMM- Hidden Markov Models.
- HMM model the spectral variability of each of he basic sounds using Gaussian distribution which is align with the speech training set and iteratively updated and improved until alignment is achieved.

Markov Property

$P(q_{t+1}=s_1|q_t=s_2) = 1/2$

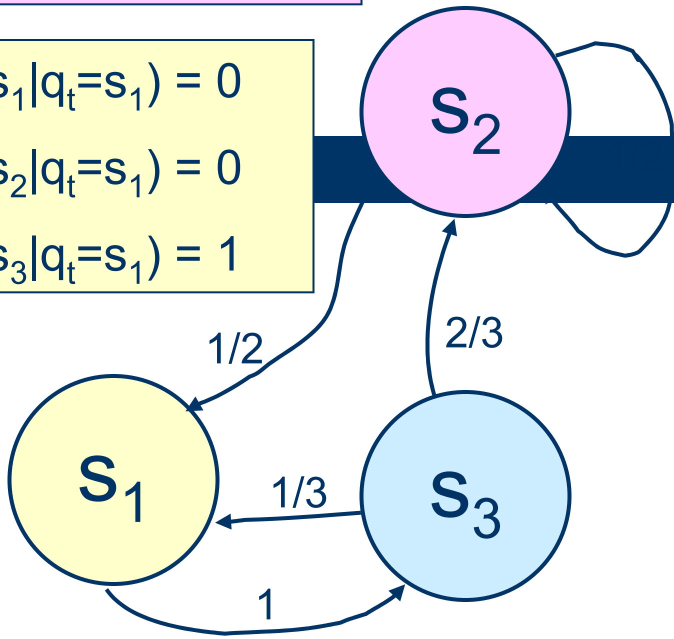
$P(q_{t+1}=s_2|q_t=s_2) = 1/2$

$P(q_{t+1}=s_3|q_t=s_2) = 0$

$P(q_{t+1}=s_1|q_t=s_1) = 0$

$P(q_{t+1}=s_2|q_t=s_1) = 0$

$P(q_{t+1}=s_3|q_t=s_1) = 1$



N = 3

t=1

$q_t=q_1=s_2$

$P(q_{t+1}=s_1|q_t=s_3) = 1/3$

$P(q_{t+1}=s_2|q_t=s_3) = 2/3$

$P(q_{t+1}=s_3|q_t=s_3) = 0$

q_{t+1} is conditionally independent of $\{ q_{t-1}, q_{t-2}, \ldots q_1, q_0 \}$ given q_t.

In other words:

$P(q_{t+1} = s_j | q_t = s_i) =$

$P(q_{t+1} = s_j | q_t = s_i$,any earlier history$)$

The sequence of q is said to be a Markov chain ,or to have the Markov property if the next state depends only upon the current state and not on any past states

From Markov To Hidden Markov

- **The previous model assumes that each state can be uniquely associated with an observable event.**

- **To make the model more flexible, we will assume that the outcomes or observations of the model are a probabilistic function of each state**

Why Hidden???

These are known a Hidden Markov Models (HMM), because the state sequence is not directly observable, it can only be approximated from the sequence of observations produced by the system

HMM Model

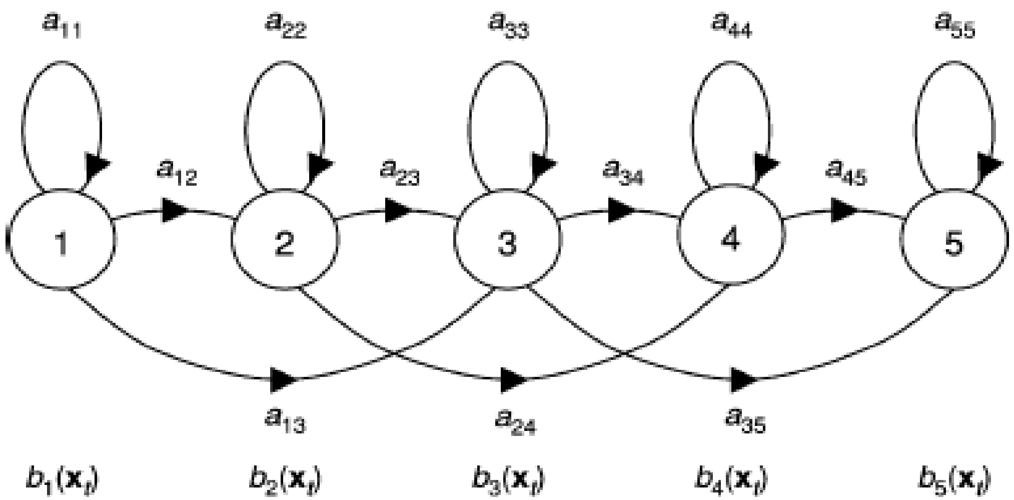
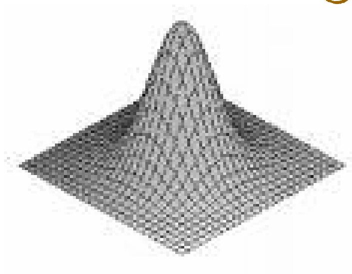


Figure 8 HMM for whole word model with five states.

- Each HMM state in our case is represented by a *Gaussians probability* density function.



.

Requirements for HMM

HMM is specified by a five-tuple (S, O, Π, A, B)

1) $S = \{1, 2, ..., N\}$

Set of hidden states

N: the number of states s_t : the state at time t

2) $O = \{o_1, o_2, ..., o_M\}$

Set of observation symbols

M: the number of observation symbols

3) $\pi = \{\pi_i\}$ $\pi_i = P(s_0 = i)$ $1 \le i \le N$

The initial state distribution

4) $A = \{a_{ij}\}$ $a_{ij} = P(s_t = j \mid s_{t-1} = i),$ $1 \le i, j \le N$

State transition probability distribution

5) $B = \{b_j(k)\}$ $b_j(k) = P(X_t = o_k \mid s_t = j)$ $1 \le j \le N, 1 \le k \le M$

Observation symbol probability distribution in state j

Creation of the Model

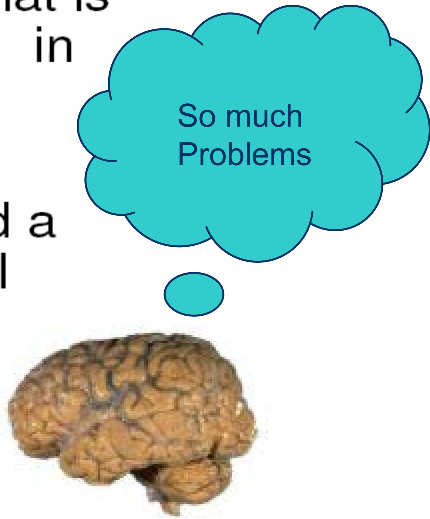
- Once the set of state transitions and state probability densities are specified we say that model φ has been created for the word or sub word unit.

Three Problems

- 1.The Evaluation Problem –Given a model Φ and a sequence of observations $X=(X_1, X_2, ..., X_T)$, what is the probability $P(X \mid \Phi)$; i.e., the probability of the model that generates the observations?

- 2.The Decoding Problem – Given a model Φ and a sequence of observation $X=(X_1, X_2, ..., X_T)$, what is the most likely state sequence $S=(s_0, s_1, ..., s_T)$ in the model that produces the observations?

- 3.The Learning Problem –Given a model Φ and a set of observations, how can we adjust the model parameter $\hat{\Phi}$ to maximize the joint probability $\prod_X P(X \mid \Phi)$?



Solution-Problem1

To calculate the probability (likelihood) $P(\mathbf{X}|\Phi)$ of the observation sequence $\mathbf{X} = (X_1, X_2, \ldots, X_T)$, given the HMM Φ ; the most intuitive way is to sum up the probabilities of all possible state sequences:

$$P(\mathbf{X} \mid \Phi) = \sum_{\text{all } \mathbf{S}} P(\mathbf{S} \mid \Phi) P(\mathbf{X} \mid \mathbf{S}, \Phi)$$

Applying Markov assumption:

$$P(\mathbf{S} \mid \Phi) = P(s_1 \mid \Phi) \prod_{t=2}^{T} P(s_t \mid s_{t-1}, \Phi) = \pi_{s_1} a_{s_1 s_2} \ldots a_{s_{T-1} s_T} = a_{s_0 s_1} a_{s_1 s_2} \ldots a_{s_{T-1} s_T}$$

Continued…….

Applying output independent assumption:

$$P(\mathbf{X} \mid \mathbf{S}, \Phi) = P(X_1^T \mid S_1^T, \Phi) = \prod_{t=1}^{T} P(X_t \mid s_t, \Phi)$$

$$= b_{s_1}(X_1) b_{s_2}(X_2) \ldots b_{s_T}(X_T)$$

$$P(\mathbf{X} \mid \Phi) = \sum_{\text{all } \mathbf{S}} P(\mathbf{S} \mid \Phi) P(\mathbf{X} \mid \mathbf{S}, \Phi)$$

$$= \sum_{\text{all } \mathbf{S}} a_{s_0 s_1} b_{s_1}(X_1) a_{s_1 s_2} b_{s_2}(X_2) \ldots a_{s_{T-1} s_T} b_{s_T}(X_T)$$

Solution- Problem 2 (Viterbi Algorithm)

Viterbi algorithm picks and remembers the best path.

Define the best-path probability:

$$V_t(i) = P(X_1^t, S_1^{t-1}, s_t = i \mid \Phi)$$

$V_t(i)$ is the probability of the most likely state sequence at time t, which has generated the observation X_1^t (until time t) and ends in state i.

Solution –Problem 3

- In order to train the HMM for each sub word unit a labeled training set of words and sentences is used.
- An efficient Training Algorithm Known as Baum-Welch Algorithm is used .
- It aligns various sub word units with spoken inputs and then estimate the appropriate means, Covariance and mixture gains for the distribution.
- The algorithm is a form of hill climbing algorithm and is iterated until a stable alignment of unit model and speech is obtained.

Language Model

- Purpose: To provide task syntax that defines acceptable spoken input sentences and enable computation of probability of the word string.

N- Gram model

- "*n*-gram" or "(*n* − 1)-order Markov model".

- As per wikipedia: "An *n*-gram model predicts x_i based on $x_{i-1}, x_{i-2}, \ldots, x_{i-n}$. When used for language modeling independence assumptions are made so that each word depends only on the last *n* words".

Trigram model

Frequency Count of word triplet occurring in training data

$$P(w_i|w_{i-1},w_{i-2}) = \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1})}$$

Frequency Count of word duplet (w_{i-2},w_{i-1}) occurring in training data

Problem of the Model

N-grams are often highly in error due to problems of data sparseness in the training set. Hence for a text training set of millions of words, and a word vocabulary of several thousand words, more than 50% of word trigrams are likely to occur either once or not at all in the training set

Alternative provided.

$$\hat{P}(w_i|w_{i-1},w_{i-2}) = p_3 \frac{C(w_{i-2},w_{i-1},w_i)}{C(w_{i-2},w_{i-1})} + p_2 \frac{C(w_{i-1},w_i)}{C(w_{i-1})} + p_1 \frac{C(w_i)}{\sum_i C(w_i)}$$

$$p_3 + p_2 + p_1 = 1$$

$$\sum_i C(w_i) = \text{size of text training corpus}$$

Where p_3, p_2, p_1 are Smoothening Probabilities

Language Complexity

- Defined as the average number of words that follow any given word of language
- Mathematically If P(W) is a language model where $W = (w_1, w_2, \ldots, w_Q)$ is a Q word sequence.
- Language Perplexity

$$PP(W) = 2^{H(W)} = P(w_1, w_2, \ldots, w_Q)^{-1/Q}$$
$$\text{as } Q \to \infty.$$

Continued……

- Where H(W) is Entropy of trigram model defined as

$$H(W) = -\frac{1}{Q}\sum_{i=1}^{Q}\log_2 P(w_i|w_{i-1}, w_{i-2})$$

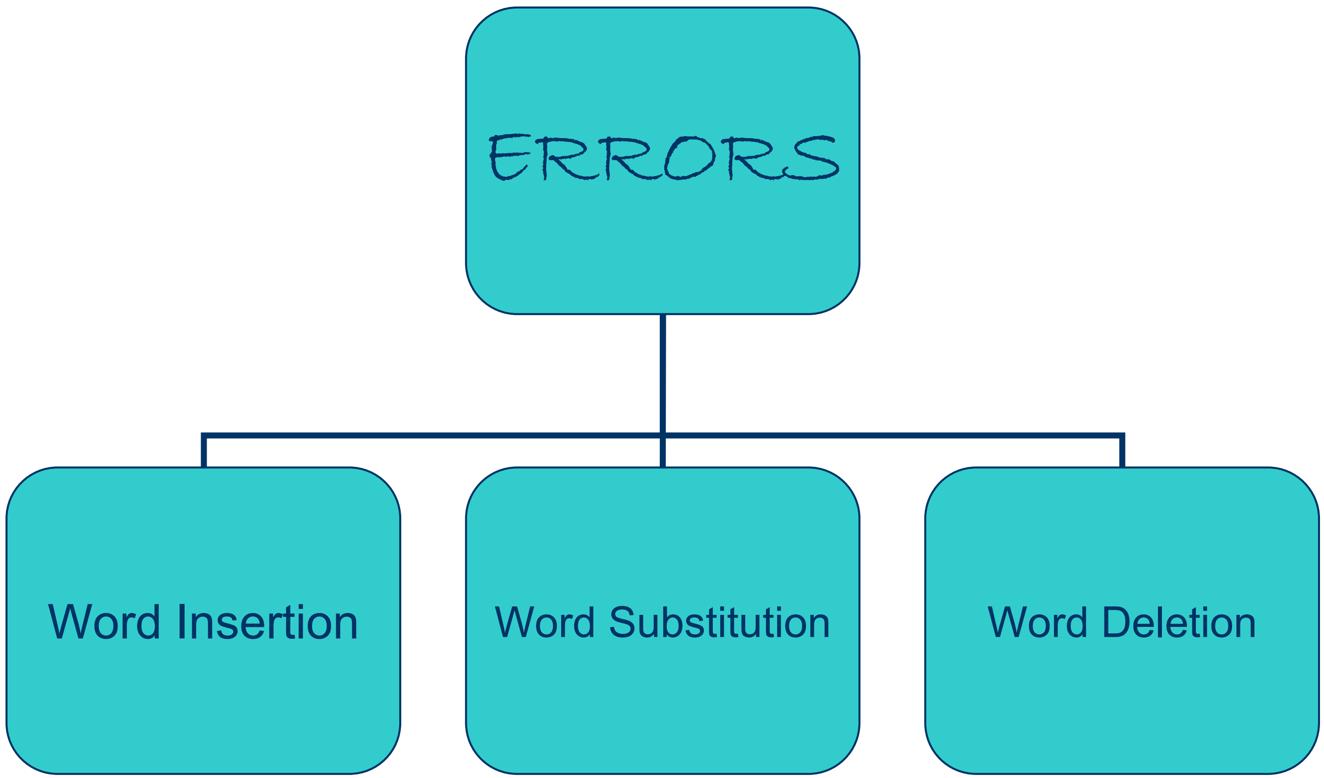
Pattern Matching

- The task of pattern matching module is to combine probabilities from acoustic models , Language model and word Lexicon to find the optimal word Sequence having highest probability among all possible word sequences.

Confidence Scoring

- Goal: To identify the recognition error as well as out of vocabulary events.

Performance of Speech Recognizer



Word Error Rate

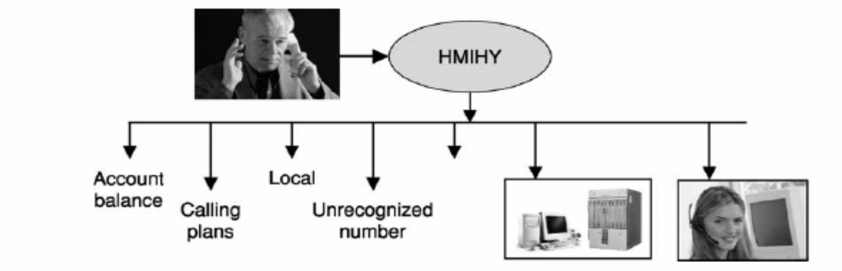
-

$$WER = \frac{NI + NS + ND}{|W|}$$

|W| is the number of words in sentence

Spoken Language Understanding

- Goal: Interpret the meaning of the key words and phrases in the recognized speech.
- Example: " Can I know my bill"



Mathematics behind SLU

$$P(C|W) = P(W|C)P(C)/P(W) \qquad C^* = \arg\max_{C} P(W|C)P(C)$$

Given the word sequence finding the best conceptual structure (meaning) using a combination of acoustic, linguistic and semantic scores.

Example: Verizon Speech recognition assistant.
SLU is useful where Vocabulary set is limited and restricted.

Dialog Management

- Role: To combine meaning of current input speech of user with the current state of the system.
- Decides the next step in the interaction.
- Key tools:

Spoken Language Generation

- Role: To translate the actions of Dialog module into textual representation.

Text to Speech Module

- Role: Converts the text generated by SLG into synthetic naturally sounding speech

Summary

- Paper Introduces us to the ASR that we have experiences most often while calling our wireless or phone company.
- However ASR is too *sensitive* to noise and till now *no* perfect ASR has been *designed* to show good performance under *noisy conditions*