The Big Picture OR
The Components of Automatic Speech Recognition (ASR)

Reference: Steve Young‘s paper - highly recommended!
(online at webpage: http://csl.anthropomatik.kit.edu > Studium und Lehre > SS2013 > Multilinguale Mensch-Maschine Kommunikation)

Donnerstag, 18. April 2013
Overview ASR (I)

- Representation of Speech
- Speech Coding
- Statistical Pattern-based Speech Recognition

- Sampling & Quantization
  - Quantization of Signals
  - Quantization of Speech Signals
  - Sampling Continuous-time Signals
  - How Frequently Should we Sample? - The Aliasing Effect
  - Feature Extraction
Overview ASR (II)

- Automatic Speech Recognition
  - Fundamental Equation of Speech Recognition
  - Acoustic Model
    - Purpose of Acoustic Model (Pronunciation Dictionary)
    - Why breaking down the words into phones
    - Speech Production seen as Stochastic Process
    - Generating an Observation of Speech Features Vectors $x_1, x_2, \ldots, x_T$
    - Hidden Markov Models
      - Formal Definition of Hidden Markov Models
      - Three Main Problems Of Hidden Markov Models
      - Hidden Markov Models in ASR
      - From the Sentence to the Sentence-HMM
      - Context Dependent Acoustic Modeling
      - From Sentence to Context Dependent HMM
Overview ASR (III)

• Automatic Speech Recognition
  – Language Model
    • Motivation
    • What do we expect from Language Models in ASR?
    • Stochastic Language Models
    • Probabilities of Word Sequences
    • Classification of Word Sequence Histories
    • Estimation of N-grams
  – Search
    • Simplified Training
    • Simplified Decoding
    • Comparing Complete Utterances
    • Alignment of Vector Sequences
    • Dynamic Time Warping
Overview – Signal Processing

- Representation of Speech
- Speech Coding
- Statistical Pattern-based Speech Recognition

- Sampling & Quantization
  - Quantization of Signals
  - Quantization of Speech Signals
  - Sampling Continuous-time Signals
  - How Frequently Should we Sample? - The Aliasing Effect
  - Feature Extraction
Automatic Speech Recognition

Input Speech → ??? → Output Text

Hello world
ASR – Signal Processing

Input Speech

Signal Pre-Processing

???

Output Text

Hello world
The purpose of Signal Preprocessing is:

1) **Signal Digitalization (Quantization and Sampling)**
   Represent an analog signal in an appropriate form to be processed by the computer

2) **Digital Signal Preprocessing (Feature Extraction)**
   Extract features that are suitable for recognition process
Representation of Speech

Definition: Digital representation of speech

Represent speech as a sequences of numbers (as a prerequisite for automatic processing using computers)

1) Direct representation of speech waveform:
   represent speech waveform as accurate as possible so that an acoustic signal can be reconstructed

2) Parametric representation
   Represent a set of properties/parameters with regard to a certain model

⇒ Decide the targeted application first:
   • Speech coding
   • Speech synthesis
   • Speech recognition

Classical paper: Schafer/Rabiner in Waibel/Lee (paper online)
Speech Coding

Objectives of Speech Coding:
– Quality versus bit rate
– Quantization Noise
– High measured intelligibility
– Low bit rate (b/s of speech)
– Low computational requirement
– Robustness to transmission errors
– Robustness to successive encode/decode cycles

Objectives for real-time:
– Low coding/decoding delay
– Work with non-speech signals (e.g. touch tone)
Goals for Digital Representation of Speech:

- Capture important phonetic information in speech
- Computational efficiency
- Efficiency in storage requirements
- Optimize generalization
Overview – Signal Processing

- Representation of Speech
- Speech Coding
- Statistical Pattern-based Speech Recognition

- Sampling & Quantization
  - Quantization of Signals
  - Quantization of Speech Signals
  - Sampling Continuous-time Signals
  - How Frequently Should we Sample? - The Aliasing Effect
  - Feature Extraction
Sampling & Quantization

Goal: Given a signal that is continuous in time and amplitude, find a discrete representation.

For it, 2 steps are necessary: sampling and quantization.
- Quantization corresponds to a discretization of the y-axis
- Sampling corresponds to a discretization of the x-axis
Quantization of Signals

- Given a discrete signal $f[i]$ to be quantized into $q[i]$
- Assume that $f$ is between $f_{\text{min}}$ and $f_{\text{max}}$

- Partition $y$-axis into a fixed number $n$ of (equally sized) intervals
- Usually $n=2^b$, in ASR typically $b=16 > n=65536$ (16-bit quantization)
- $q[i]$ can only have values that are centers of the intervals
- **Quantization**: assign $q[i]$ the center of the interval in which lies $f[i]$

- Quantization makes errors, i.e. adds noise to the signal $f[i]=q[i]+e[i]$
- The average **quantization error** $e[i]$ is $(f_{\text{max}}-f_{\text{min}})/(2n)$
- Define **signal to noise ratio** $\text{SNR}[\text{dB}] = \frac{\text{power}(f[i])}{\text{power}(e[i])}$
Quantization of Speech Signals

Choice of sampling depth:

• Speech signals are usually in the range between 50 dB and 60 dB
• The lower the SNR, the lower the speech recognition performance
• To get a reasonable SNR, $b$ should be at least 10 to 12
• Each bit contributes to about 6db of SNR (see e.g. http://cnx.org/content/m0051/latest/)
• Typically in ASR the samples are quantized with 16 bits
Sampling Continuous-time Signals

Original speech waveform and its samples:

![Original speech signal](image1)

![Sampled version of signal](image2)
How Frequently Should we Sample?

Undersampling at 10 kHz:

Input frequency 8 kHz

Resulting frequency 2 kHz
Nyquist or sampling theorem:

- When a $f_i$-band-limited signal is sampled with a sampling rate of at least $2f_i$, then the signal can be exactly reproduced from the samples.
- When the sampling rate is too low, the samples can contain "incorrect" frequencies.

Prevention:

- increase sampling rate
- anti-aliasing filter (restrict signal bandwidth)
Feature Extraction

WHY
• Capture important phonetic information in speech
• Computational efficiency, Efficiency in storage requirements
• Optimize generalization

WHAT
• Features in frequency domain – Reason: It is hard to infer much from time domain waveform
• Human hearing is based on frequency analysis
• Use of frequency analysis simplifies signal processing
• Use of frequency analysis facilitates understanding
Automatic Speech Recognition

Two sessions
Digital Signal Processing

Input Speech

Signal Pre-Processing

???

Output Text

Hello world
Overview

• Automatic Speech Recognition
  – Fundamental Equation of Speech Recognition
  – Acoustic Model
    • Purpose of Acoustic Model (Pronunciation Dictionary)
    • Why breaking down the words into phones
    • Speech Production seen as Stochastic Process
    • Generating an Observation of Speech Features Vectors $x_1, x_2, \ldots, x_T$
    • Hidden Markov Models
      – Formal Definition of Hidden Markov Models
      – Three Main Problems Of Hidden Markov Models
      – Hidden Markov Models in ASR
      – From the Sentence to the Sentence-HMM
      – Context Dependent Acoustic Modeling
      – From Sentence to Context Dependent HMM
Automatic Speech Recognition

Fundamental Equation of Speech Recognition:
Observe a sequence of feature vectors $X$
Find the most likely word sequence $W$

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

Input Speech

Signal Pre-Processing

Output Text

Hello world
The Big Picture OR The components of ASR

Automatic Speech Recognition

\[
\text{arg max } P(W \mid X) = \text{arg max } \frac{P(W)p(X \mid W)}{P(X)}
\]

Signal Pre-Processing

Acoustic Model

Input Speech

\[ p(X \mid W) \]

\[ \Rightarrow \]

Output Text

Hello world
The Big Picture OR The components of ASR

\[
\arg \max_W P(W \mid X) = \arg \max_W \frac{P(W) p(X \mid W)}{P(X)}
\]

Input Speech

Signal Pre-Processing

Acoustic Model

Language Model

\[ p(X \mid W) \]

\[ P(W) \]

Output Text

Hello world
Automatic Speech Recognition

**Search**
how to efficiently try all $W$

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

- **Input Speech**
- **Pre-Processing**
- **Acoustic Model**
- **Language Model**
- **Output Text**

Hello world
Overview

• Automatic Speech Recognition
  – Fundamental Equation of Speech Recognition
  – Acoustic Model
    • Purpose of Acoustic Model (Pronunciation Dictionary)
    • Why breaking down the words into phones
    • Speech Production seen as Stochastic Process
    • Generating an Observation of Speech Features Vectors \( x_1, x_2, \ldots, x_T \)
    • Hidden Markov Models
      – Formal Definition of Hidden Markov Models
      – Three Main Problems Of Hidden Markov Models
      – Hidden Markov Models in ASR
      – From the Sentence to the Sentence-HMM
      – Context Dependent Acoustic Modeling
      – From Sentence to Context Dependent HMM
Automatic Speech Recognition

The Big Picture OR The components of ASR

Input Speech

Signal Pre-Processing

$\text{Acoustic Model}$

$p(X|W)$ $P(W)$

Output Text

Hello world
Purpose of Acoustic Model:
Given W, what is the likelihood to see feature vector(s) X
⇒ we need a representation for W in terms of feature vectors

Usually a two-part representation / modeling:
pronunciation dictionary: describe W as concatenation of phones
Phones models that explain phones in terms of feature vectors

\[ p(X|W) \]
Why breaking down the words into phones

- Need collection of reference patterns for each word
- High computational effort (esp. for large vocabularies), proportional to vocabulary size
- Large vocabulary also means: need huge amount of training data
- Difficult to train suitable references (or sets of references)
- Impossible to recognize untrained words

⇒ Replace whole words by suitable sub units

- Poor performance when the environment changes
- Works only well for speaker-dependent recognition (variations)
- Unsuitable where speaker is unknown and no training is feasible
- Unsuitable for continuous speech (combinatorial explosion)
- Difficult to train/recognize subword units

⇒ Replace the pattern approach by a better modeling process
Automatic Speech Recognition

$\begin{align*}
p(X|W) & \quad P(W) \\
\text{Input Speech} & \rightarrow \text{Output Text}
\end{align*}$

Signal Pre-Processing

Acoustic Model

Output

Hello world
Speech Production seen as Stochastic Process

- The same word / phoneme sounds different every time it is uttered
- Regard words / phonemes as states of a speech production process
- In a given state we can observe different acoustic sounds
- Not all sounds are possible / likely in every state

- We say:
  In a given state the speech process "emits" sounds according to some probability distribution

- The production process makes transitions from one state to another
- Not all transitions are possible, they have different probabilities

⇒ When we specify the probabilities for sound-emissions (emission probabilities) and for the state transitions, we call this a **model**.
The term "hidden" comes from observing observations and drawing conclusions without knowing the *hidden* sequence of states.
Formal Definition of Hidden Markov Models

A Hidden Markov Model is a **five-tuple** consisting of:

- **S** The set of **States** \( S = \{s_1, s_2, \ldots , s_n\} \)
- \( \pi \) The initial probability distribution \( \pi(s_i) = \) probability of \( s_i \) being the first state of a state sequence
- **A** The matrix of **state transition probabilities**: \( A= (a_{ij}) \) where \( a_{ij} \) is the probability of state \( s_j \) following \( s_i \)
- **B** The set of **emission probability** distributions/densities, \( B= \{b_1, b_2, \ldots , b_n\} \) where \( b_i(x) \) is the probability of observing \( x \) when the system is in state \( s_i \)
- **V** The observable **feature space** can be discrete: \( V= \{x_1, x_2, \ldots , x_v\} \), or continuous \( V=\mathbb{R}^d \)
Three Main Problems Of Hidden Markov Models

- The evaluation problem:
  given an HMM $\lambda$ and an observation $x_1, x_2, \ldots, x_T$,
  compute the probability of the observation $p(x_1, x_2, \ldots, x_T | \lambda)$

- The decoding problem:
  given an HMM $\lambda$ and an observation $x_1, x_2, \ldots, x_T$,
  compute the most likely state sequence $s_{q_1}, s_{q_2}, \ldots, s_{q_T}$,
  i.e. $\text{argmax}_{q_1, \ldots, q_T} p(q_1, \ldots, q_T | x_1, x_2, \ldots, x_T, \lambda)$

- The learning / optimization problem:
  given an HMM $\lambda$ and an observation $x_1, x_2, \ldots, x_T$,
  find an HMM $\lambda'$ such that $p(x_1, x_2, \ldots, x_T | \lambda') > p(x_1, x_2, \ldots, x_T | \lambda)$
Hidden Markov Models in ASR

- States that correspond to the same acoustic phaenomenon share the same "acoustic model"
- Training data is better used
- In this HMM: \( b_1 = b_7 = b_{g-b} \)
- Emission prob parameters are estimated more robustly
- Save computation time: (don't evaluate \( b(\cdot \)) for every \( s_i \))
Generate **word lattice** of possible **word sequences**:

Generate **phoneme lattice** of possible **pronunciations**:

Generate **state lattice** (HMM) of possible **state sequences**:
Consider the pronunciations of TRUE, TRAIN, TABLE, and TELL.

Most common lexicon entries are:

<table>
<thead>
<tr>
<th></th>
<th>TRUE</th>
<th>T R UW</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAIN</td>
<td>T R EY N</td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>T EY B L</td>
<td></td>
</tr>
<tr>
<td>TELL</td>
<td>T EH L</td>
<td></td>
</tr>
</tbody>
</table>

Notice that the actual pronunciation sounds a bit like:

<table>
<thead>
<tr>
<th></th>
<th>TRUE</th>
<th>CH R UW</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAIN</td>
<td>CH R EY N</td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>T HH EY B L</td>
<td></td>
</tr>
<tr>
<td>TELL</td>
<td>T HH EH L</td>
<td></td>
</tr>
</tbody>
</table>

Statement: The phoneme T sounds different depending on whether the following phoneme is an R or a vowel.
Context Dependent Acoustic Modeling

First idea:
use actual pronunciations in the lexicon: i.e. CH R UW instead of T R UW.

Problem: The CH in TRUE does sound different from the CH in CHURCH.

Second idea:
Introduce new acoustic units such that the lexicon looks like:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>T(R) R UW</td>
</tr>
<tr>
<td>TRAIN</td>
<td>T(R) R EY N</td>
</tr>
<tr>
<td>TABLE</td>
<td>T(vowel) EY B L</td>
</tr>
<tr>
<td>TELL</td>
<td>T(vowel) EH L</td>
</tr>
</tbody>
</table>

i.e. use context dependent models of the phoneme T
From Sentence to Context Dependent HMM

A context independent HMM for the sentence "HELLO WORLD":

Making the phoneme H dependend on it successor (context dependent),

out of we make

Typical improvements of speech recognizers when introducing context dependence: 30% - 50% fewer errors.
Automatic Speech Recognition

Two lectures on Hidden Markov Modeling
Two lectures on Acoustic Modeling (CI, CD)
One lecture on Pronunciation Modeling, Variants, Adaptation …

\[
p(X|W) \quad P(W)
\]

Input Speech

Signal Pre-Processing

Acoustic Model + Pronunciation Dict

Output

Text

Hello world
Automatic Speech Recognition

The Big Picture OR The components of ASR

Input Speech

Signal Pre-Processing

$p(X|W) \quad P(W)$

Language Model

Output Text

Hello world

p(X|W)        P(W)

Hello world

I /i/ you /j/ /u/ we /v/ /e/
eu sou você é ela é
Overview

• Automatic Speech Recognition
  – Language Model
    • Motivation
    • What do we expect from Language Models in ASR?
    • Stochastic Language Models
    • Probabilities of Word Sequences
    • Classification of Word Sequence Histories
    • Estimation of N-grams
  – Search
    • Simplified Training
    • Simplified Decoding
    • Comparing Complete Utterances
    • Alignment of Vector Sequences
    • Dynamic Time Warping
**Motivation – Language Model**

Equally important to recognize and understand natural speech:

**Acoustic** pattern matching and knowledge about **language**

<table>
<thead>
<tr>
<th>Language Knowledge</th>
<th>in SR covered by:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical knowledge</td>
<td>vocabulary</td>
</tr>
<tr>
<td></td>
<td>dictionary</td>
</tr>
<tr>
<td></td>
<td>LM</td>
</tr>
<tr>
<td>Syntax and Semantics, i.e. rules that determine:</td>
<td>/ Grammar</td>
</tr>
<tr>
<td>word sequence is grammatically well-formed</td>
<td></td>
</tr>
<tr>
<td>word sequence is meaningful</td>
<td></td>
</tr>
<tr>
<td>Pragmatics</td>
<td></td>
</tr>
<tr>
<td>structure of extended discourse</td>
<td></td>
</tr>
<tr>
<td>what is likely to be said in particular context</td>
<td></td>
</tr>
</tbody>
</table>

• These different levels of knowledge are tightly integrated!!!
What do we expect from Language Models in ASR?

• Improve speech recognizer
  add another information source

• Disambiguate homophones
  find out that "I OWE YOU TOO" is more likely than "EYE O U TWO"

• Search space reduction
  when vocabulary is $n$ words, don't consider all $n^k$ possible $k$-word sequences

• Analysis
  analyze utterance to understand what has been said
  disambiguate homonyms (bank: money vs river)
Stochastic Language Models

- In formal language theory $P(W)$ is regarded either as
  - 1.0 if word sequence $W$ is accepted
  - 0.0 if word sequence $W$ is rejected

- Inappropriate for spoken language since,
  - grammar has no complete coverage
  - (conversational) spoken language is often ungrammatical

- Describe $P(W)$ from the **probabilistic** viewpoint
  - Occurrence of word sequence $W$ is described by a probability $P(W)$
  - find a good way to accurately estimate $P(W)$

- **Training problem:**
  - reliably estimate probabilities of $W$

- **Recognition problem:**
  - compute probabilities for generating $W$
Probabilities of Word Sequences

The probability of a word sequence can be decomposed as:
\[ P(W) = P(w_1 \ w_2 \ldots \ w_n) = P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1w_2) \cdots P(w_n | w_1w_2 \ldots w_{n-1}) \]

The choice of \( w_n \) thus depends on the entire history of the input, so when computing \( P(w \mid \text{history}) \), we have a problem:

For a vocabulary of 64,000 words and average sentence lengths of 25 words (typical for Wall Street Journal), we end up with a huge number of possible histories (64,000\(^{25} > 10^{120}\)).

So it is impossible to precompute a special \( P(w \mid \text{history}) \) for every history.

Two possible solutions:
- compute \( P(w \mid \text{history}) \) "on the fly" (rarely used, very expensive)
- replace the history by one out of a limited feasible number of equivalence classes \( C \) such that \( P'(w \mid \text{history}) = P(w \mid C(\text{history})) \)

Question: how do we find good equivalence classes \( C \)?
Classification of Word Sequence Histories

We can use different equivalence classes using information about:

- Grammatical content (phrases like noun-phrase, etc.)
- POS = part of speech of previous word(s) (e.g. subject, object, ...)
- Semantic meaning of previous word(s)
- Context similarity (words that are observed in similar contexts are treated equally, e.g. weekdays, people's names etc.)
- Apply some kind of automatic clustering (top-down, bottom-up)

- Classes are simply based on previous words
  - unigram: \( P'(w_k | w_1w_2 ... w_{k-1}) = P(w_k) \)
  - bigram: \( P'(w_k | w_1w_2 ... w_{k-1}) = P(w_k | w_{k-1}) \)
  - trigram: \( P'(w_k | w_1w_2 ... w_{k-1}) = P(w_k | w_{k-2} w_{k-1}) \)
  - n-gram: \( P'(w_k | w_1w_2 ... w_{k-1}) = P(w_k | w_{k-(n-1)}w_{k-n-2} ... w_{k-1}) \)
Estimation of N-grams

The standard approach to estimate $P(w | \text{history})$ is

- to use a large amount of training corpus (There's no data like more data)
- determine the frequency with which the word $w$ occurs given the $\text{history}$
- simply count how often the word sequence $\text{history}w$ occurs in the text
- normalize the count by the number of times $\text{history}$ occurs

\[
P(w | \text{history}) = \frac{\text{Count}(\text{history}w)}{\text{Count}(\text{history})}
\]

**Example:** Let our training corpus consists of 3 sentences, use bigram model


P(John|<s>) = C(<s>,John) / C(<s>) = 2/3
P(read|John) = C(John,read) / C(John) = 2/2
P(a|read) = C(read,a) / C(read) = 2/3
P(book|a) = C(a,book) / C(a) = 1/2

Now calculate the probability of sentence *John read a book.*


But what about the sentence *Mulan read her book?* - We don’t have P(read|Mulan).
Two lectures on Language Modeling

$p(X|W)$  $P(W)$

Language Model

Input Speech

Signal Pre-Processing

Output Text

Hello world
Overview

- Automatic Speech Recognition
  - Language Model
    - Motivation
    - What do we expect from Language Models in ASR?
    - Stochastic Language Models
    - Probabilities of Word Sequences
    - Classification of Word Sequence Histories
    - Estimation of N-grams
  
- Search
  - Simplified Training
  - Simplified Decoding
  - Comparing Complete Utterances
  - Alignment of Vector Sequences
  - Dynamic Time Warping
Automatic Speech Recognition

The Big Picture OR The components of ASR

Search
how to efficiently try all W

\[
\arg \max_W P(W \mid X) = \arg \max_W \frac{P(W) p(X \mid W)}{P(X)}
\]

Signal Pre-Processing

Input Speech

\[p(X \mid W) \quad P(W)\]

Output Text

Hello world
Search

• The entire set of possible sequences of pattern is called the search space

• Typical search spaces have
  • 1,000 time frames (10sec speech) and 500,000 possible sequences of pattern
  • With an average of 25 words per sentence (e.g. WSJ) and a vocabulary of 64,000 words, more possible word sequences than the universe has atoms!

• It is not feasible to compute the most likely sequence of words by evaluating the scores of all possible sequences

• We need an intelligent algorithm that scans the search space and finds the best (or at least a very good) hypothesis

• This problem is referred to search or decoding
Simplified Training

Aligned Speech

Feature extraction

Speech features

Train Classifier

Improved Classifiers

Use all aligned speech features (e.g. of phoneme /e/) to train the reference vectors of /e/ (=Codebook)
- kmeans
- LVQ

One lecture on Classification
Simplified Decoding

Speech

Feature extraction

Speech features

Decision (apply trained classifiers)

Hypotheses (phonemes)

Speech

Feature extraction

Speech features

Decision

Hypotheses

/h/ /e/ /l/ /o/
/w/ /o/ /r/ /l/ /d/
/h/
Comparing Complete Utterances

What we had so far:
- Record a sound signal
- Compute frequency representation
- Quantize/classify vectors

We now have:
- A sequence of pattern vectors

Want we want:
- The similarity between two such sequences

Obviously: The order of vectors is important!
Comparing Complete Utterances

Comparing speech vector sequences has to overcome three problems:

1) **Speaking rate** characterizes speakers (speaker dependent!)
   if the speaker is speaking faster, we get fewer vectors
2) Changing **speaking rate** by purpose: e.g. talking to a foreign person
3) Changing **speaking rate** non-purposely: speaking disfluencies

⇒ So we have to find a way to decide which vectors
to compare to another
• Impose some **constraints**!
  (compare every vector to all others is too costly)
Alignment of Vector Sequences

First idea to overcome the varying length of Utterances, Problem (2):

1. Normalize their length
2. Make a linear alignment

Linear alignment can handle the problem of different speaking rates

But:
It can not handle the problem of varying speaking rates during the same utterance.
Dynamic Time Warping (DTW)

**Goal:** Identify example pattern that is most similar to unknown input

⇒ compare patterns of different length

Note: all patterns are preprocessed

⇒ 100 vectors / second of speech

**DTW:** Find alignment between unknown input and the example pattern that minimizes the overall distance

Find average vector distance, but which frame-pairs?

Input = unknown pattern

Euclidean Distance
Automatic Speech Recognition

Search how to efficiently try all W

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

Signal Pre-Processing

Input Speech

$p(X \mid W)$
$P(W)$

Output Text
Hello world

The Big Picture OR The components of ASR

Two lectures on Search
Thanks for your interest!