Lectures 36-38:
Automatic Speech Recognition (ASR)

Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332, USA
chl@ece.gatech.edu
Lecture Outline

• Automatic Speech Recognition (ASR)
  – System components and technology dimensions
  – Implementation issues

• Spoken Dialogue Processing
  – System components and technology dimensions
  – Implementation issues

• Theory of Markov Models
  – Discrete Markov processes
  – Hidden Markov processes

• Solutions to the Three Basic Problems of HMM’s
  – Computation of observation probability
  – Determination of optimal state sequence
  – Optimal training of model
**Automatic Speech Recognition (ASR)**

**Goal:** Accurately and efficiently convert a speech signal into a text message independent of the device, speaker or the environment.

**Applications:** Automation of complex operator-based tasks, e.g., customer care, dictation, form filling applications, provisioning of new services, customer help lines, e-commerce, etc.

**Capabilities and Limitations:** Seinfeld
A Brief History of Speech Recognition

- Mathematical Formalization
- Global optimization
- Automatic Learning from data

Speech Recognition Accuracy

- Heuristics
- Handcrafted rules
- Local optimization

HMMs

Time

Cristopher C. discovers a new continent.
Milestones in ASR Technology Research

Small Vocabulary, Acoustic Phonetics-based

1962
Filter-bank analysis; Time-normalization; Dynamic programming

1967
Isolated Words

1972
Medium Vocabulary, Template-based

1977
Isolated Words; Connected Digits; Continuous Speech

1982
Large Vocabulary, Statistical-based

1987
Connected Words; Continuous Speech

1992
Large Vocabulary; Syntax, Semantics

1997
Continuous Speech; Speech Understanding

2002
Very Large Vocabulary; Semantics, Multimodal Dialog, TTS

Spoken dialog; Multiple modalities

Concatenative synthesis; Machine learning; Mixed-initiative dialog

Year
Isolated Word Recognition (IWR)

• Computing word score using a sequence of phone scores

Please say the isolated command now.

Ants: Score = 12.2
EDtv: Score = 32.5
Payback: Score = 29.4
This is a test.

Thousands of training samples are combined to build 40 sub-word models, one for each phoneme.
Automatic Speech Recognition (ASR)

Concept: a sequence of symbols

S₁  S₂  S₃  etc

Speech Waveform

Parameterise

Speech Vectors

Recognise

S₁  S₂  S₃
Voice-Enabled System Technology Components

- **TTS**: Text-to-Speech Synthesis
- **ASR**: Automatic Speech Recognition
- **SLG**: Spoken Language Generation
- **SLU**: Spoken Language Understanding
- **DM**: Dialog Management

**Meaning**

**Speech**

**Data, Rules**

**Words**

**Speech**

**Words**

**Action**

**Speech**

**Meaning**
The Speech Dialog Circle

- **TTS** (Text-to-Speech Synthesis)
  - Words: What’s next?
  - Speech: "Determine correct number"

- **SLG** (Spoken Language Generation)
  - Action: "I dialed a wrong number"
  - Meaning: "Billing credit"

- **ASR** (Automatic Speech Recognition)
  - Words spoken: "I dialed a wrong number"

- **SLU** (Spoken Language Understanding)
  - Spoken Language: "What number did you want to call?"

- **DM** (Dialog Management)
  - Voice reply to customer: "What number did you want to call?"

The cycle continues with the next interaction.
ASR Knowledge Sources

Speech

|---- Acoustic Match -------|

Inventory of Speech Recognition Units

Word Dictionary (In Terms of Chosen Units)

|------ Linguistic Match ------|

Grammar

Task Model

Recognized Sentence
ASR System Components

• Feature Extraction
  – Framing and short-time spectral/cepstral analysis

• Acoustic Modeling of Speech Units
  – Fundamental speech unit selection
  – Statistical pattern matching (HMM unit) modeling

• Lexical Modeling
  – Pronunciation network

• Syntactic and Semantic Modeling
  – Deterministic or stochastic finite state grammar
  – N-gram language model

• Search and Decision Strategies
  – Best-first or depth-first, DP-based search
  – Modular vs. integrated decision strategies
ASR Terminology

• Vocabulary
  – Words that can be recognized in an application
  – More words imply more errors and more computation

• Word spotting
  – Listening for a few specific words within an utterance

• Extraneous speech screening (rejection)
  – Capability to decide whether a candidate key word is a close enough match to be declared a valid key word

• Grammars
  – Syntax (word order) that can be used
  – The way words are put together to form phrases & sentences, some are more likely than others
  – Can be deterministic or stochastic

• Semantics
  – Usually not properly modeled or represented
ASR System and Technology Dimensions

• Isolated word vs. continuous speech recognition
  • Isolated = pauses required between each word
  • Continuous = 0 pauses required
• Speech unit selection: whole vs. sub-word
  • Whole word = requires data collection of the words to be recognized
  • Sub-word = recognizing basic units of speech (phonemes), so word recognition is easier for new applications
• Read vs. spontaneous (degree of fluency)
• Small vs. medium or large vocabulary
• Multilingual vs. dialect/accent variations
Basic ASR Formulation (Bayes Method)

Speaker’s Intention \( W \) \xrightarrow{\text{Speech Production Mechanisms}} s(n) \xrightarrow{\text{Acoustic Processor}} X \xrightarrow{\text{Linguistic Decoder}} \hat{W}

**Speaker Model**

\[
\hat{W} = \arg \max_W P(W | X)
\]

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

\[
= \arg \max_W P_A(X | W) P_L(W)
\]

---

**Step 1**

**Step 2**

**Step 3**
Steps in Speech Recognition

Step 1- acoustic modeling: assign probabilities to acoustic realizations of a sequence of words. Compute $P_A(X/W)$ using hidden Markov models of acoustic signals and words.

Step 2- language modeling: assign probabilities to sequences of words in the language. Train $P_L(W)$ from generic text or from transcriptions of task-specific dialogues.

Step 3- hypothesis search: find the word sequence with the maximum a posteriori probability. Search through all possible word sequences to determine $\arg\max$ over $W$. 
Step 1 - The Acoustic Model

- We build acoustic models by learning statistics of the acoustic features, $X$, from a training set where we compute the variability of the acoustic features during the production of the sounds represented by the models.
- It is impractical to create a separate acoustic model, $P_A(X | W)$, for every possible word in the language -- it requires too much training data for words in every possible context.
- Instead we build acoustic-phonetic models for the ~50 phonemes in the English language and construct the model for a word by concatenating (stringing together sequentially) the models for the constituent phones in the word.
- We build sentences (sequences of words) by concatenating word models.
Step 2-The Language Model

- The language model describes the probability of a sequence of words that form a valid sentence in the language.

- A simple statistical method works well based on a Markovian assumption, namely that the probability of a word in a sentence is conditioned on only the previous N-words, namely an N-gram language model.

\[ P_L(W) = P_L(w_1, w_2, \ldots, w_k) \]

\[ = \prod_{n=1}^{k} P_L(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N}) \]

where \( P_L(w_n \mid w_{n-1}, w_{n-2}, \ldots, w_{n-N}) \) is estimated by simply counting up the relative frequencies \( f \) of \( N \)-tuples in a large corpus of text.
Step 3-The Search Problem

• The **search** problem is one of searching the space of all valid sound sequences, conditioned on the word grammar, the language syntax, and the task constraints, to find the word sequence with the maximum likelihood.

• The **size of the search space** can be astronomically large and take inordinate amounts of computing power to solve by heuristic methods.

• The use of methods from the field of **Finite State Automata Theory** provide **Finite State Networks (FSN)** that reduce the computational burden by orders of magnitude, thereby enabling exact solutions in computationally feasible times, for large ASR problems.
Speech Recognition Processes (I)

• **Choose the task** => sounds, word vocabulary, task syntax (grammar), task semantics

• **Example**: isolated digit recognition task
  – Sounds to be recognized—whole words
  – Word vocabulary—zero, oh, one, two, …, nine
  – Task syntax—any digit is allowed
  – Task semantics—sequence of isolated digits must form a valid telephone number
Speech Recognition Processes (II)

- **Train the Models** => create a method for building *acoustic word models* from a speech training data set, a *word lexicon*, a *word grammar* (language model), and a *task grammar* from a text training data set
  - Speech training set—e.g., 100 people each speaking the 11 digits 10 times in isolation
  - Text data training set—e.g., listing of valid telephone numbers (or equivalently algorithm that generates valid telephone numbers)
Speech Recognition Processes (III)

• **Evaluate performance** => determine word error rate, task error rate for recognizer
  – Speech testing data set—25 new people each speaking 10 telephone numbers as sequences of isolated digits
  – Evaluate digit error rate, phone number error rate

• **Testing algorithm** => method for evaluating recognizer performance from the testing set of speech utterances
Speech Recognition Process

Input Speech

Feature Analysis (Spectral Analysis)

Acoustic Model (HMM)

Language Model (N-gram)

Pattern Classification (Decoding, Search)

Word Lexicon

Confidence Scoring (Utterance Verification)

\[ x_n, W \]

\[ \hat{W} \]

"Hello World"

(0.9) (0.8)
Feature Extraction

**Goal:** extract robust features (information) from the speech that are relevant for ASR.

**Method:** spectral analysis through either a bank-of-filters or through LPC followed by non-linearity and normalization (cepstrum).

**Result:** signal compression where for each window of speech samples where 30 or so cepstral features are extracted (64,000 b/s -> 5,200 b/s).

**Challenges:** robustness to environment (office, airport, car), devices (speakerphones, cellphones), speakers (acents, dialect, style, speaking defects), noise and echo. **Feature set** for recognition—cepstral features or those from a high dimensionality space.
What Features to Use?

- **Short-time Spectral Analysis:**
  - Acoustic features:
    - Mel cepstrum (LPC, filterbank, wavelets), energy
    - Formant frequencies, pitch, prosody
  - **Acoustic-Phonetic features:**
    - Manner of articulation (e.g., stop, nasal, voiced)
    - Place of articulation (e.g., labial, dental, velar)
  - **Articulatory features:**
    - Tongue position, jaw, lips, velum
  - **Auditory features:**
    - Ensemble interval histogram (EIH), synchrony
- **Temporal Analysis:**
  - Approximation of the velocity and acceleration typically through first and second order central differences.
Feature Extraction Process

- Sampling and Quantization
  - Preemphasis
  - Segmentation (blocking)
  - Windowing

- Spectral Analysis
  - Energy Zero-Crossing

- Filtering
  - Noise Removal, Normalization

- Cepstral Analysis
  - Pitch Formants

- Equalization
  - Bias removal or normalization

- Temporal Derivative
  - Delta cepstrum
  - Delta^2 cepstrum
ASR Formulation

- ASR can be viewed as a (noisy) channel decoding or pattern classification problem.
- The solution to ASR (the MAP decision rule):

\[ \hat{W} = \arg \max_{W \in \Gamma} p(W \mid X) = \arg \max_{W \in \Gamma} P(W) \cdot p(X \mid W) \]

\[ = \arg \max_{W \in \Gamma} \overline{P}_{\Gamma}(W) \cdot \overline{p}_\Lambda(X \mid W) \]
ASR Solution

\[ \hat{W} = \arg \max_{W \in \Gamma} \overline{P}_{\Omega}(W) \cdot \overline{p}_{\Lambda}(X \mid W) \]

\[ \overline{p}_{\Lambda}(X \mid W) \] — Acoustic Model (AM): gives the probability of generating feature \( X \) when \( W \) is uttered

– Need a model for every \( W \) to model all speech features from \( W \) à HMM is an ideal model for speech units
  • Sub-word unit is more flexible (better)

\[ \overline{P}_{\Omega}(W) \] — Language Model (LM): gives the probability of \( W \) (word, phrase, sentence) is chosen to say.

– Need a flexible model to calculate the probability for all kinds of \( W \) à Markov chain model (\( n \)-gram)
HMM: An Ideal Speech Model

- Variations in speech signals: temporal & spectral
- Each state represents a process of measurable observations
- Inter-process transition is governed by a finite state Markov chain
- Processes are stochastic and individual observations do not immediately identify the hidden state.

HMM models spectral and temporal variations simultaneously
In a typical system, each phoneme in the language is modeled by a 3-state left-to-right continuous density HMM (CDHMM), and background noise is modeled by a 1-state CDHMM.

Up to thousand of hours of speech data have been used to train HMM’s.
**Acoustic Model**

- **Goal:** map acoustic features into distinct phonetic labels (e.g., /s/, /aa/) or word labels.

- **Hidden Markov Model (HMM):** statistical method for characterizing the spectral properties of speech by a parametric random process. A collection of HMMs is associated with a phone or a word. HMMs are also assigned for modeling extraneous events.

- **Advantages:** powerful statistical method for dealing with a wide range of data and reliably recognizing speech.

- **Challenges:** understanding the role of classification models (ML Training) versus discriminative models (MMI training). What comes after the HMM—are there data driven models that work better for some or all vocabularies.
**Ergodic model** — can get to any state from any other state;

**States are hidden** — observe the effects, not the states.
HMM for Speech

- Phone model: ‘s’

- Word model: ‘is’ (ih-z)

\[ s_1 \rightarrow s_2 \rightarrow s_3 \]

\[ i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \]

‘ih’ ‘Z’
Isolated Word HMM

Left-right HMM – highly constrained state sequences
Basic Problems in HMMs

- Given acoustic observation $X$ and model $\Phi$:

**Evaluation:** compute $P(X \mid \Phi)$

**Decoding:** choose optimal state sequence

**Re-estimation:** adjust $\Phi$ to maximize $P(X \mid \Phi)$
Isolated Word Recognition Using HMM’s

Assume a vocabulary of \( V \) words, with \( K \) occurrences of each spoken word in a training set. Observation vectors are spectral characterizations of the word. For isolated word recognition, we do the following:

1. for each word, \( v \), in the vocabulary, we must build an HMM, \( \lambda^v \), i.e., we must re-estimate model parameters \((A,B,\Pi)\) that optimize the likelihood of the training set observation vectors for the \( v \)-th word. (TRAINING)

2. for each unknown word which is to be recognized, we do the following:
   a. measure the observation sequence \( O = [O_1 O_2 \ldots O_T] \)
   b. calculate model likelihoods, \( P(O|\lambda^v) \), \( 1 \leq v \leq V \)
   c. select the word whose model likelihood score is highest
      \[
      v^* = \arg\max_{1 \leq v \leq V} P(O|\lambda^v)
      \]

Computation is on the order of \( V \cdot N^2 T \) required; \( V = 100, N = 5, T = 40 \)
\[ \Rightarrow 10^5 \text{ computations} \]
Most general form of pdf with a valid re-estimation procedure is:

$$b_j(x) = \sum_{m=1}^{M} c_{jm} \mathcal{D} \left[ x, \mu_{jm}, U_{jm} \right], \quad 1 \leq j \leq N$$

- $x$ = observation vector = $\{x_1, x_2, \ldots, x_D\}$
- $M$ = number of mixture densities
- $c_{jm}$ = gain of $m$-th mixture in state $j$
- $\mathcal{D}$ = any log-concave or elliptically symmetric density (e.g., a Gaussian)
- $\mu_{jm}$ = mean vector for mixture $m$, state $j$
- $U_{jm}$ = covariance matrix for mixture $m$, state $j$

**Conditions:**

- $c_{jm} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq m \leq M$
- $\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N$
- $\int_{-\infty}^{\infty} b_j(x) dx = 1, \quad 1 \leq j \leq N$
HMM Feature Vector Densities

WORD: ZERO, STATE 1

PARAMETER RANGE

COUNT

LOG E
**Word Lexicon**

**Goal:**
Map legal phone sequences into words according to **phonotactic** rules. For example,

David      /d/ /ey/ /v/ /ih/ /d/

**Multiple Pronunciation:**
Several words may have multiple pronunciations. For example

Data       /d/ /ae/ /t/ /ax/
Data       /d/ /ey/ /t/ /ax/

**Challenges:**
How do you generate a word lexicon automatically; how do you add new variant dialects and word pronunciations.
Lexical Modeling

- Assume each HMM $\rightarrow$ a monophone model
  American English: 42 monophone $\rightarrow$ 42 HMMs
  - Concatenation of phone models (phone HMM’s)
  - Lexicon: /science/ = /s/+/ai/+/e/+/n/+/s/ or /s/+/ai/+/n/+/s/
  - Multiple pronunciations and pronunciation network

\[ /s/ \rightarrow /s/ \rightarrow /ai/ \rightarrow /e/ \rightarrow /n/ \rightarrow /s/ \]

\[ /s/ \rightarrow /s/ \rightarrow /n/ \rightarrow /e/ \rightarrow /ai/ \rightarrow /0/ \]
Word-Juncture Modeling

- Co-articulation Effect
  - Simple concatenation of word models (word HMM’s)
  - Hard change: “did you” = /d/+/-i/+/-dzj/+/-u/
  - Soft change: possible pronunciation variations
  - Source of major errors in many ASR systems
  - Easier to handle in syllabic languages with open syllables (vowel or nasal endings, e.g. Japanese, Mandarin)
Modeling Triphone

- Monophone modeling is too simple to model coarticulation phenomenon ubiquitous in speech
- Modeling context-dependent phonemes: triphone
  - American English: 42X42X42 triphones $\rightarrow$ 74,088 HMMs
Grammar Network Expansion: Monophones
Grammar Network Expansion: Triphones
Grammar Network Expansion: Cross-Word
ASR: Viterbi Search

• Assume we build the grammar network for the task, with all trained HMMs attached in the network
• Unknown utterance $\rightarrow$ a sequence of feature vectors $Y$
• Speech recognition is nothing more than a viterbi search:
  – The whole network viewed as a composite HMM $\Lambda$
  – $Y$ viewed as input data, find the optimal path (viterbi path) $S^*$ traversing the whole network (from START to END)

$$S^* = \arg \max_{S \in \Theta} \Pr(S) \cdot p(Y, S | \Lambda) = \arg \max_{S \in \Theta} \Pr(W_S) \cdot p(Y, S | \Lambda)$$

  – Once $S^*$ is found, the recognition results (word sequence) can be derived by backtracking the Viterbi path
ASR Issues

• Training stage:
  – *Acoustic modeling*: how to select speech unit and estimate HMMs reliably and efficiently from available training data.
  – *Language modeling*: how to estimate n-gram model from text training data; handle data sparseness problem.

• Test stage:
  – *Search*: given HMM’s and n-gram model, how to efficiently search the optimal path from the huge grammar network.
    • Search space is extremely large
    • Call for an efficient pruning strategy
Acoustic Modeling

• Selection of Speech Units: modeled by a HMMDigit
  string recognition: a digit by a HMM ➔ 10-12 HMMs
    – Large vocabulary: monophone ➔ biphone ➔ triphone

• HMM topology selection
  – Phoneme: 3-state left-right without skipping state
  – Digit/word: 6-12 states left-right no state skipping

• HMM type selection
  – Top choice: Gaussian mixture CDHMM
  – Number of Gaussian mixtures in each state (e.g., 16)

• HMM parameters estimation:
  – ML (Baum-Welch algorithm) and others
Selecting Speech Segments for each HMM

Monophone HMMs

Reference Segmentation

Triphone HMMs

Reference Segmentation
Reference Segmentation

• Where the segmentation information comes from?
  – Human labeling: tedious, time-consuming;
    • Only a small amount is affordable; used for bootstrap.
  – Automatic segmentation if a simple HMM set is available.
    • Forced-alignment: Viterbi algorithm; Need transcription only
    • HMMs + transcription \(\Rightarrow\) segmentation information

Transcription: This is a test.

Word network
phoneme network
Composite HMM

Run the Viterbi algorithm to backtrack segmentation information
Embedded Training

- Only need transcription for each utterance; no segmentation is needed; automatically tune to optimal segmentation during training

Transcription: This is a test.

Add optional 1-state silence models between words
Isolated Word HMM Recognizer
Choice of Model Parameters

- Left-to-right model preferable to ergodic model (speech is a left-right process)
- Number of states in range 2-40 (from sounds to frames)
  - Order of number of distinct sounds in the word
  - Order of average number of observations in word
- Observation vectors
  - Cepstral coefficients (and their time derivatives) derived from LPC (1-9 mixtures), diagonal covariance matrices
  - Vector quantized discrete symbols (16-256 codebook sizes)
- Constraints on $b_j(O)$ densities
  - $b_j(k) > \varepsilon$ for discrete densities
  - $C_{jm} > \delta$, $U_{jm}(r,r) > \delta$ for continuous densities
Performance vs. Number of States in Model

![Graph showing error rate in percent vs. number of states in HMM](image)
HMM Segmentation for /SIX/

log ENERGY

Σ log f

STATE

FRAME NUMBER

b₁ b₂ b₃ b₄ b₅ = 49 = T
Digit Recognition Using HMM’s

unknown log energy

frame likelihood scores

frame cumulative scores

state segmentation

one

nine

one

nine

one

nine

one

nine

one

nine

one

nine

one

nine

one

nine

one

nine
Digit Recognition Using HMM’s

unknown
log
energy

frame
likelihood
scores

state
segmentation

frame
cumulative
scores

seven

six

seven

seven

six

six
HMM PERFORMANCE ON SPEAKER INDEPENDENT, ISOLATED DIGITS

<table>
<thead>
<tr>
<th>Recognizer Type</th>
<th>Original Training Set</th>
<th>Test Set 2</th>
<th>Test Set 3</th>
<th>Test Set 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPC/DTW</td>
<td>0.1</td>
<td>0.2</td>
<td>2.0</td>
<td>1.1</td>
</tr>
<tr>
<td>LPC/DTW/VQ</td>
<td>–</td>
<td>3.5</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>HMM/VQ</td>
<td>–</td>
<td>3.7</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>HMM/CD</td>
<td>0</td>
<td>0.2</td>
<td>1.3</td>
<td>1.8</td>
</tr>
<tr>
<td>HMM/AR</td>
<td>0.3</td>
<td>1.8</td>
<td>3.4</td>
<td>4.1</td>
</tr>
</tbody>
</table>

AVERAGE DIGIT ERROR RATES (%)

LPC/DTW – Conventional template-based recognizer using dynamic time warping (DTW) alignment.

LPC/DTW/VQ – Conventional recognizer with vector quantization ($M = 64$ codebook).

HMM/VQ – HMM Recognizer with $M = 64$ codebook.

HMM/CD – HMM recognizer using continuous density model with 5 mixtures per state.

MHM/AR – HMM recognizer using mixture autoregressive observation density.
Goal: Model “acceptable” spoken phrases, constrained by task syntax.

Rule-based: Deterministic grammars that are knowledge driven. For example,

\[\text{flying from } \$\text{city} \text{ to } \$\text{city} \text{ on } \$\text{date}\]

Statistical: Compute estimate of word probabilities (N-gram, class-based, CFG). For example

\[\text{flying from Newark to Boston tomorrow}\]

Challenges: How do you build a language model rapidly for a new task?
Formal Grammars

Rewrite Rules:

1. \( S \rightarrow NP \ VP \)
2. \( VP \rightarrow V \ NP \)
3. \( VP \rightarrow AUX \ VP \)
4. \( NP \rightarrow ART \ NP1 \)
5. \( NP \rightarrow ADJ \ NP1 \)
6. \( NP1 \rightarrow ADJ \ NP1 \)
7. \( NP1 \rightarrow N \)
8. \( NP \rightarrow NAME \)
9. \( NP \rightarrow PRON \)
10. \( NAME \rightarrow Mary \)
11. \( V \rightarrow loves \)
12. \( ADJ \rightarrow that \)
13. \( N \rightarrow person \)
From Words to Word Sequences

- Word $\rightarrow$ word sequence $\rightarrow$ beyond
- Syntax model: a huge HMM network (a huge composite HMM) to represent all possible and valid word sequences
  - Finite state approximation of word constraints
  - Deterministic or stochastic finite state grammar
  - Large word network for large problems (e.g., $|V|=60K$)
A Finite-State Grammar Example

- Finite-state grammar map for a simple account query task:
  - Each arc represents a word or phrase except those marked "*" which allow parts of the phrase to be bypassed.
  - This grammar allows phrases such as "Please tell me my checking account balance."

Deterministic or Stochastic FSG
Other Examples of Grammar Network

**Word-loop grammar:**

- For all possible sentences.
- Each branch represents a word in vocabulary
- May add transition probabilities from language models
**N-Grams**

\[ P(W) = P(w_1, w_2, \ldots, w_N) \]
\[ = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2) \cdots P(w_N \mid w_1, w_2, \ldots, w_{N-1}) \]
\[ = \prod_{i=1}^{N} P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \]

**Trigram Estimation**

\[ P(w_i \mid w_{i-1}, w_{i-2}) = \frac{C(w_{i-2}, w_{i-1}, w_i)}{C(w_{i-2}, w_{i-1})} \]
**Example: bi-grams**

(Probability of a word given the previous word)

I want to go from Boston to Denver tomorrow
I want to go to Denver tomorrow from Boston
Tomorrow I want to go from Boston to Denver
with United
A flight on United going from Boston to Denver tomorrow
A flight on United
Boston to Denver
Going to Denver
United from Boston
Boston with United tomorrow
Tomorrow a flight to Denver
Going to Denver tomorrow with United
Boston Denver with United
A flight with United tomorrow

... 0.03 0.02

I want to go from Boston to Denver 1.2e-12

0.02 0.01 0.05 0.01 0.02
Generalization for $N$-grams (back-off)

If the bi-gram $ab$ was never observed, we can estimate its probability:

Probability of word $b$ following word $a$ as a fraction of probability of word $b$
**Goal:** Combine information (probabilities) from the acoustic model, language model and word lexicon to generate an “optimal” word sequence (highest probability).

**Method:** Decoder searches through all possible recognition choices using a Viterbi decoding algorithm.

**Challenges:** How do we build efficient structures (FSMs) for decoding and searching large vocabulary, complex language models tasks;

- features x HMM units x phones x words x sentences can lead to search networks with $10^{22}$ states
- FSM methods can compile the network to $10^8$ states—14 orders of magnitude more efficient

What is the theoretical limit of efficiency that can be achieved?
Goal: Identify possible recognition errors and out-of-vocabulary events. Potentially improves the performance of ASR, SLU and DM.

Method: A confidence score based on a hypothesis test is associated with each recognized word. For example:

<table>
<thead>
<tr>
<th>Label</th>
<th>credit please</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recognized</td>
<td>credit fees</td>
</tr>
<tr>
<td>Confidence</td>
<td>(0.9) (0.3)</td>
</tr>
</tbody>
</table>

\[
P(W \mid X) = \frac{P(W)P(X \mid W)}{\sum_{W} P(W)P(X \mid W)}
\]

Challenges: Rejection of extraneous acoustic events (noise, background speech, door slams) without rejection of valid user input speech.
Robustness

Problem:
a mismatch in the speech signal between the training phase and testing phase can result in performance degradation.

Methods:
traditional techniques for improving system robustness are based on signal enhancement, feature normalization or/and model adaptation.

Perception Approach:
extract fundamental acoustic information in narrow bands of speech. Robust integration of features across time and frequency.
Robust Speech Recognition

- A mismatch in the speech signal between the training phase and testing phase results in performance degradation.
Rejection

Problem:
Extraneous acoustic events, noise, background speech and out-of-domain speech deteriorate system performance.

Measure of Confidence:
Associating word strings with a verification cost that provide an effective measure of confidence (Utterance Verification).

Effect:
Improvement in the performance of the recognizer, understanding system and dialogue manager.
State-of-the-Art Performance

Acoustic Model

Feature Extraction

Pattern Classification (Decoding, Search)

Language Model

Word Lexicon

Confidence Scoring

Recognized Sentence

Speech Recognition (ASR)

Speech Synthesis (TTS)

SLG

DM

SLU

Input Speech

ECE6255 Spring 2010

Center of Signal and Image Processing
Georgia Institute of Technology
How to Evaluate Performance?

- Dictation applications: Insertions, substitutions and deletions

Word Error Rate  = \(100\% \times \frac{\# \text{Subs} + \# \text{Dels} + \# \text{Ins}}{\text{No. of words in the correct sentence}}\)

- Command-and-control: false rejection and false acceptance => ROC curves
# Word Error Rates

<table>
<thead>
<tr>
<th>CORPUS</th>
<th>TYPE</th>
<th>VOCABULARY SIZE</th>
<th>WORD ERROR RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connected Digit Strings--TI Database</td>
<td>Spontaneous</td>
<td>11 (zero-nine, oh)</td>
<td>0.3%</td>
</tr>
<tr>
<td>Connected Digit Strings--Mall Recordings</td>
<td>Spontaneous</td>
<td>11 (zero-nine, oh)</td>
<td>2.0%</td>
</tr>
<tr>
<td>Connected Digits Strings--HMIHY</td>
<td>Conversational</td>
<td>11 (zero-nine, oh)</td>
<td>5.0%</td>
</tr>
<tr>
<td>RM (Resource Management)</td>
<td>Read Speech</td>
<td>1000</td>
<td>2.0%</td>
</tr>
<tr>
<td>ATIS (Airline Travel Information System)</td>
<td>Spontaneous</td>
<td>2500</td>
<td>2.5%</td>
</tr>
<tr>
<td>NAB (North American Business)</td>
<td>Read Text</td>
<td>64,000</td>
<td>6.6%</td>
</tr>
<tr>
<td>Broadcast News</td>
<td>News Show</td>
<td>210,000</td>
<td>13-17%</td>
</tr>
<tr>
<td>Switchboard</td>
<td>Conversational</td>
<td>45,000</td>
<td>25-29%</td>
</tr>
<tr>
<td>Call Home</td>
<td>Conversational</td>
<td>28,000</td>
<td>40%</td>
</tr>
</tbody>
</table>

factor of 17 increase in digit error rate
NIST Benchmark Performance

Year vs. Word Error Rate

- Read Speech
- Conversational Speech
- Broadcast Speech
- Spontaneous Speech
- Varied Microphones
- Noisy
- Resource Management
- NAB
- ATIS

Year:

Word Error Rate:
100% 10% 1%
Accuracy for Speech Recognition

Switchboard/Call Home Vocabulary: 
40,000 words  Perplexity: 85
Human Speech Recognition vs ASR

Machines Outperform Humans

HUMAN ERROR (%)

Digits RM-LM NAB-mic WSJ
RM-null NAB-omni SWBD WSJ-22dB

MACHINE ERROR (%)
Challenges in ASR

System Performance
- Accuracy
- Efficiency (speed, memory)
- Robustness

Operational Performance
- End-point detection
- User barge-in
- Utterance rejection
- Confidence scoring

Machines are 10-100 times less accurate than humans
Voice-Enabled System Technology Components

Text-to-Speech Synthesis

Speech

ASR

Automatic Speech Recognition

Speech

TTS

Data, Rules

Words

SLG

Spoken Language Generation

Words

SLU

Spoken Language Understanding

DM

Dialog Management

Action

Meaning
Spoken Language Understanding (SLU)

Goal: interpret the meaning of key words in the recognized speech string, & map them to actions that the speech understanding system should take

- Accurate understanding can often be achieved without correctly recognizing every word
- SLU makes it possible to offer services where the customer can speak naturally without learning a specific set of terms

Methodology: exploit task grammar (syntax) and semantics to restrict the range of meaning associated with the recognized words; exploit ‘salient’ words to map high information word sequences to appropriate meaning

Performance Evaluation: accuracy of speech understanding system on various tasks and in various operating environments

Applications: automation of complex operator-based tasks, e.g., customer care, catalog ordering, form filling systems, provisioning of new services, customer help lines, etc.

Challenges: what goes beyond simple classifications systems but below full Natural Language voice dialogue systems
# Knowledge Sources for Speech Understanding

<table>
<thead>
<tr>
<th>ASR</th>
<th>SLU</th>
<th>DM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acoustic/Phonetic</td>
<td>Phonotactic</td>
<td>Syntactic</td>
</tr>
<tr>
<td>Relationship of speech sounds and English phonemes</td>
<td>Rules for phoneme sequences and pronunciation</td>
<td>Structure of words, phrases in a sentence</td>
</tr>
<tr>
<td>Acoustic Model</td>
<td>Word Lexicon</td>
<td>Language Model</td>
</tr>
</tbody>
</table>
Speech Recognition Capabilities

Spontaneous Speech
Fluent Speech
Read Speech
Connected Speech
Isolated Words

By B. S. Atal

Vocabulary Size

1997
2001

2
20
200
2000
20000

word spotting
system driven dialogue
user driven dialogue
natural conversation
transcription
office dictation

Name Dialing
voice commands
speaker verification
digit strings

ECE6255, Spring 2010

Center of Signal and Image Processing
Georgia Institute of Technology
Bell Labs Voice Call Transactions

- VRCP
  - 1 B calls per year (1992)
- Voice Prompter
  - 900 M calls/year (1992)
- SDN/NRA
  - 250 M calls/year (1996)
- Universal Card
  - 50 M calls/year (1995)
- MovieFone
  - 40 M calls/year (1999)
- Talking Call Waiting
  - ~110 M calls/year (2000)

Total: ≥ 2 billion calls/year
VRCP: Fully Deployment

• System deployment
  —Fully deployed in the 48 continental states and still being used
  —Known as 0+ service (dialing 0 followed by 10 numbers)

• System Impact
  —Handle over 1B call transactions a year (30M+ per day)
  —Offer a savings of over $300M a year for service providers
  —Stand as the most widely used voice-enabled services as of today
  —Lead to many successful automated speech applications

• A key patent (Lee, Rabiner, and Wilpon) made it possible
  —98% accuracy was obtained within 3 months after the initial trial

• Societal perception
  —General public: no noticeable difference
  —Union workers: system labeled as an evil empire
Spoken Language Translation (C3P-O) and Voice User Interface (R2-D2)

(Speech in Language A) → Speech Recognizer → Language Analyzer → Machine Translation or Dialog Management → Language Generator & TTS Synthesizer → (Speech in Language B)

(Text in Language A) → Language Analyzer → Machine Translation or Dialog Management → Language Generator & TTS Synthesizer → (Speech in Language B)

(Text Understanding in Language A/B) → Machine Translation or Dialog Management → Language Generator & TTS Synthesizer → (Speech in Language B)

(Text Reply in Language B) → Machine Translation or Dialog Management → Language Generator & TTS Synthesizer → (Speech in Language B)

Acoustic & Language Models → Language Analyzer

Semantic Rules → Machine Translation or Dialog Management

Bilingual Databases Translation Models → Language Generator & TTS Synthesizer

Bilingual Databases Translation Models → Language Generator & TTS Synthesizer

Text Analysis & Pronunciation Rules → Language Generator & TTS Synthesizer

Sci-Fi: great inspiration & false expectations
DARPA Communicator

DARPA sponsored R&D of mixed-initiative dialogue systems

Travel task involving airline, hotel and car information and reservation

“Yeah I uh I would like to go from New York to Boston tomorrow night with United”

SLU output (Concept decoding)

Topic: Itinerary
Origin: New York
Destination: Boston
Day of the week: Sunday
Date: May 25th, 2002
Time: >6pm
Airline: United

XML Schema

<itinerary>
  <origin>
    <city></city>
    <state></state>
  </origin>
  <destination>
    <city></city>
    <state></state>
  </destination>
  <date></date>
  <time></time>
  <airline></airline>
</itinerary>
Voice-Enabled System Technology Components

- Text-to-Speech Synthesis (TTS)
- Automatic Speech Recognition (ASR)
- Spoken Language Understanding (SLU)
- Spoken Language Generation (SLG)
- Dialog Management (DM)

Speech: Input and Output for all components.

Data, Rules: Central to the system, managing interactions.

Words: Input and output for SLG and SLU.

Action: Output for DM.

Meaning: Input for DM.
Building Good Speech-Based Applications

• Good user interfaces--make the application easy-to-use and robust to the kinds of confusion that arise in human-machine communications by voice

• Good models of dialog--keep the conversation moving forward, even in periods of great uncertainty on the parts of either the user or the machine

• Matching the task to the technology--be realistic about the capabilities of the technology
  • Fail-soft methods
  • Error correcting techniques (data dips)
User Interface

Good User Interfaces

– Make the application easy-to-use and robust to the kinds of confusion that arise in human-machine communications by voice

– Keep the conversation moving forward, even in periods of great uncertainty on the parts of either the user or the machine

– A great UI cannot save a system with poor ASR and NLU. But, the **UI can make or break a system**, even with excellent ASR & NLU

– Effective UI design is based on a set of elementary principles and common widgets; sequenced screen presentations; simple error-trap dialog; a user manual
Multimodal System Technology Components

- TTS (Text-to-Speech Synthesis)
- ASR (Automatic Speech Recognition)
- SLG (Spoken Language Generation)
- SLU (Spoken Language Understanding)
- DM (Dialog Management)

Visual

Speech

Pen Gesture

Data, Rules

Words

Action

Meaning

Text-to-Speech Synthesis

Spoken Language Generation

Spoken Language Understanding

Automatic Speech Recognition
Multimodal Experience (iPhone)

• Users need to have access to automated dialog applications at anytime and anywhere using any access device

• Selection of “most appropriate” UI mode or combination of modes depends on the device, the task, the environment, and the user’s abilities and preferences
Microsoft MiPad

Multimodal Interactive Pad

• Usability studies show double throughput for English inputs
• Speech is mostly useful in cases with lots of alternatives
• Demo (MiPad)
The Speech Advantage

• **Reduce costs**
  – Reduce labor expenses while still providing customers an easy-to-use and natural way to access information and services

• **New revenue opportunities**
  – 24x7 high-quality customer care automation
  – Access to information without a keyboard or touch-tones

• **Customer retention**
  – Provide personal services for customer preferences
  – Improve customer experience
Voice-Enabled Services

**Desktop applications** -- dictation, command and control of desktop, control of document properties (fonts, styles, bullets, …)

**Agent technology** – simple tasks like stock quotes, traffic reports, weather; access to communications, e.g., voice dialing, voice access to directories (800 services); access to messaging (text and voice messages); access to calendars and appointments

**Voice Portals** – ‘convert any web page to a voice-enabled site’ where any question that can be answered on-line can be answered via a voice query; protocols like VXML, SALT, SMIL, SOAP and others are key

**E-Contact services** – Call Centers, Customer Care (HMIHY) and Help Desks where calls are triaged and answered appropriately using natural language voice dialogues

**Telematics** – command and control of automotive features (comfort systems, radio, windows, sunroof)

**Small devices** – control of cellphones, PDAs from voice commands
Future of ASR Technologies

- Very Large Vocabulary, Limited Tasks, Controlled Environment
- Very Large Vocabulary, Limited Tasks, Arbitrary Environment
- Unlimited Vocabulary, Unlimited Tasks, Many Languages

Dialog Systems
Robust Systems
Multilingual Systems; Multimodal Speech Enabled Devices

Year
Summary

• Today’s class
  – Automatic speech recognition (ASR)

• Next classes
  – Plenty to follow up in research

• Final Exam: 2:50-5:40pm, May 5 (Wednesday)

• Project: presentations on 4/26 -4/30
  – Project Report due by noon on May 5, Thursday

• Reading assignment