Automatic Speech Recognition
The Noisy Channel Model

- **Automatic speech recognition** (ASR) is a process by which an acoustic speech signal is converted into a set of words [Rabiner et al., 1993]

- **The noisy channel model** [Lee et al., 1996]
  - Acoustic input considered a noisy version of a source sentence

![Diagram of the Noisy Channel Model]

 버스 정류장이 어디에 있나요?

 버스 정류장이 어디에 있나요?
The Noisy Channel Model

- What is the most likely sentence out of all sentences in the language $L$ given some acoustic input $O$?
- Treat acoustic input $O$ as sequence of individual observations
  - $O = o_1, o_2, o_3, ..., o_t$
- Define a sentence as a sequence of words:
  - $W = w_1, w_2, w_3, ..., w_n$

$$
\hat{W} = \arg \max_{W \in L} P(W | O)
$$

Bayes rule

$$
\hat{W} = \arg \max_{W \in L} P(O | W)P(W) / P(O)
$$

Golden rule

$$
\hat{W} = \arg \max_{W \in L} P(O | W)P(W)
$$

IlVB-2006 Tutorial
Speech Recognition Architecture Meets Noisy Channel

\[ \hat{W} = \arg \max_{W \in L} P(O | W) P(W) \]

Speech Signals \( O \) → Feature Extraction \( \rightarrow \) Decoding \( \rightarrow \) Word Sequence \( W \)

Speech DB → HMM Estimation → Decoding

Text Corpora → G2P → Pronunciation Model

- Acoustic Model
- Pronunciation Model
- Language Model

버스 정류장이 어디에 있나요?
Feature Extraction

- **The Mel-Frequency Cepstrum Coefficients (MFCC) is a popular choice** [Paliwal, 1992]
  - Frame size: 25ms / Frame rate: 10ms
  - 39 feature per 10ms frame
    - Absolute: Log Frame Energy (1) and MFCCs (12)
    - Delta: First-order derivatives of the 13 absolute coefficients
    - Delta-Delta: Second-order derivatives of the 13 absolute coefficients
Acoustic Model

- Provide \( P(O|Q) = P(\text{features}|\text{phone}) \)
- Modeling Units [Bahl et al., 1986]
  - Context-independent: Phoneme
  - Context-dependent: Diphone, Triphone, Quinphone
    - \( p_L-p+p_R \): left-right context triphone
- Typical acoustic model [Juang et al., 1986]
  - Continuous-density Hidden Markov Model \( \lambda = (A, B, \pi) \)
  - Distribution: Gaussian Mixture
    \[
    b_j(x_j) = \sum_{k=1}^K c_{jk} N(x_j; \mu_{jk}, \Sigma_{jk})
    \]
  - HMM Topology: 3-state left-to-right model for each phone, 1-state for silence or pause
Pronunciation Model

- Provide $P(Q|W) = P(\text{phone}|\text{word})$
- Word Lexicon [Hazen et al., 2002]
  - Map legal phone sequences into words according to phonotactic rules
  - G2P (Grapheme to phoneme) : Generate a word lexicon automatically
  - Several word may have multiple pronunciations
- Example
  - Tomato

- $P([\text{t}][\text{ow}]) = P([\text{t}][\text{ow}]) = 0.1$
- $P([\text{t}][\text{ah}][\text{m}][\text{ey}][\text{a}][\text{t}][\text{ow}]) = 0.4$
Training

• Training process [Lee et al., 1996]

  Speech DB → Feature Extraction → Baum-Welch Re-estimation → Converged?

  HMM

  yes → End

  no → HMM

• Network for training

  Sentence HMM

  ONE TWO THREE ONE

  Word HMM

  ONE

  Phone HMM

  W
Language Model

- Provide \( P(W) \); the probability of the sentence [Beaujard et al., 1999]
  - We saw this was also used in the decoding process as the probability of transitioning from one word to another.
  - Word sequence: \( W = w_1, w_2, w_3, ..., w_n \)
  \[
P(w_1 \cdots w_n) = \prod_{i=1}^{n} P(w_i | w_1 \cdots w_{i-1})
\]
  - The problem is that we cannot reliably estimate the conditional word probabilities, \( P(w_i | w_1 \cdots w_{i-1}) \) for all words and all sequence lengths in a given language
  - n-gram Language Model
    - n-gram language models use the previous n-1 words to represent the history
    \[
P(w_i | w_1 \cdots w_{i-1}) = P(w_i | w_{i-(n-1)} \cdots w_{i-1})
\]
    - Bi-grams are easily incorporated in a viterbi search
Language Model

- Example
  - Finite State Network (FSN)
  - Context Free Grammar (CFG)
  - Bigram

\[
P(\text{에서|서울})=0.2 \quad P(\text{세시|에서})=0.5 \\
P(\text{출발|세시})=1.0 \quad P(\text{하는|출발})=0.5 \\
P(\text{출발|서울})=0.5 \quad P(\text{도착|대구})=0.9 \\
... \\
\]
Network Construction

- Expanding every word to state level, we get a search network [Demuynck et al., 1997]

![Diagram of search network with Acoustic Model, Pronunciation Model, and Language Model](image)

**Acoustic Model**
- Expanding every word to state level, we get a search network [Demuynck et al., 1997]

**Pronunciation Model**
- Expanding every word to state level, we get a search network [Demuynck et al., 1997]

**Language Model**
- Expanding every word to state level, we get a search network [Demuynck et al., 1997]

**Search Network**
- Expanding every word to state level, we get a search network [Demuynck et al., 1997]
Decoding

- Find $\hat{W} = \arg \max_{W \in L} P(W | O)$

- **Viterbi Search**: Dynamic Programming
  - Token Passing Algorithm [Young et al., 1989]

  - Initialize all states with a token with a null history and the likelihood that it’s a start state
  - For each frame $a_k$
    - For each token $t$ in state $s$ with probability $P(t)$, history $H$
      - For each state $r$
        - Add new token to $s$ with probability $P(t) P_{s,r} P_r(a_k)$, and history $s.H$
Decoding

• Pruning [Young et al., 1996]
  – Entire search space for Viterbi search is much too large
  – Solution is to prune tokens for paths whose score is too low
  – Typical method is to use:
    – histogram: only keep at most n total hypotheses
    – beam: only keep hypotheses whose score is a fraction of best score

• N-best Hypotheses and Word Graphs
  – Keep multiple tokens and return n-best paths/scores
  – Can produce a packed word graph (lattice)

• Multiple Pass Decoding
  – Perform multiple passes, applying successively more fine-grained language models
Large Vocabulary Continuous Speech Recognition (LVCSR)

- Decoding continuous speech over large vocabulary
  - Computationally complex because of huge potential search space

- **Weighted Finite State Transducers (WFST)** [Mohri et al., 2002]
  - Efficiency in time and space
    - Word : Sentence
    - Phone : Word
    - HMM : Phone
    - State : HMM

- Dynamic Decoding
  - On-demand network constructions
  - Much less memory requirements