Automatic Speech Recognition - Part II

Dr. Mathew Magimai Doss

September 21, 2022

Outline

Recap

Statistical model-based approach

Training

Summary

Outline

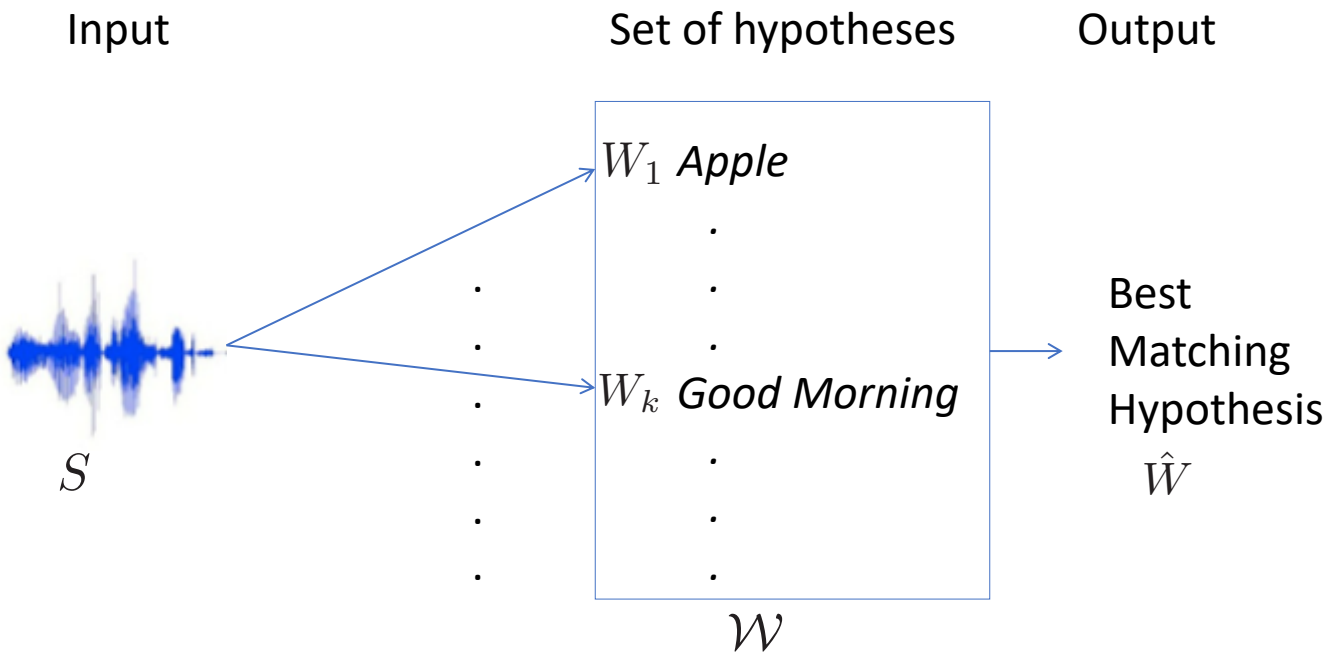
Recap

Statistical model-based approach

Training

Summary

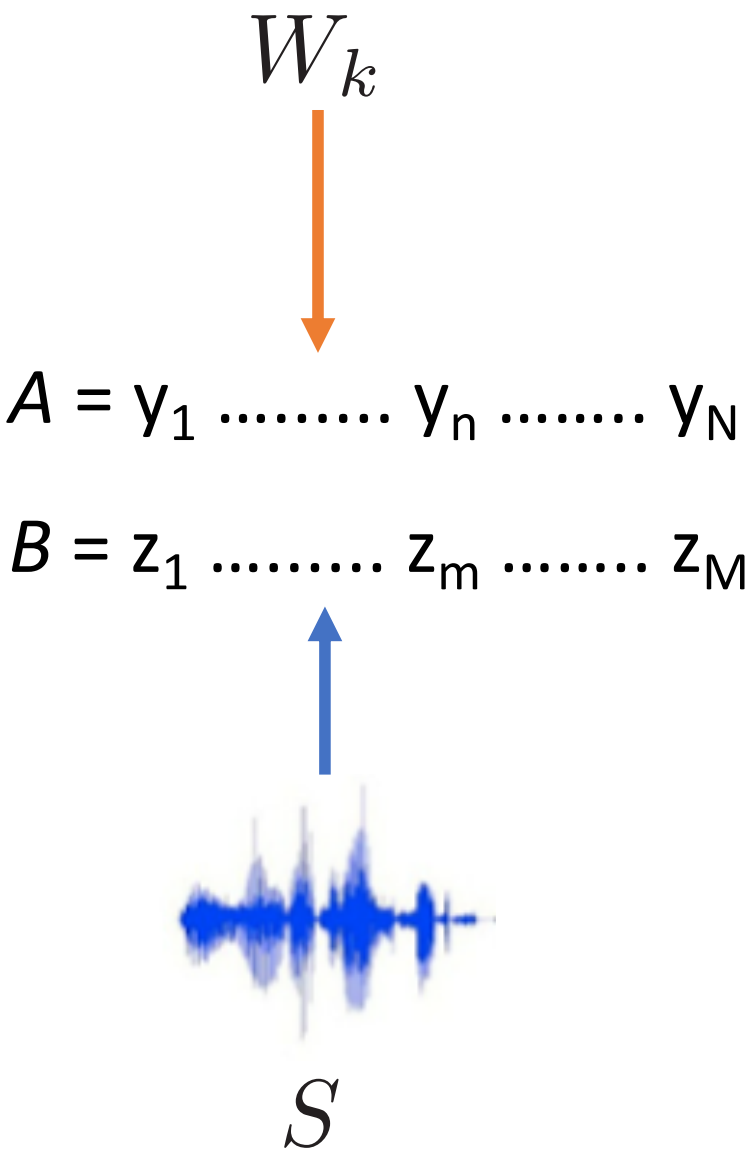
Automatic speech recognition (ASR)



$$\hat{W} = \underset{W_k \in \mathcal{W}}{\arg\max}\, \mathrm{Match}(W_k, S)$$

How to match an observed speech signal S with a word hypothesis W_k?

Abstract formulation for matching S and W_k

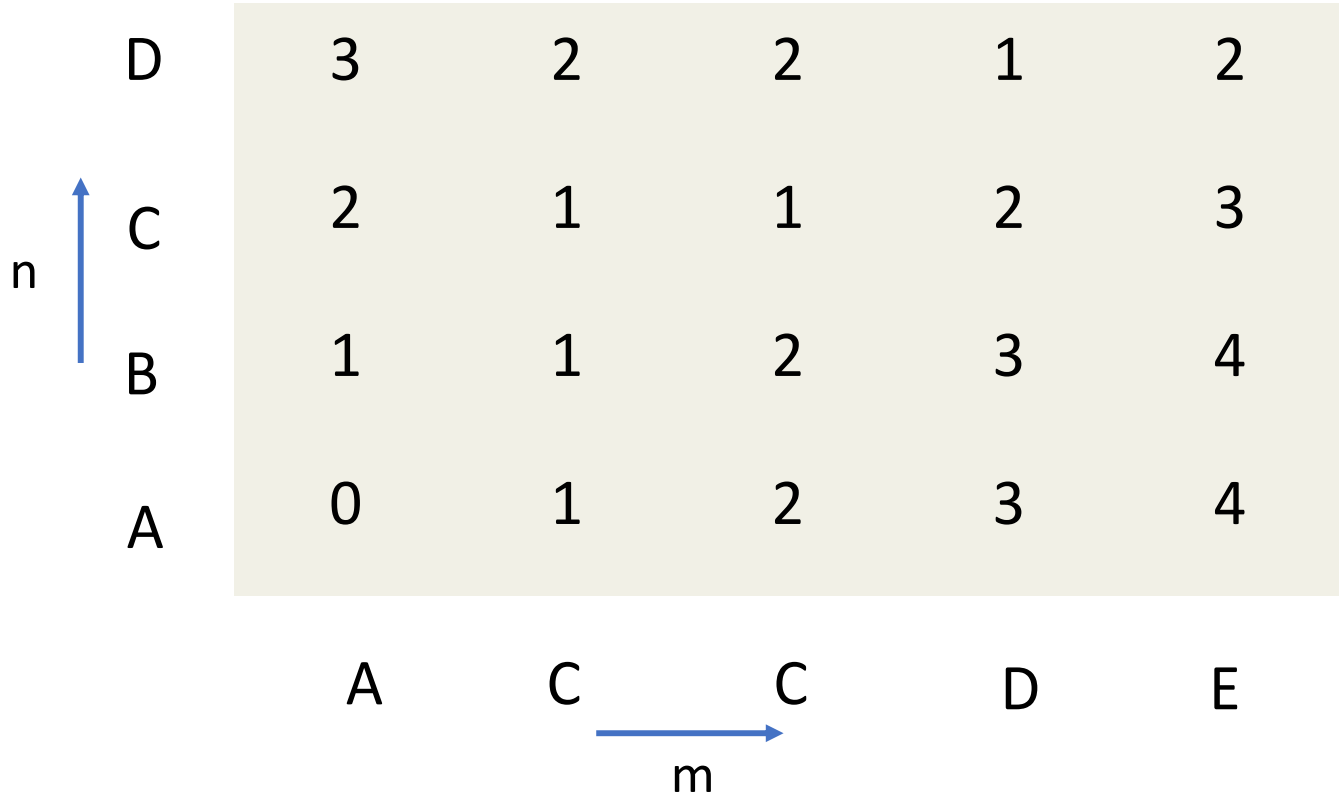


Core Idea

1. Map S and W_k to a shared latent symbol space
2. Match the resulting two latent symbol sequences A and B

String Matching

Matching two sequences of symbols through dynamic programming



symbols can be alphabets of a language or words in a language

Four sub questions

Q1: What is the shared latent symbol set?

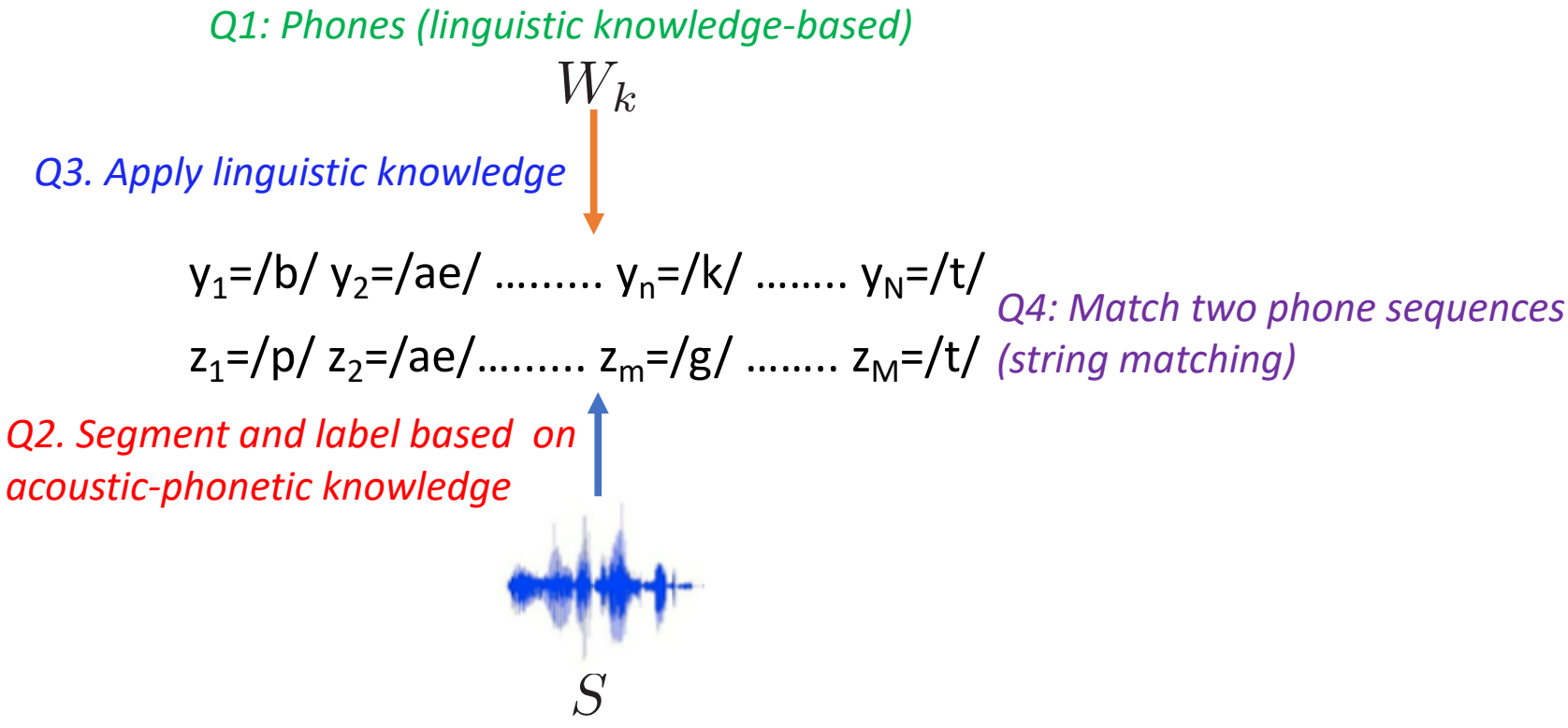
Q2: How to map S to a latent symbol sequence B?

Q3: How to map W_k to a latent symbol sequence A?

Q4: How to match the two latent symbol sequences A and B?

Different ASR methods mainly differ on how these four sub questions are addressed.

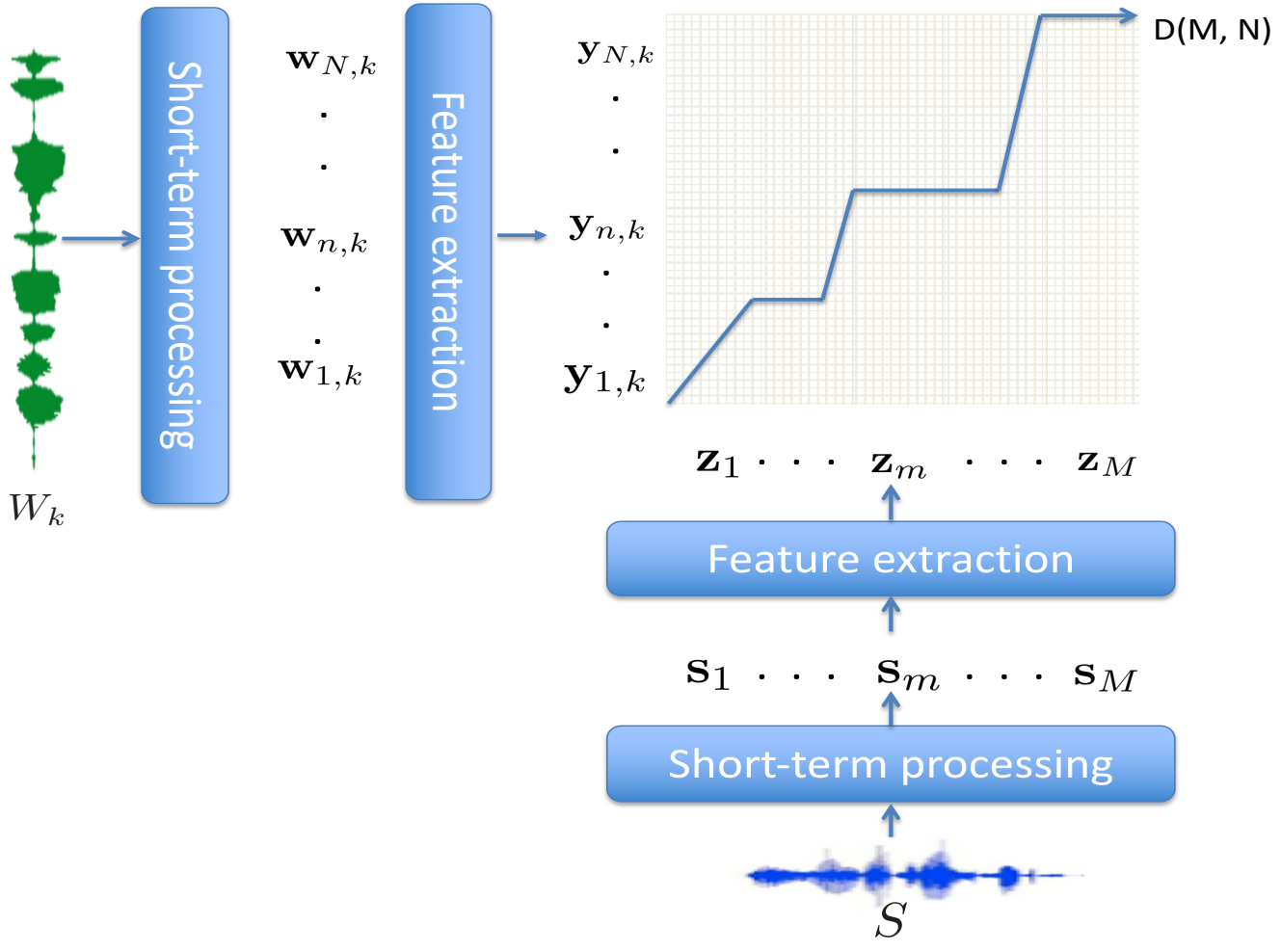
Knowledge-based ASR approach



Limitations:

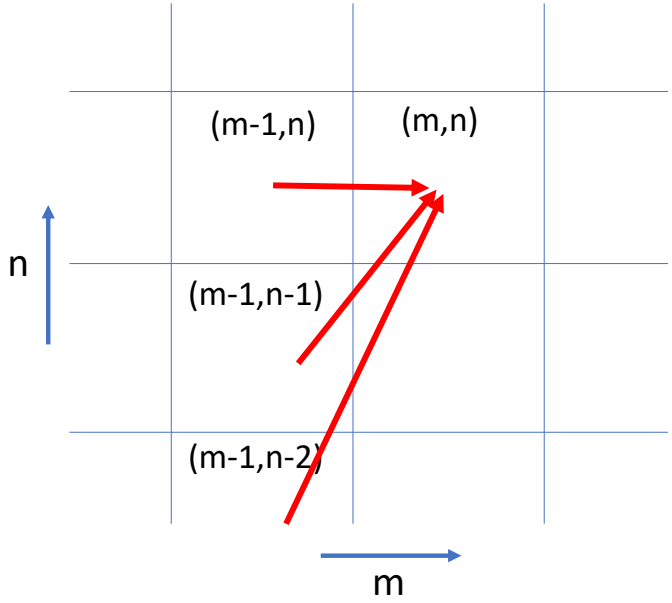
- Overly relies on knowledge
- Makes early decision so difficult to recover from errors such as, segmentation and labeling errors

Instance-based approach to match S an



- s_m and $w_{n,k}$ denote of frame of speech signal
- z_m and $y_{n,k}$ denote the corresponding feature vectors

Dynamic Time Warping (DTW)



local score $d(m, n)$:

- Cepstral features: Euclidean distance between z_m and $y_{n,k}$
- Linear prediction coefficients: Itakura distance between z_m and $y_{n,k}$
- Spectral information: Itakura-Saito distance between z_m and $y_{n,k}$

1. Initial condition: path starts at $(1, 1)$
2. Recursion:

$$\begin{aligned} D(m,n) =& d(m,n) + min[D(m-1,n), \\ & D(m-1,n-1), \\ & D(m-1,n-2)] \end{aligned}$$

$$\begin{aligned} \mathrm{Path}(m,n) =& \arg\min[D(m-1,n), \\ & D(m-1,n-1), \\ & D(m,n-2)] \end{aligned}$$

$\forall m \in \{1 \ldots M\}$ and $n \in \{1, \ldots N\}$

3. Final condition: path ends at (M, N) and $D(M, N)$ is the global score

$D(m, n)$ is referred to as cumulative score. $\mathrm{Path}(m, n)$ denotes the path index. Path can be traced back from $\mathrm{Path}(M, N)$

Four sub questions for instance-based approach

Q1 : Short-term spectral feature vectors are the latent symbols. The set of symbols is undefined, as there is no unique feature vector representation for speech sounds due to variabilities.

Q2 : Short-term speech processing-based feature extraction

Q3 : Short-term speech processing-based feature extraction

Q4 : Dynamic programming, i.e. DTW, with appropriate local score and local constraints

Outline

Recap

Statistical model-based approach

Training

Summary

Three Key Statistical Rules

1. Bayes's rule:

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A)$$

2. If B_k $(k = 1, \ldots, K)$ are mutually exclusive and collectively exhaustive $(\sum_{k=1}^{K} P(B_k) = 1)$

$$P(A) = \sum_{k=1}^{K} P(A, B_k)$$

3. Gibbs sampler:

$$P(B_1, \ldots, B_k, \ldots, B_K) = \prod_{k=1}^{K} P(B_k|B_{k-1}, \ldots, B_1)$$

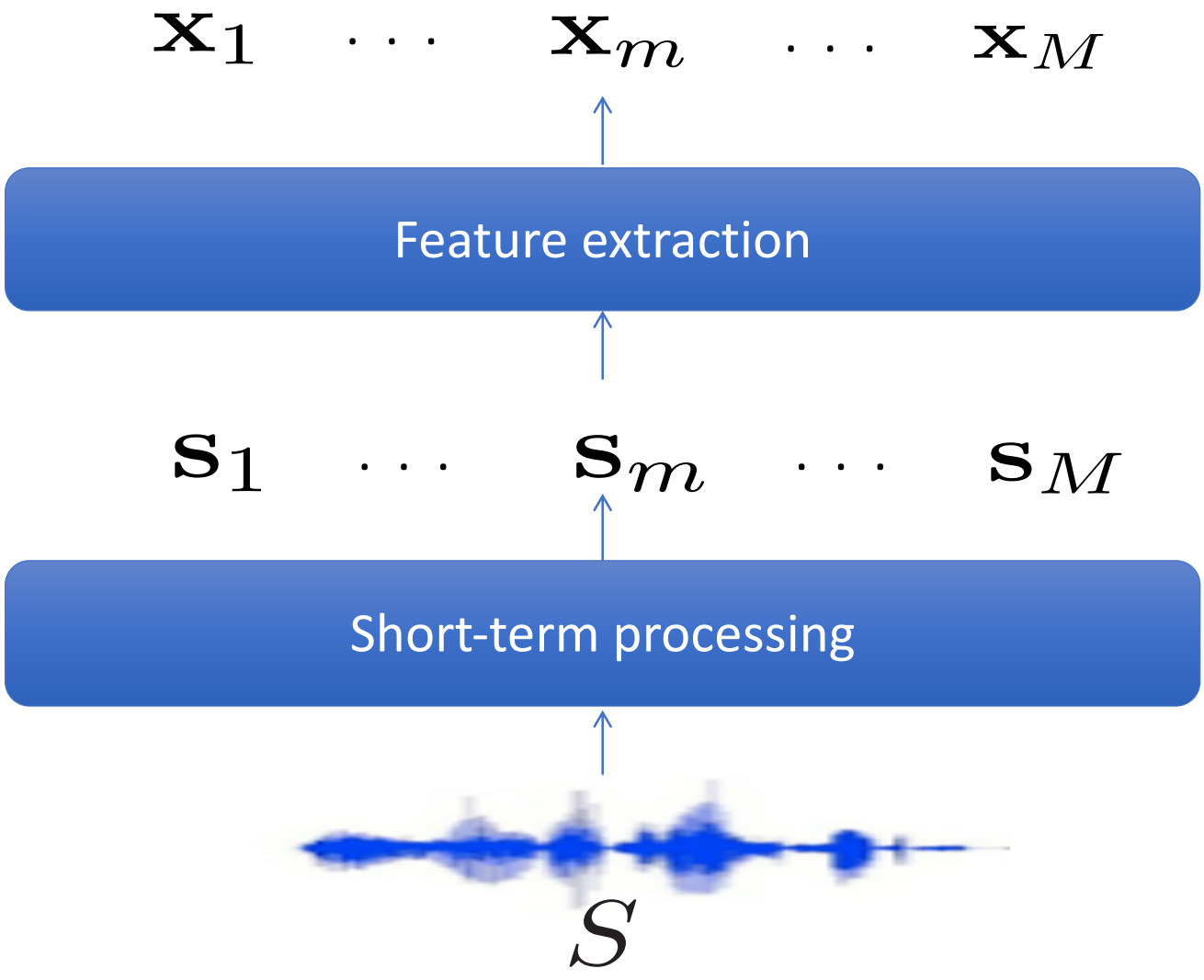
Statistical formulation for matching S and W_k

$$\hat{W} = \arg\max_{W_k \in \mathcal{W}} P(W_k|S) = \arg\max_{W_k \in \mathcal{W}} \frac{p(W_k, S)}{p(S)}$$

Statistical Automatic Speech Recognition

- Posterior-based approach: estimate $P(W_k|S)$
  Attempts through neural networks, e.g. Transition-based ASR, Listen, Attend and Spell
- Likelihood-based approach: estimate $p(W_k, S)$

Speech signal S representation



$$S \Rightarrow X = \{x_1, \cdots x_m, \cdots x_M\}$$

HMM-based ASR approach

$$p(W_k, X|\Theta_a, \Theta_l) = p(X|W_k, \Theta_a) \cdot P(W_k|\Theta_l)$$

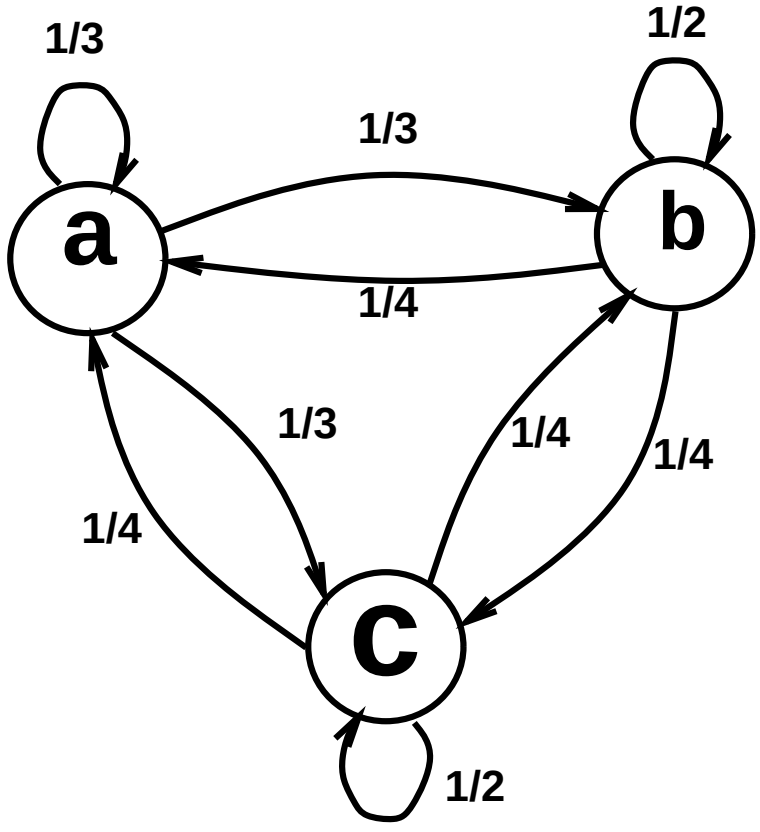
- $p(X|W_k, \Theta_a)$: acoustic likelihood estimated using hidden Markov models (HMMs)
- $P(W_k, \Theta_l)$: language model probability estimated using discrete Markov models (DMMs)

Three problems:

1. How to estimate $p(W_k, X)$ or simply $p(X|W_k, \Theta_a)$ and $P(W_k|\Theta_l)$?
2. How to estimate Θ_a and Θ_l? (Training)
3. How to find the most likely word hypothesis $\hat{W}$? (Recognition or decoding)

$$\hat{W} = \arg\max_{W_k \in \mathcal{W}} p(W_k, X|\Theta_a, \Theta_l)$$

Recall: discrete Markov model



Example of fully connected discrete Markov model with $\Omega = \{a, b, c\}$. For example, in case of weather model: "a" = "cloudy", "b"= "rainy" and "c" = "sunny"

Parameterized entirely by transition probabilities

Estimation of $P(W_k)$

Let $W_k = \{w_{k,1}, \ldots w_{k,j}, \ldots w_{k,J}\}$ (sequence of words)

$$\begin{aligned} P(W_k) &= P(w_{1,k}, \ldots w_{j,k}, \ldots w_{J,k}) \\ &= P(w_{J,k}|w_{J-1,k}, \ldots, w_{1,k}) \cdot P(w_{J-1,k}, \ldots w_{1,k}) \\ &= P(w_{J,k}|w_{J-1,k}, \ldots, w_{1,k}) \cdot P(w_{J-1,k}|w_{J-2,k}, \ldots, w_{1,k}) \cdots \\ &\quad P(w_{3,k}|w_{2,k}, w_{1,k}) \cdot P(w_{2,k}|w_{1,k}) \cdot P(w_{1,k}) \end{aligned}$$

Usually $P(w_{1,k}) = P(w_{1,k}|\text{initial state})$

Example: $W_k = \{\text{my}, \text{name}, \text{is}, \text{bond}\}$

$$\begin{aligned} P(\text{my}, \text{name}, \text{is}, \text{bond}) &= P(\text{bond}|\text{is}, \text{name}, \text{my}) \cdot P(\text{is}, \text{name}, \text{my}) \\ &= P(\text{bond}|\text{is}, \text{name}, \text{my}) \cdot P(\text{is}|\text{name}, \text{my}) \cdot \\ &\quad P(\text{name}|\text{my}) \cdot P(\text{my}) \end{aligned}$$

Estimation of $P(W_k)$: n-gram

- Challenge: variable history length
- Solution: Markov assumption
- n-gram is a $(n-1)$ order Markov model
  - bigram language model: first order Markov model

$$P(\text{my}, \text{name}, \text{is}, \text{bond}) = P(\text{bond}|\text{is}) \cdot P(\text{is}|\text{name}) \cdot P(\text{name}|\text{my}) \cdot P(\text{my}|\text{initial state})$$

  - trigram language model: second order Markov model

$$P(\text{my}, \text{name}, \text{is}, \text{bond}) = P(\text{bond}|\text{is}, \text{name}) \cdot P(\text{is}|\text{name}, \text{my}) \cdot P(\text{name}|\text{my}) \cdot P(\text{my}|\text{initial state})$$

Suppose the vocabulary size is O then the number of parameters or transition probabilities to estimate are:

- Bigram language model: $O \times O + O$
- Trigram language model: $O \times O \times O + O \times O + O$

n-gram parameter estimation

- Requires large amount of text e.g. books, web
- Parameter estimation through counting, e.g. trigram

$$P(w_n|w_{n-1}, w_{n-2}) = \frac{\text{Count}(w_n, w_{n-1}, w_{n-2})}{\sum_w \text{Count}(w, w_{n-1}, w_{n-2})}$$

Example:

$$P(\text{bond}|\text{is}, \text{name}) = \frac{\text{Count}(\text{bond}, \text{is}, \text{name})}{\sum_w \text{Count}(w, \text{is}, \text{name})}$$

- Small probability estimation problems, e.g. not enough examples, unseen word contexts
  - Interpolation with lower n-gram probabilities
  - Discounting: assign a small probability mass
  - Back-off to lower order n-gram probabilities
    $P(w_n|w_{n-1}, w_{n-2}) \Rightarrow P(w_n|w_{n-1})$
- Written text versus spoken language (conversational speech)

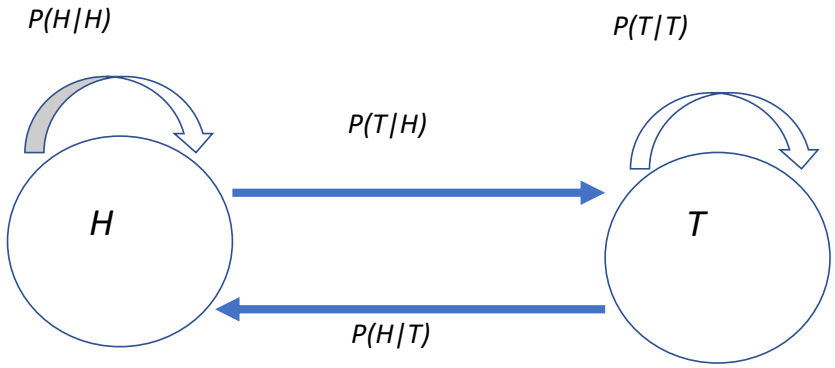
HMM-based ASR approach

$$p(W_k, X) = p(X|W_k, \Theta_a) \cdot P(W_k|\Theta_l)$$

- $p(X|W_k, \Theta_a)$: acoustic likelihood using hidden Markov models (HMMs)
- $P(W_k, \Theta_l)$: language model probability using discrete Markov models

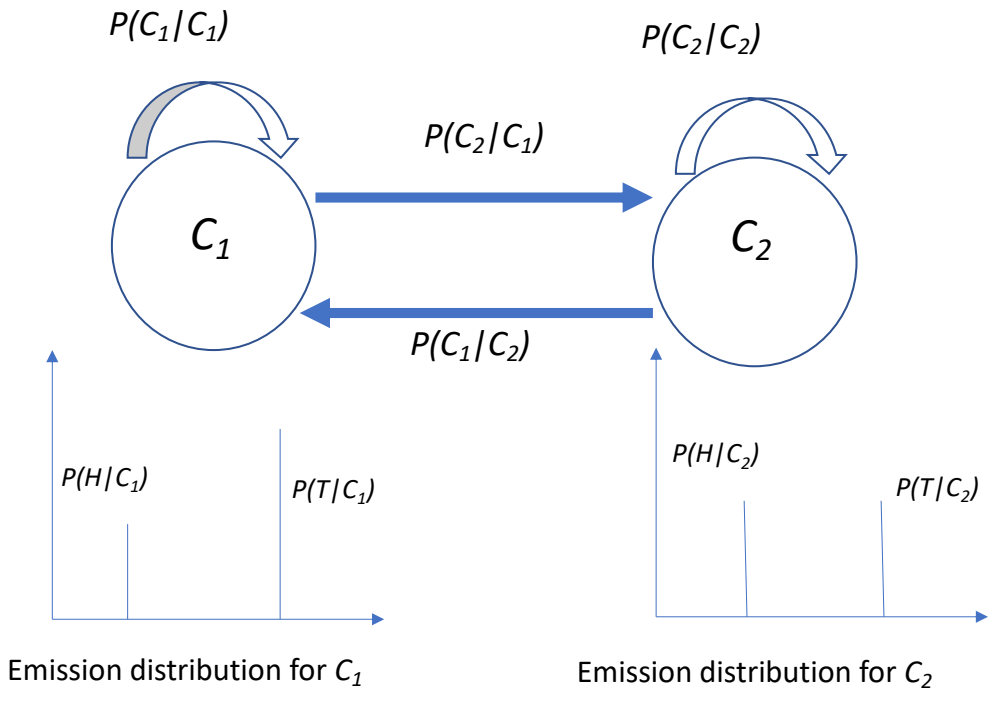
Recall: From DMM to HMM (1)

- Coin tossing and only the results of each coin toss i.e. Heads (H) or Tails (T) is revealed
  $H\ H\ T\ T\ T\ H\ H\ H\ H\ T\ T \ldots$
- 1-coin model: DMM with two states $\Omega = \{H, T\}$ and all transition probabilities are equal to 0.5.



From DMM to HMM (2)

- 2-coin model: HMM with two states $\Omega = \{C_1, C_2\}$; each state associated with $P(H|C_i)$ and $P(T|C_i)$ $i \in \{1, 2\}$; and the transition probability reflects the probability of choosing one of the (possibly biased) coins



From X we can not directly know about W_k.

Estimation of $P(X|W_k, \Theta_a)$ using HMMs (1)

$$
\begin{aligned}
P(X|W_k, \Theta_a) &= \sum_{Q \in W_i} P(X, Q|W_k, \Theta_a) \\
&= \sum_{Q \in W_k} P(X|Q, W_k) P(Q|W_k),
\end{aligned}
$$

where $Q = \{q_1, \cdots q_m, \cdots q_M\}$ denotes sequence of HMM states.
After i.i.d and first order Markov assumption

$$
\begin{aligned}
P(X|W_k) &= \sum_{Q \in W_k} \prod_{m=1}^{M} p(\mathrm{x}_m|q_m) \cdot \prod_{m=1}^{M} P(q_m|q_{m-1}) \\
&= \sum_{Q \in W_k} \prod_{m=1}^{M} p(\mathrm{x}_m|q_m) \cdot P(q_m|q_{m-1}) \text{ Full likelihood} \\
&\approx \max_{Q \in W_i} \prod_{m=1}^{M} p(\mathrm{x}_m|q_m) \cdot P(q_m|q_{m-1}) \text{ Viterbi approx.}
\end{aligned}
$$

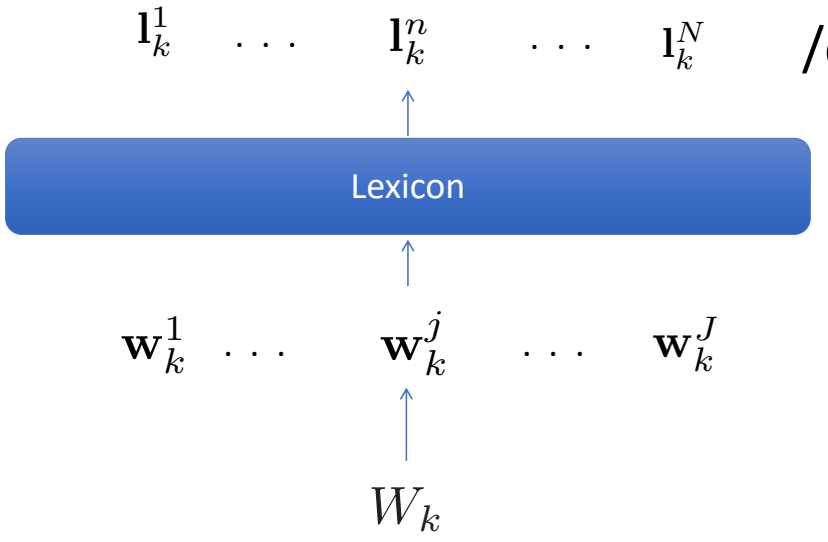
Estimation of $P(X|W_k, \Theta_a)$ using HMMs (2)

$$
\begin{aligned}
P(X|W_k) &\approx \max_{Q \in W_k} \prod_{m=1}^{M} p(x_m|q_m) \cdot P(q_m|q_{m-1}) \\
&\approx \max_{Q \in W_k} \prod_{m=1}^{M} \left(\sum_{d=1}^{D} p(x_m, a^d|q_m)\right) \cdot P(q_m|q_{m-1}) \\
&\approx \max_{Q \in W_k} \prod_{m=1}^{M} \left(\sum_{d=1}^{D} p(x_m, |a^d, q_m) \cdot P(a^d|q_m)\right) \cdot P(q_m|q_{m-1}) \\
&\approx \max_{Q \in W_k} \prod_{m=1}^{M} \left(\sum_{d=1}^{D} p(x_m, |a^d) \cdot P(a^d|q_m)\right) \cdot P(q_m|q_{m-1})
\end{aligned}
$$

Assumption x_m is independent of q_m given a^d.
$\{a^d\}_{d=1}^{D}$ are the latent symbols:

- context-independent phones
  CAT: /k/ /ae/ /t/
- "clustered" context-dependent phones
  CAT: /k/+/ae/ /k/-/ae/+/t/ /ae/-/t/

Modeling of $P(a^d|q_m)$



Let us denote $\mathrm{y}_{n,k} = [P(a^1|q_m = l_k^n) \cdots P(a^D|q_m = l_k^n)]^{\mathrm{T}}$

- Modeling of $\mathrm{y}_{n,k}$ is based on
  - prior knowledge: $\{a^d\}_{d=1}^D$ are context-independent phones
  - both prior knowledge and data: $\{a^d\}_{d=1}^D$ are clustered context-dependent states
- Typically, one-to-one mapping between l_k^n and a^d, i.e. $\mathrm{y}_{n,k}$ is a Kronecker delta distribution.

Modeling of $p(\mathrm{x}_m|a^d)$

- Gaussians and Gaussian-Mixtures

$$p(\mathrm{x}_m|a^d) = N(\mathrm{x}_m, \mu_{a^d}, \Sigma_{a^d})$$

$$= \frac{1}{(2\pi)^{R/2}|\Sigma_{a^d}|^{1/2}} \exp\left(-\frac{(\mathrm{x}_m - \mu_{a^d})^t \Sigma_{a^d}^{-1}(x_m - \mu_{a^d})}{2}\right)$$

$$\mathrm{OR}:$$

$$p(\mathrm{x}_m|a^d) = \sum_{j=1}^{J} c_{a^d}^j N(\mathrm{x}_m, \mu_{a^d}^j, \Sigma_{a^d}^j)$$

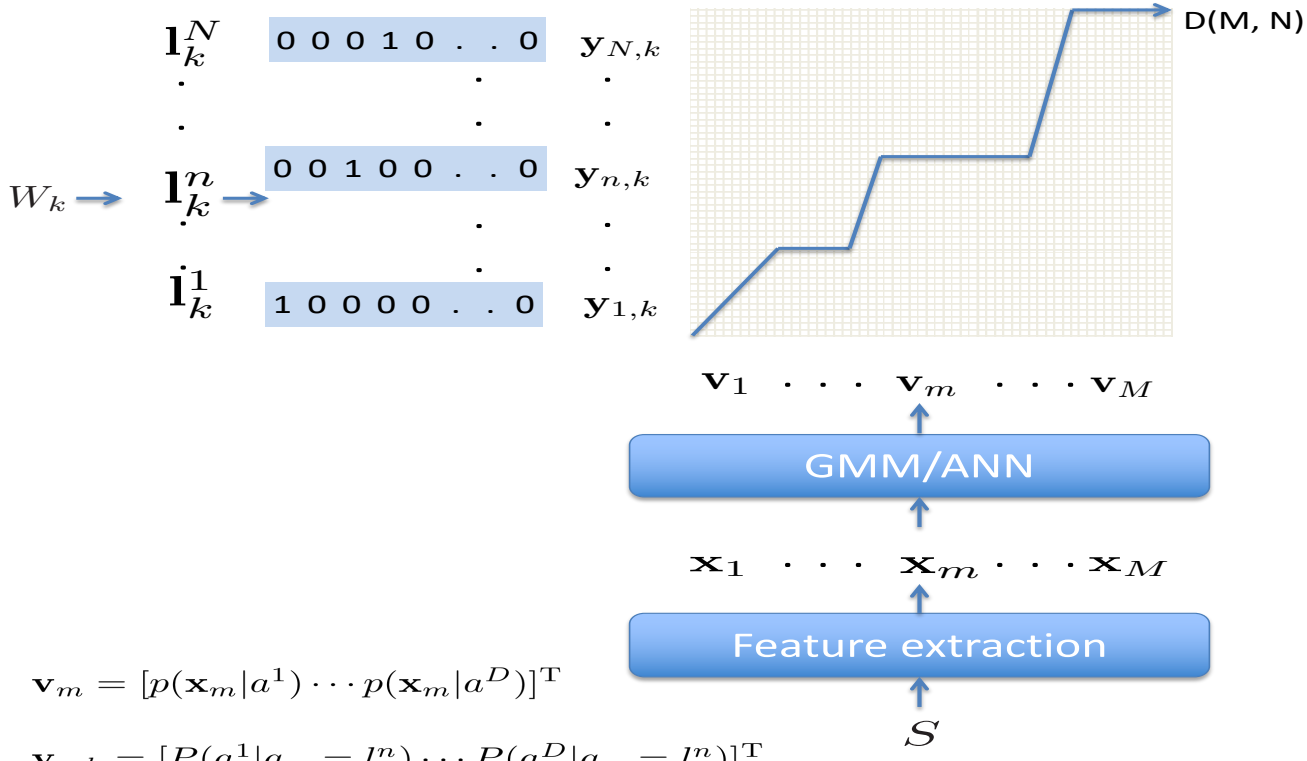
R is the dimension of the feature vector.

- Artificial neural networks: estimate scaled-likelihood $p_{sl}(\mathrm{x}_m|a^d)$

$$p_{sl}(\mathrm{x}_m|a^d) = \frac{p(\mathrm{x}_m|a^d)}{p(\mathrm{x}_m)} = \frac{P(a^d|\mathrm{x}_m)}{P(a^d)}$$

Let us denote $\mathrm{v}_m = [p(\mathrm{x}_m|a^1) \cdots p(\mathrm{x}_m|a^D)]^{\mathrm{T}}$ or
$\mathrm{v}_m = [p_{sl}(\mathrm{x}_m|a^1) \cdots p_{sl}(\mathrm{x}_m|a^D)]^{\mathrm{T}}$

Matching X and W_k: $P(W_k, X)$



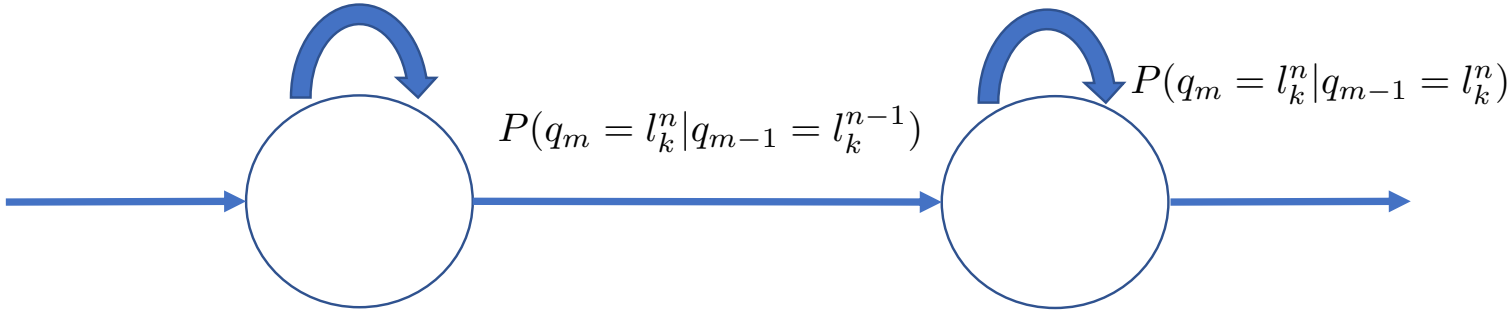
$$\mathbf{v}_m = [p(\mathbf{x}_m|a^1) \cdots p(\mathbf{x}_m|a^D)]^{\mathrm{T}}$$

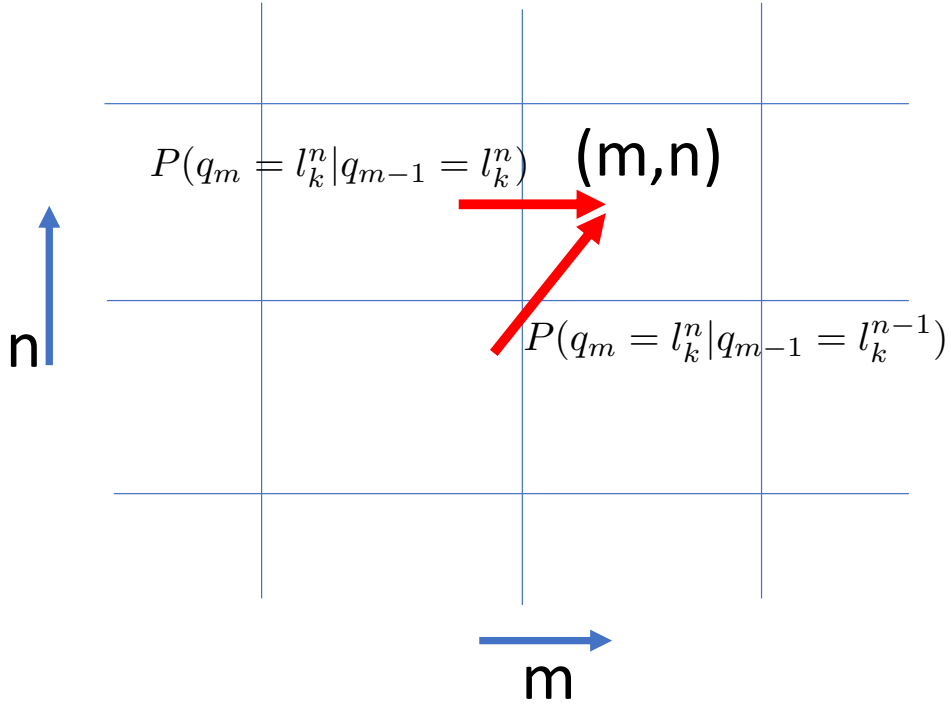
$$\mathbf{y}_{n,k} = [P(a^1|q_m = l_k^n) \cdots P(a^D|q_m = l_k^n)]^{\mathrm{T}}$$

$$\log(P(X|W_k)) \approx \max_{Q \in W_k} \sum_{m=1}^{M} \log\Big(\sum_{d=1}^{D} p(\mathrm{x}_m, |a^d) \cdot P(a^d|q_m = l_k^n)\Big) + \log(P(q_m = l_k^n|q_{m-1}))$$

$$\approx \max_{Q \in W_k} \sum_{m=1}^{M} \log(\mathrm{v}_m^{\mathrm{T}} \mathrm{y}_{n,k}) + \log(P(q_m = l_k^n|q_{m-1}))$$

Dynamic programming (1)





Dynamic programming (2)

Local score: $d(m,n) = -\log(\mathrm{v}_m^{\mathrm{T}}\mathrm{y}_{n,k})$

1. Initial condition: path starts at $(1,1)$
2. Recursion:

$$D(m,n) = d(m,n) + min[D(m-1,n) - \log(P(q_m = l_k^n | q_{m-1} = l_k^n),$$
$$D(m-1,n-1) - \log(P(q_m = l_k^n | q_{m-1} = l_k^{n-1})]$$

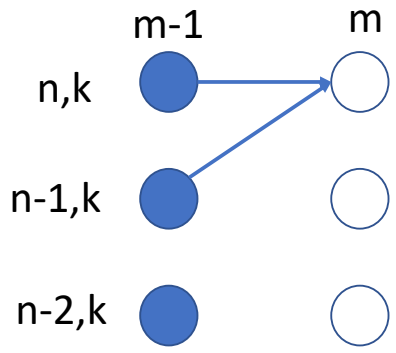
$$\mathrm{Path}(m,n) = \arg\min[D(m-1,n) - \log(P(q_m = l_k^n | q_{m-1} = l_k^n),$$
$$D(m-1,n-1) - \log(P(q_m = l_k^n | q_{m-1} = l_k^{n-1})]$$

$\forall m \in \{1 \ldots M\}$ and $n \in \{1, \ldots N\}$

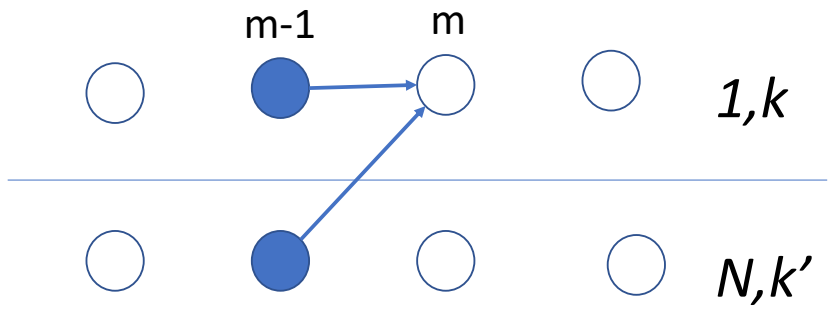
3. Final condition: path ends at (M,N) and $D(M,N)$ is the global score

$\mathrm{Path}(m,n)$ denotes the path index. Path can be traced back from $\mathrm{Path}(M,N)$

Across word local constraint and LM



Within word constraint



Across word constraint

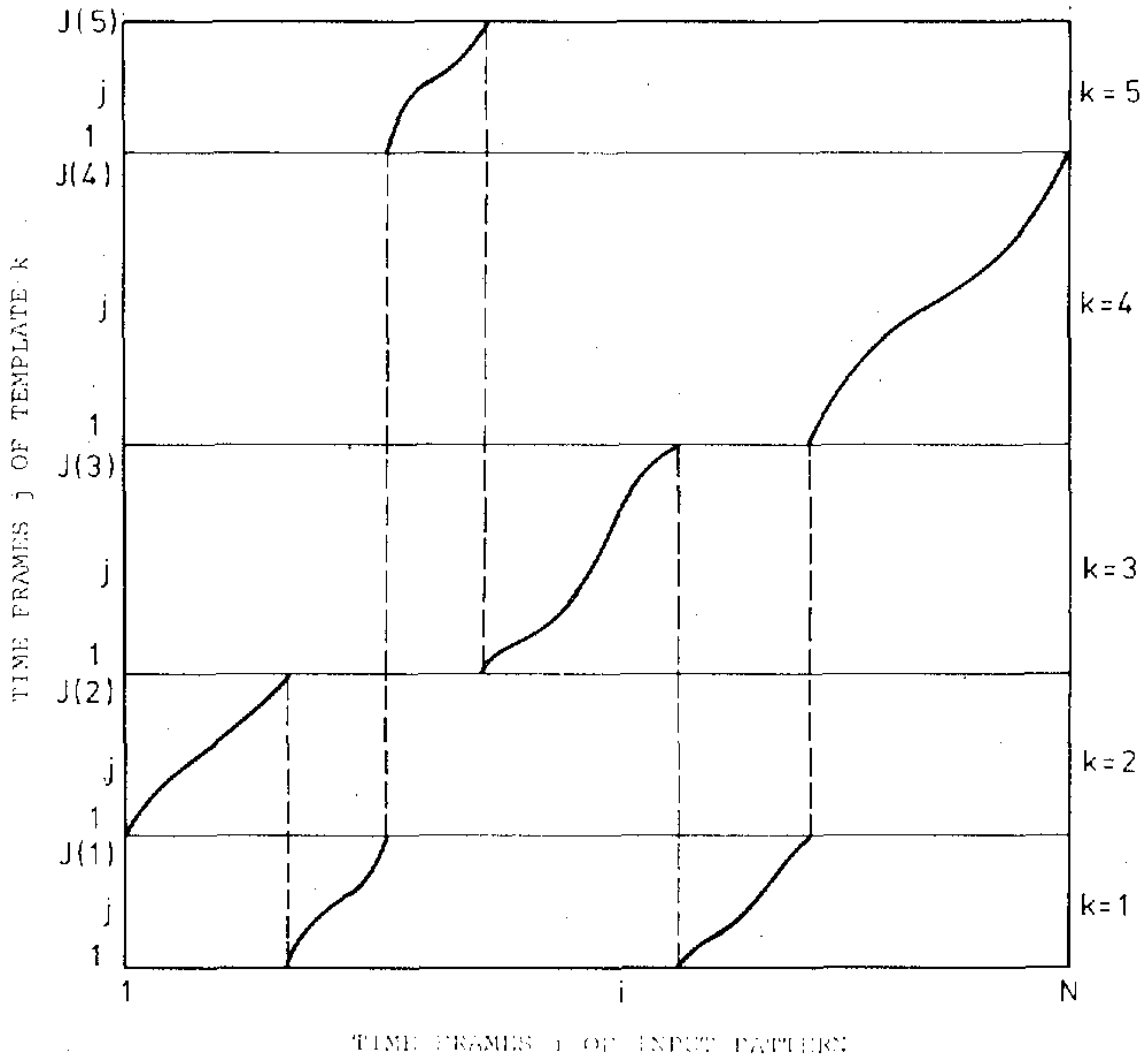
$\forall k^{'} \in \{1, \cdots K\}$ the transition between w_k and $w_k^{'}$ has a cost based on the language model, e.g. $P(w_k|w_k^{'})$ (assuming bigram language model)

$$\log(P(X|W_k) \cdot P(W_k)) \approx \max_{Q \in W_k} \sum_{m=1}^{M} \log(\sum_{d=1}^{D} p(x_m, |a^d) \cdot P(a^d|q_m = l_k^n)) + \log(P(q_m = l_k^n|q_{m-1}))$$

$$\approx \max_{Q \in W_k} \sum_{m=1}^{M} \log(v_m^{\mathrm{T}} y_{n,k}) + \log(P(q_m = l_k^n|q_{m-1}))$$

Transition between words: $P(q_m = l_k^n|q_{m-1})$ is estimated from the language model.

Continuous speech recognition



Source: H. Ney,"The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition ", IEEE Trans. on Acoustics, Speech, and Signal Processing, 32(2), 1984.

Four sub questions for HMM-based approach

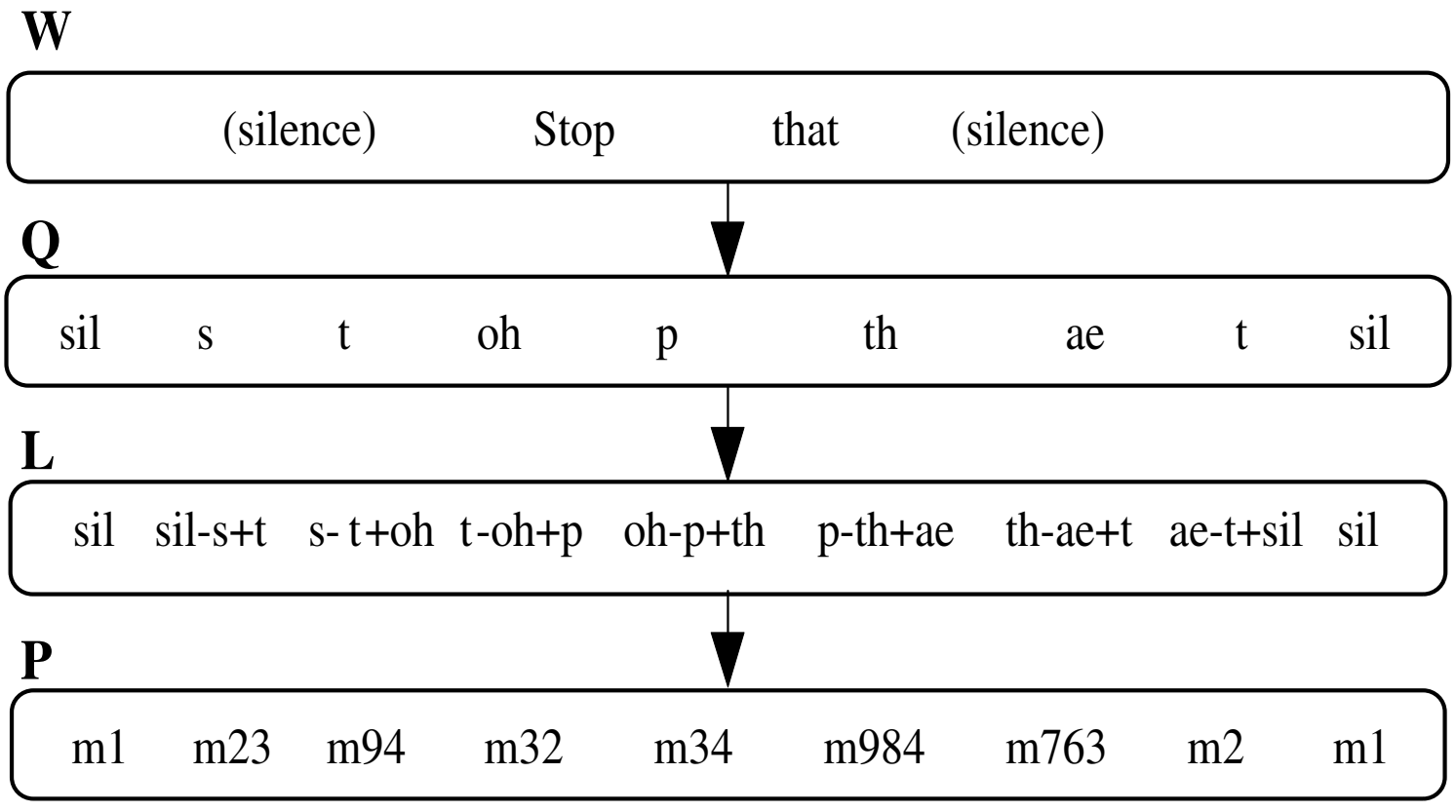
Q1 : Latent symbol set $\{a^d\}_{d=1}^{D}$: typically clustered context-dependent phones

Q2 : Estimation of emission likelihood vector v_m per frame using GMMs or ANNs

Q3 : Estimation of $\mathrm{y}_{n,k}$ Typically, one-to-one mapping between context-dependent phone unit I_k^n and a^d (i.e., Kronecker delta distribution) based on the state tying decision tree

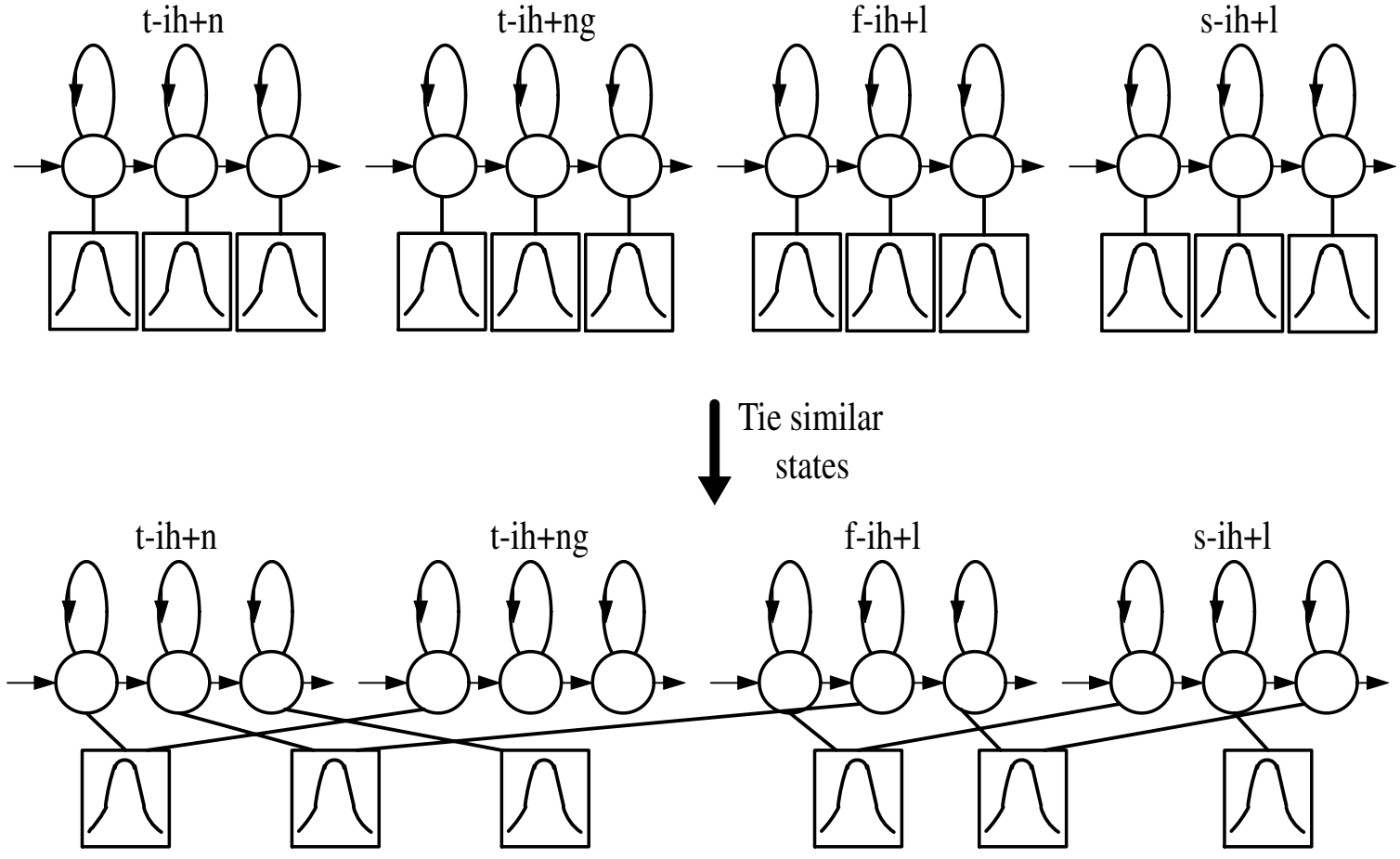
Q4 : Dynamic programming with local score $-\log(\mathrm{v}_m^{\mathrm{T}}\mathrm{y}_{n,k})$ and local constraints are based on the HMM topology and transitions can have unequal cost.

Context-dependent phone modeling (1)



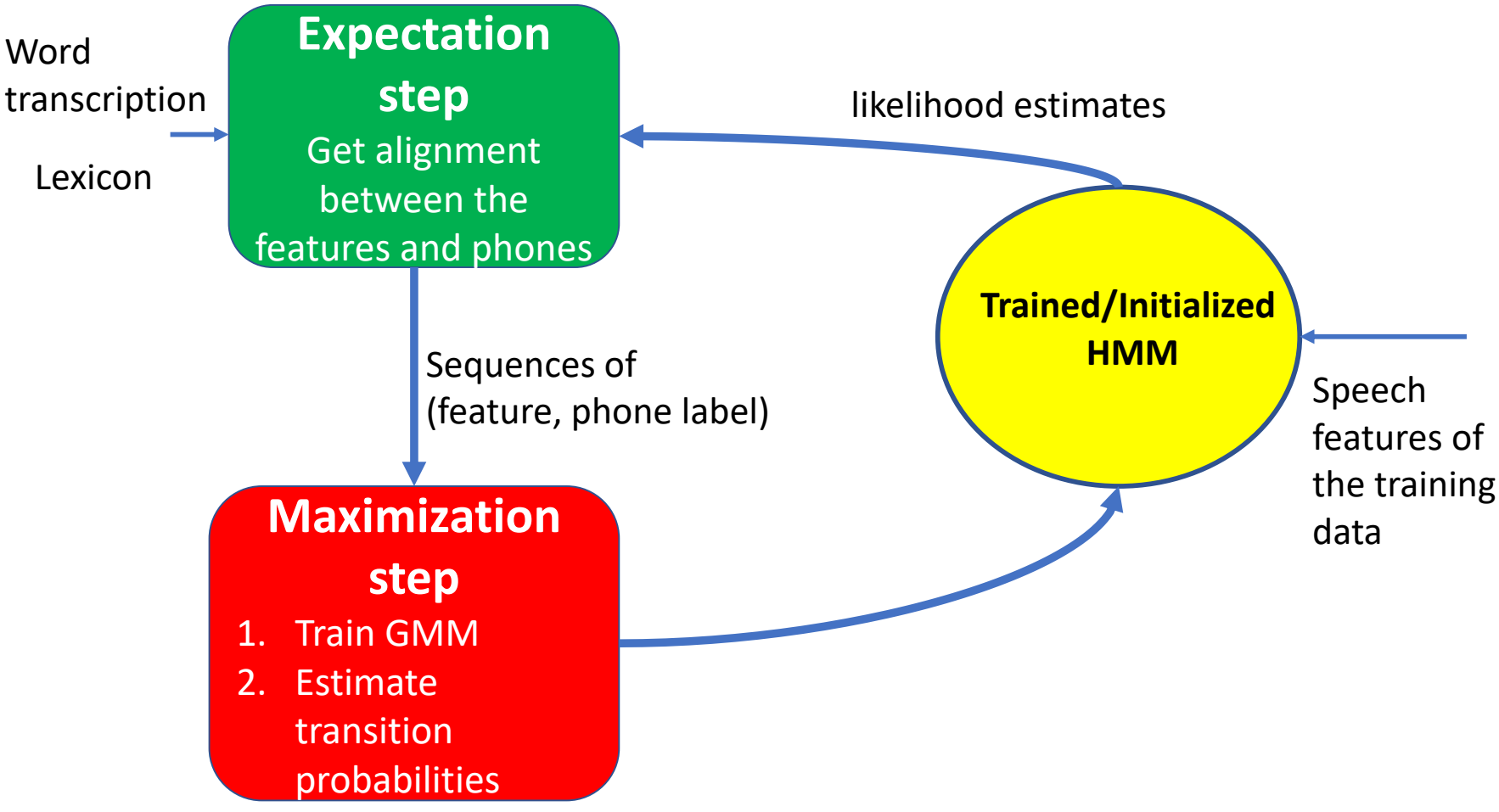
(Source: M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends inSignal Processing,Vol. 1, No. 3 (2007) 195–304)

Context-dependent phone modeling (2)

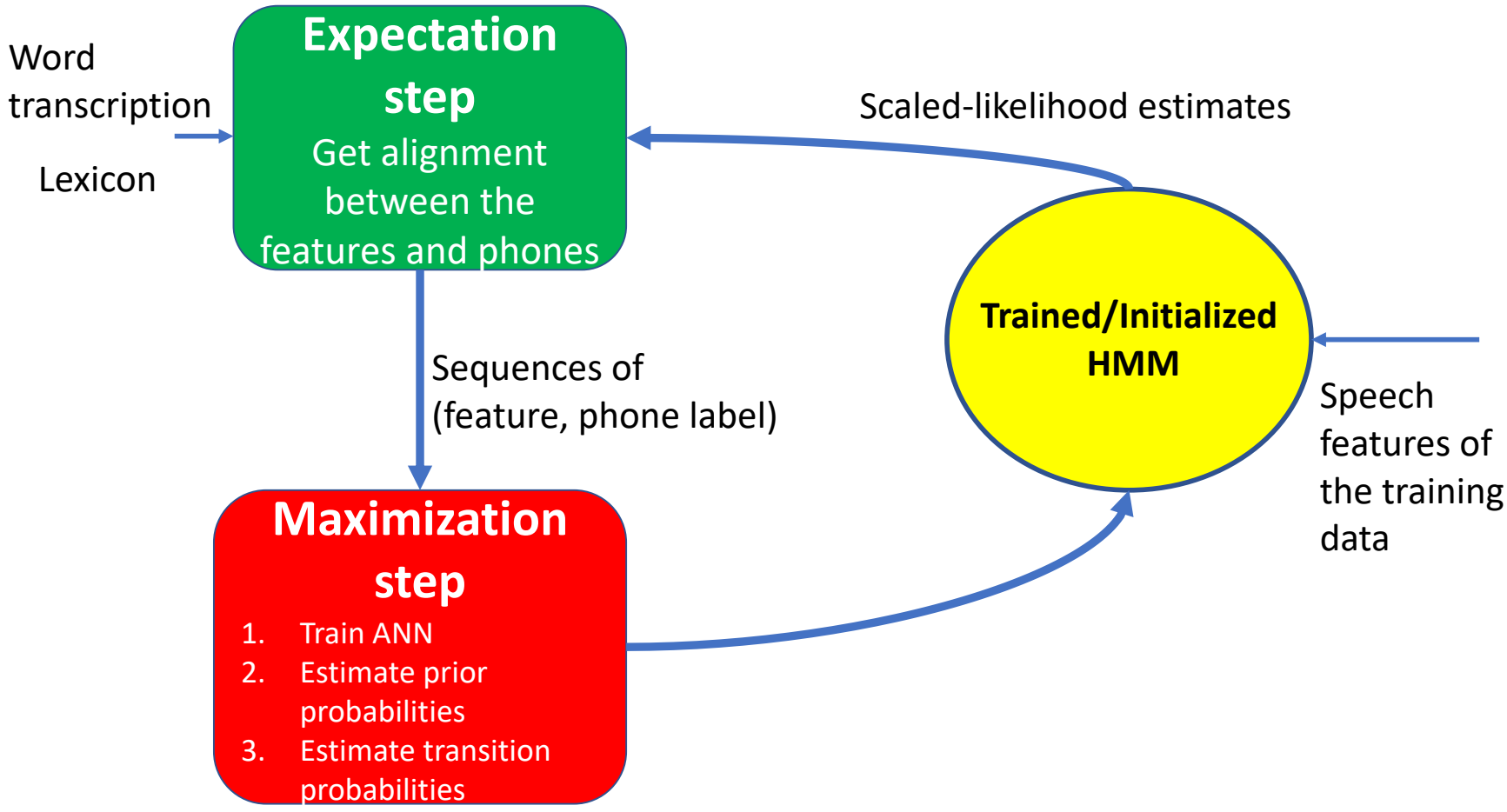


(Source: M. Gales and S. Young, "The Application of Hidden Markov Models in Speech Recognition", Foundations and Trends inSignal Processing,Vol. 1, No. 3 (2007) 195–304)

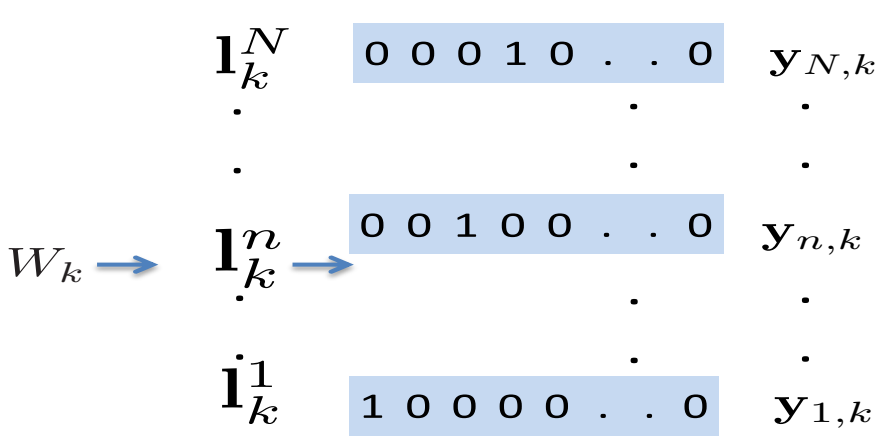
HMM/GMM training

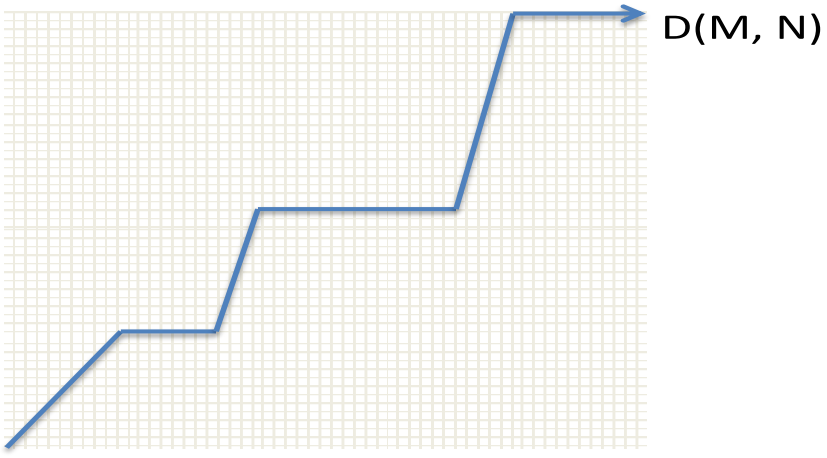


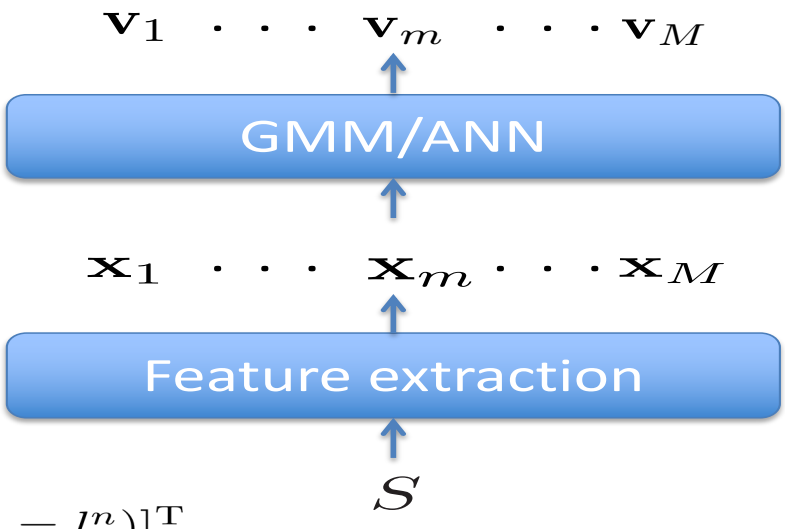
Hybrid HMM/ANN training



Expectation step







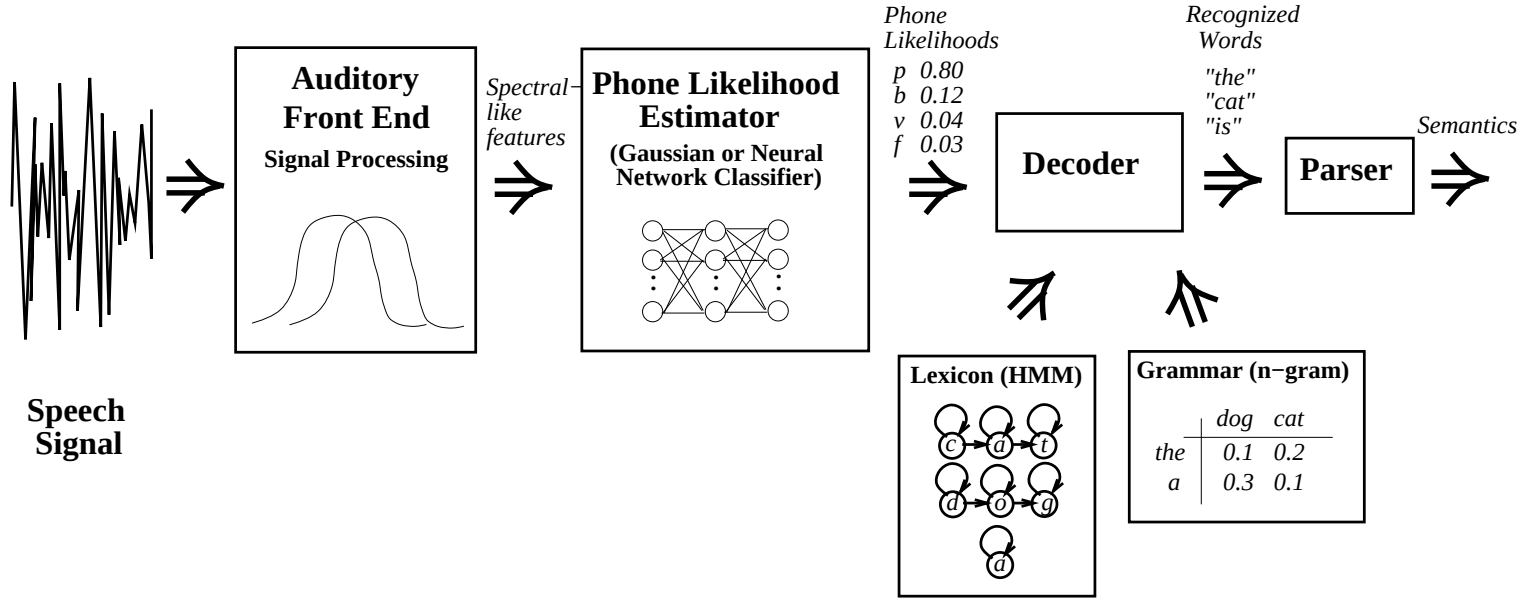
$$\mathbf{v}_m = [p(\mathbf{x}_m|a^1) \cdots p(\mathbf{x}_m|a^D)]^{\mathrm{T}}$$

$$\mathbf{y}_{n,k} = [P(a^1|q_m = l_k^n) \cdots P(a^D|q_m = l_k^n)]^{\mathrm{T}}$$

Resources needed

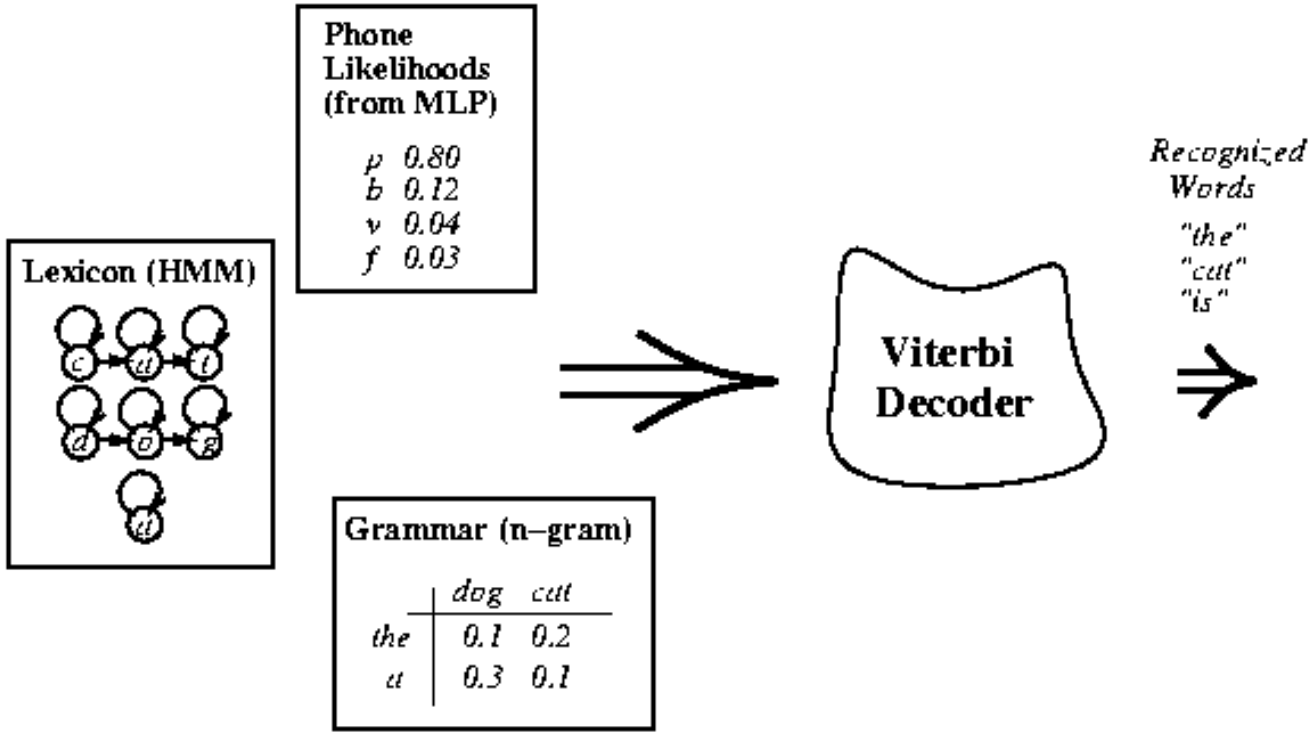
- Large set of speech signals with word level transcriptions
- Phonetic lexicon that transcribes each word as a sequence of phones. Alternative is to use graphemic representations (orthographic transcription of word)
  - Fine for languages with shallow grapheme-to-phoneme relation (e.g., Spanish, Finnish)
  - Not-so-good for languages with deep grapheme-to-phoneme relation (e.g., English)
- Large text resources for language model

HMM-based ASR system



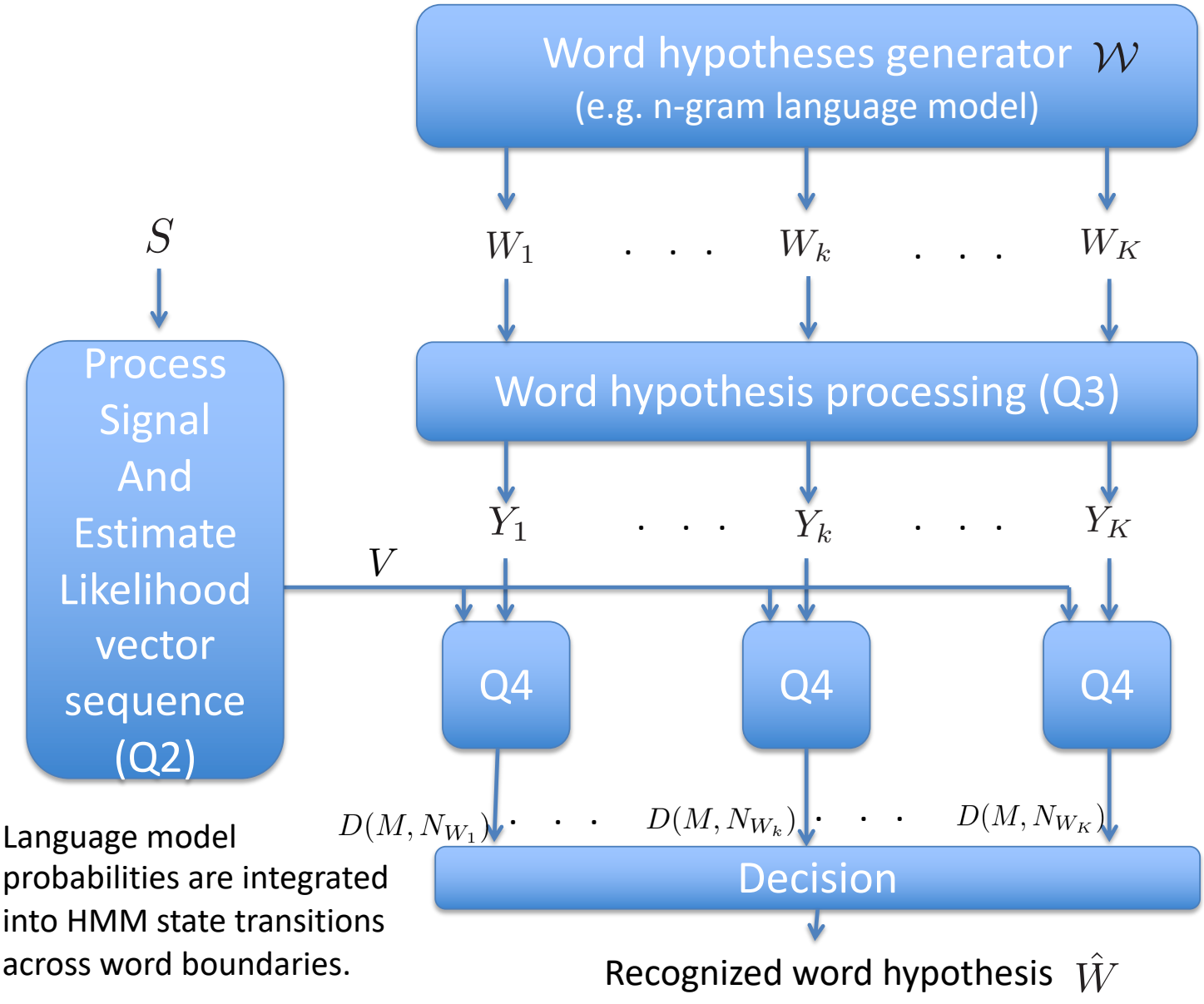
- Prior knowledge
  - Syntactical or grammatical constraints using DMMs, i.e. language model
  - Lexical constraints using DMMs
    BAT: /b/ $\rightarrow$ /ae/ $\rightarrow$ /t/
- Data-driven acoustic evidence
  - Phone/subword units HMMs

Decoder



- Implementation of decoder typically done using weighted finite state transducers (composition of a big graph and searching through it)
- Full breadth search practically infeasible
- Need for search heuristics e.g., Beam search
- Other hyper-parameters: acoustic scaling factor, language scaling factor, word insertion penalty (to avoid insertion of short words)

Interpretation



Language model probabilities are integrated into HMM state transitions across word boundaries.

Outline

Recap

Statistical model-based approach

Training

Summary

Data preparation

Resources needed: Speech utterances and their word level transcriptions, Phonetic dictionary (also called as lexicon)

Let $\{S_j, H_j\}_{j=1}^{J}$ denote a set of speech utterances and their corresponding word level transcriptions.

1. Extract the acoustic feature sequence $X_j = (\mathrm{x}_1, \cdots \mathrm{x}_m, \cdots \mathrm{x}_{M_j})$ corresponding to each speech utterance S_j. x_m is typically 39 dimensional cepstral feature vector ($C_0 - C_{12}$ and their approximate first and second temporal derivatives).
2. For each corresponding transcription H_j create a left-to-right HMM model W_j by using the phonetic dictionary. For example, see creation of HMM for "AND", "IT" and "AND IT".

Goal of training

Infer the HMM parameters, i.e., emission distribution parameters and the transition probabilities for each HMM state such that it maximizes

$$\prod_{j=1}^{J} p(X_j|W_j)$$

Emission distribution parameters (besides the lexical model parameter y):

1. single multivariate Gaussian: mean vector and covariance matrix for each latent symbol a^d
2. GMMs: mixture weights, mean vectors and covariance matrices of the GMM for each a^d
3. ANNs: weights and biases and prior probability of each a^d, i.e. $P(a^d)$.

Direct optimization of the above objective function is not possible.

Parameters are iteratively estimated using Expectation-Maximization (EM) algorithm.

Case 1: Single Gaussian

Step 1: For each latent symbol a^d, randomly initiliaze the mean vector and covariance matrix of the multivariate Gaussian and the transition probabilities.

Step 2: Given the parameters, $\forall j \in \{1, \cdots J\}$, estimate $\log(P(X_j|W_j))$ by dynamic programming and obtain the alignment between the feature sequence X_j and the state sequence in W_j by backtracking using $Pa(m, n)$.

Step 3: In the alignment, map the states to a^d based on the one-to-one mapping in the lexical model parameter $y_{n,j}$. For each a^d, collect all the feature vectors x_m that belong to that latent symbol and estimate the new mean vector and covariance matrix.
From the aligned sequence of states, estimate the self transition probabilities for each state by counting.

Step 4: Go to Step 2.

Repeat Step 2 (E-step) and Step 3 (M-step) until convergence.

Case 2: GMMs

Step 1: For each acoustic unit a^d, randomly initiliaze the GMM parameters, i.e. mixture weights, mean vector and covariance matrix each multivariate Gaussian, and the transition probabilities.

Step 2: Given the parameters, $\forall j \in \{1, \cdots J\}$, estimate $\log(P(X_j|W_j))$ by dynamic programming and obtain the alignment between the feature sequence X_j and the state sequence in W_j by backtracking using $Pa(m, n)$.

Step 3: In the alignment, map the states to a^d based on the one-to-one mapping in the lexical model parameter $y_{n,j}$. For each a^d, collect all the feature vectors x_m that belong to that latent symbol and estimate the new GMM parameters. From the aligned sequence of states, estimate the self transition probabilities for each state by counting.

Step 4: Go to Step 2.

Repeat Step 2 (E-step) and Step 3 (M-step) until convergence.

Case 3: ANNs

Step 1: Initialize the weights and biases of the ANN that takes as input x_m and classifies the acoustic units $\{a^d\}_{d=1}^{D}$. Randomly initialize the transition probability of states. Assume equal prior probability for the acoustic units.

Step 2: Given the parameters, $\forall j \in \{1, \cdots J\}$, estimate $\log(P(X_j|W_j))$ by dynamic programming and obtain the alignment between the feature sequence X_j and the state sequence in W_j by backtracking using $Pa(m, n)$.

Step 3: In the alignment, map the states to a^d based on the one-to-one mapping in the lexical model parameter $y_{n,j}$. Train a new ANN classifier that classifies the latent symbols $\{a^d\}_{d=1}^{D}$ using cross entropy or mean square error criterion given x_m as input.

From the alignment, estimate the prior probability for each a^d

From the aligned sequence of states, estimate the self transition probabilities for each state by counting.

Step 4: Go to Step 2.

Repeat Step 2 (E-step) and Step 3 (M-step) until convergence.

Training in this fashion is quite expensive. In practice, as part of E-step, a GMM-based system is trained to obtain a good alignment between X_j and W_j $\forall j$, and the M-step is carried out once.

Outline

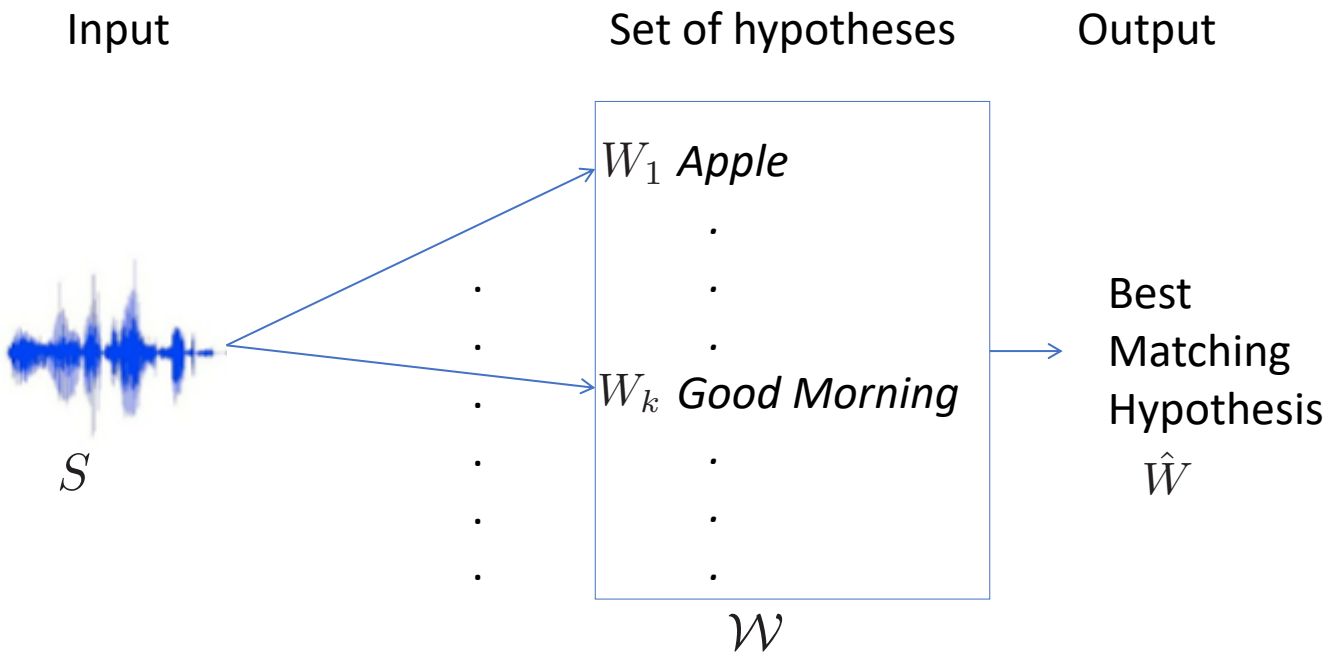
Recap

Statistical model-based approach

Training

Summary

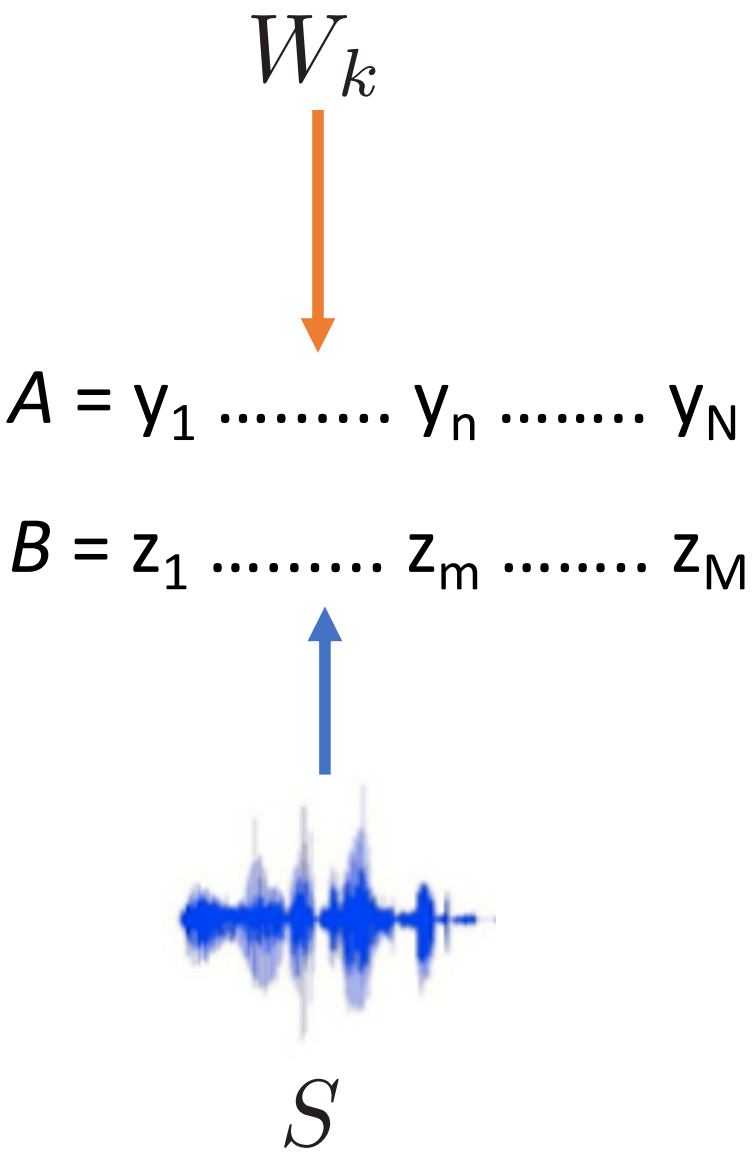
Automatic speech recognition (ASR)



$$\hat{W} = \underset{W_k \in \mathcal{W}}{\arg\max}\, \mathrm{Match}(W_k, S)$$

How to match an observed speech signal S with a word hypothesis W_k?

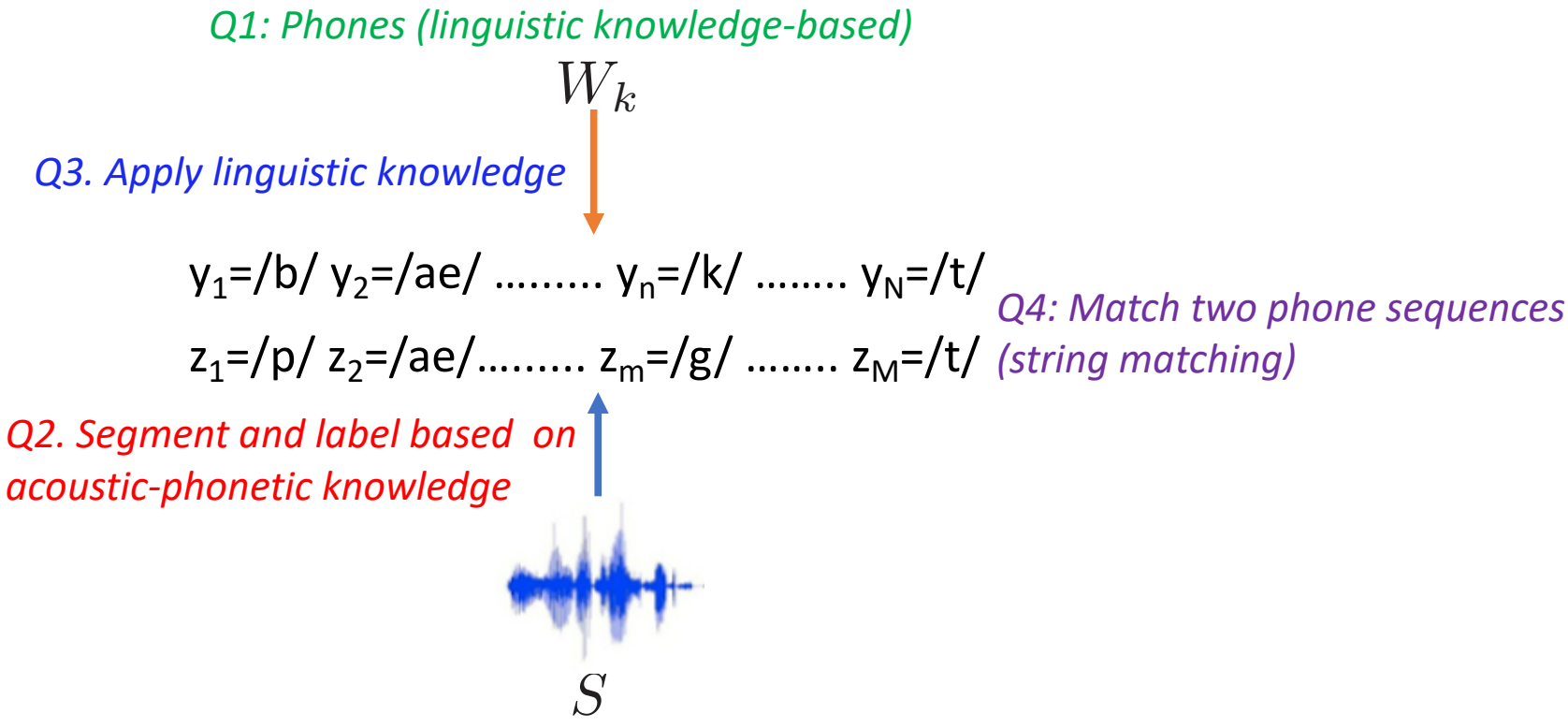
Abstract formulation for matching S and W_k



Core Idea

1. Map S and W_k to a shared latent symbol space
2. Match the resulting two latent symbol sequences A and B

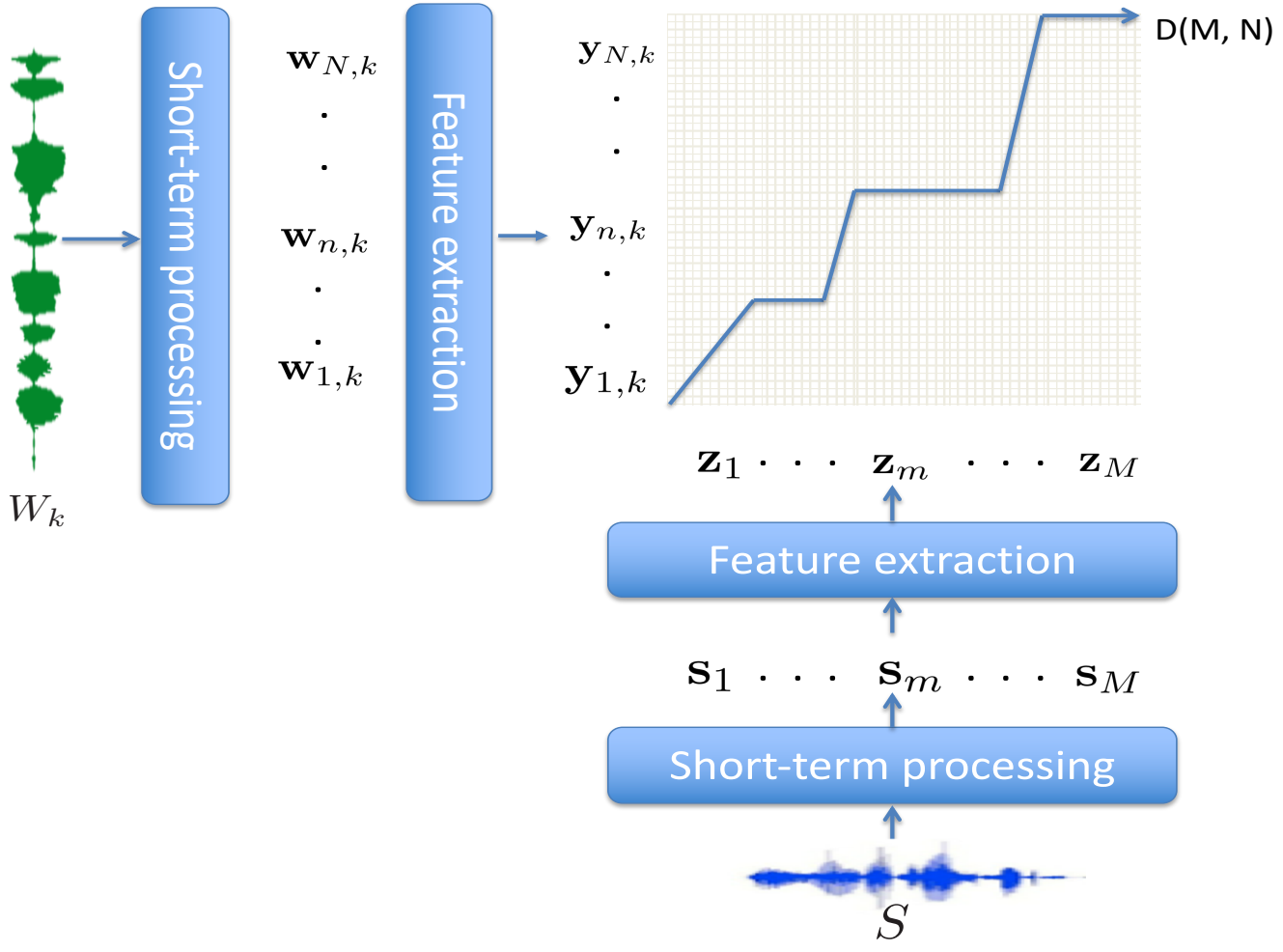
Knowledge-based ASR approach



Limitations:

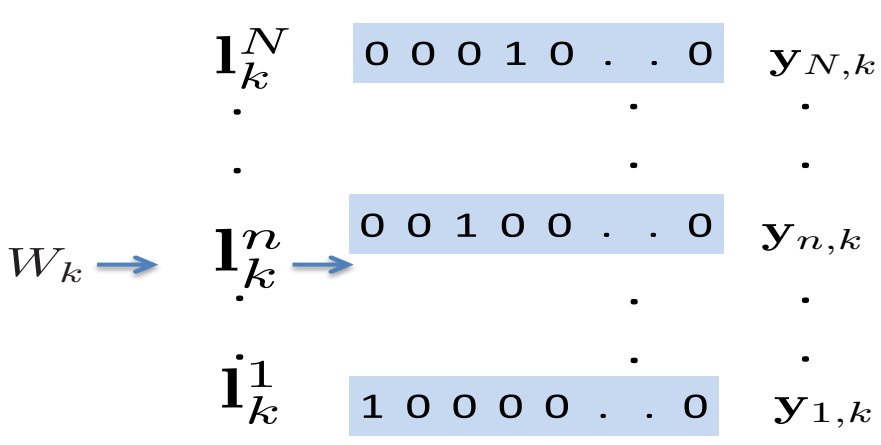
- Overly relies on knowledge
- Makes early decision so difficult to recover from errors such as, segmentation and labeling errors

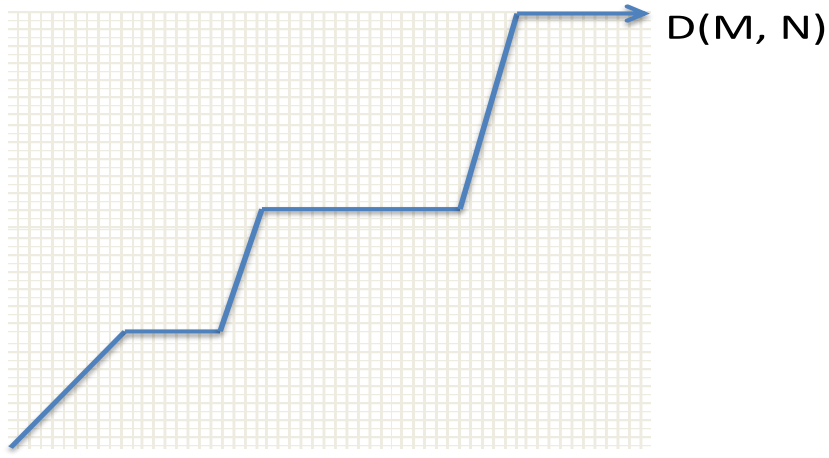
Instance-based approach to match S an

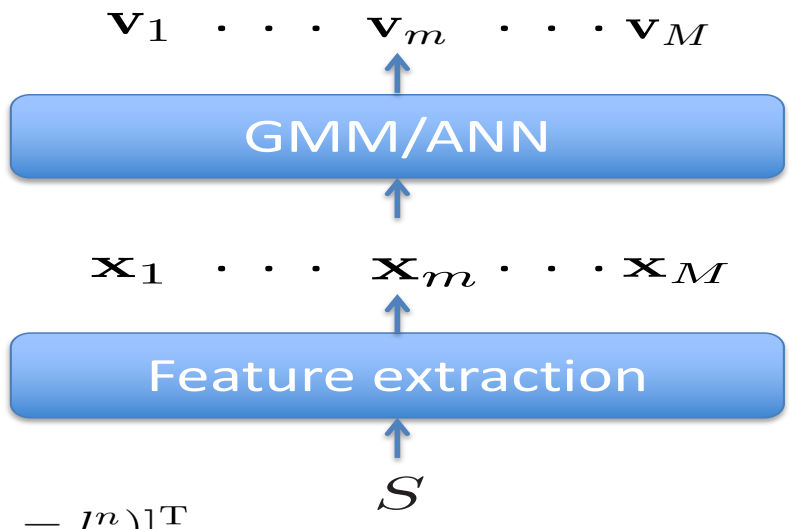


- s_m and $w_{n,k}$ denote of frame of speech signal
- z_m and $y_{n,k}$ denote the corresponding feature vectors

Matching S and W_k: $P(W_k, S)$







$$\mathbf{v}_m = [p(\mathbf{x}_m|a^1) \cdots p(\mathbf{x}_m|a^D)]^{\mathrm{T}}$$

$$\mathbf{y}_{n,k} = [P(a^1|q_m = l_k^n) \cdots P(a^D|q_m = l_k^n)]^{\mathrm{T}}$$

Thank you for your attention!

Dr. Mathew Magimai Doss

Idiap Research Institute, Martigny, Switzerland