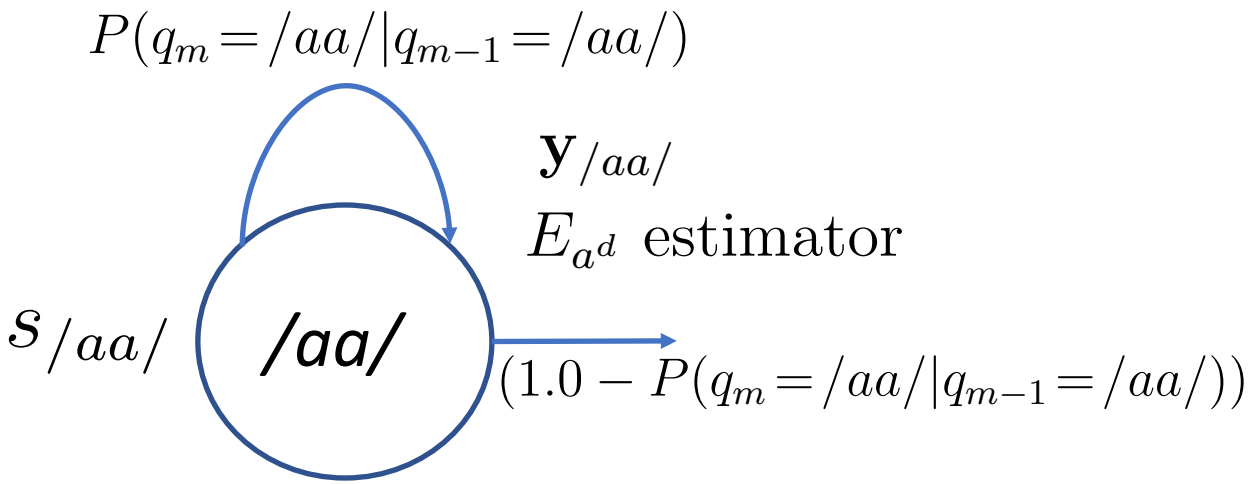
HMM-based ASR

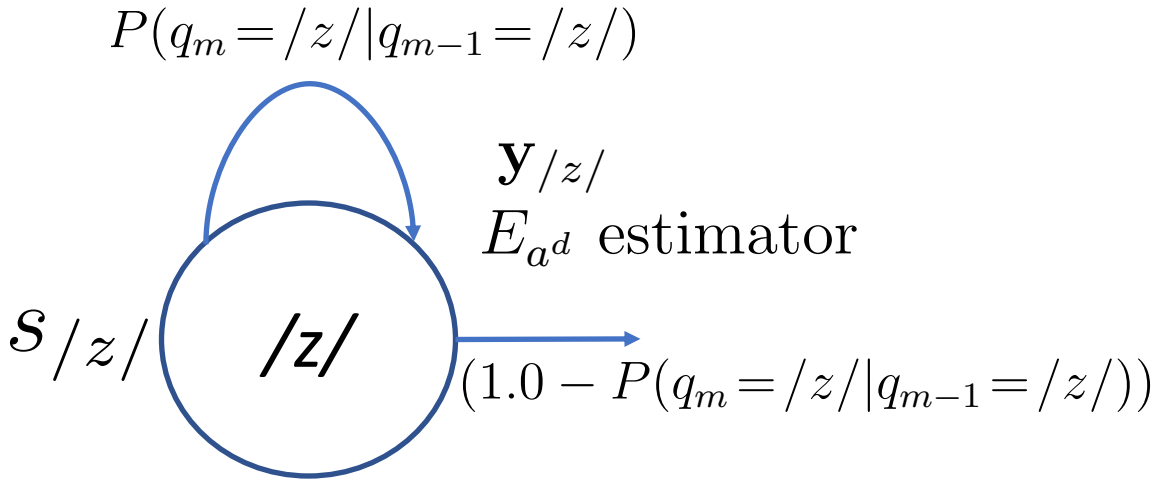
Illustrations

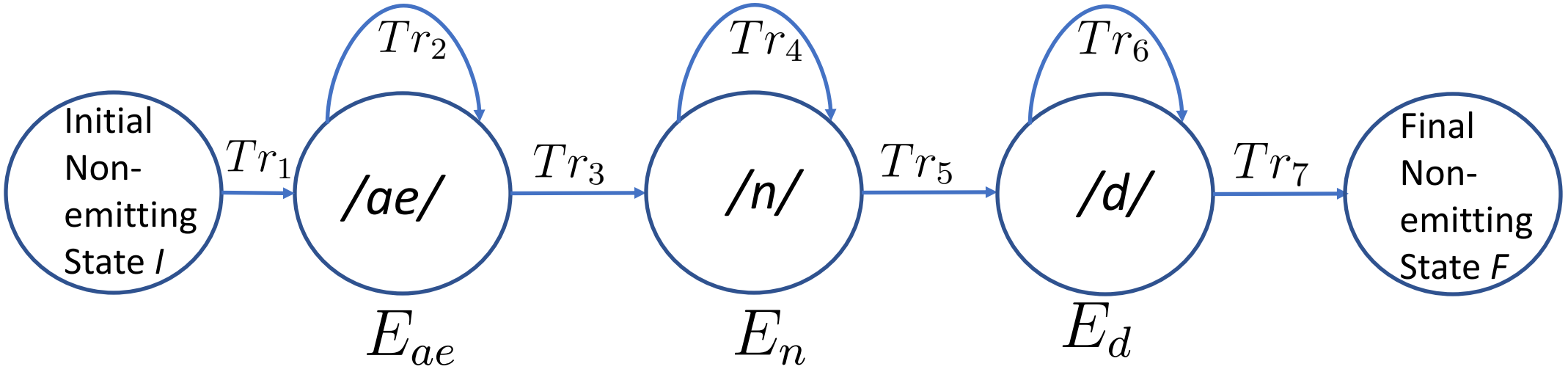


Let us denote the states as *k*

$$k \in \{/aa/, \cdots, /z/\}$$

E_{a^d} estimators: Single Gaussian, GMMs, ANN





$$Tr_1 = P(q_1 = /ae/|I) = 1.0$$

$$Tr_2 = P(q_m = /ae/|q_{m-1} = /ae/)$$

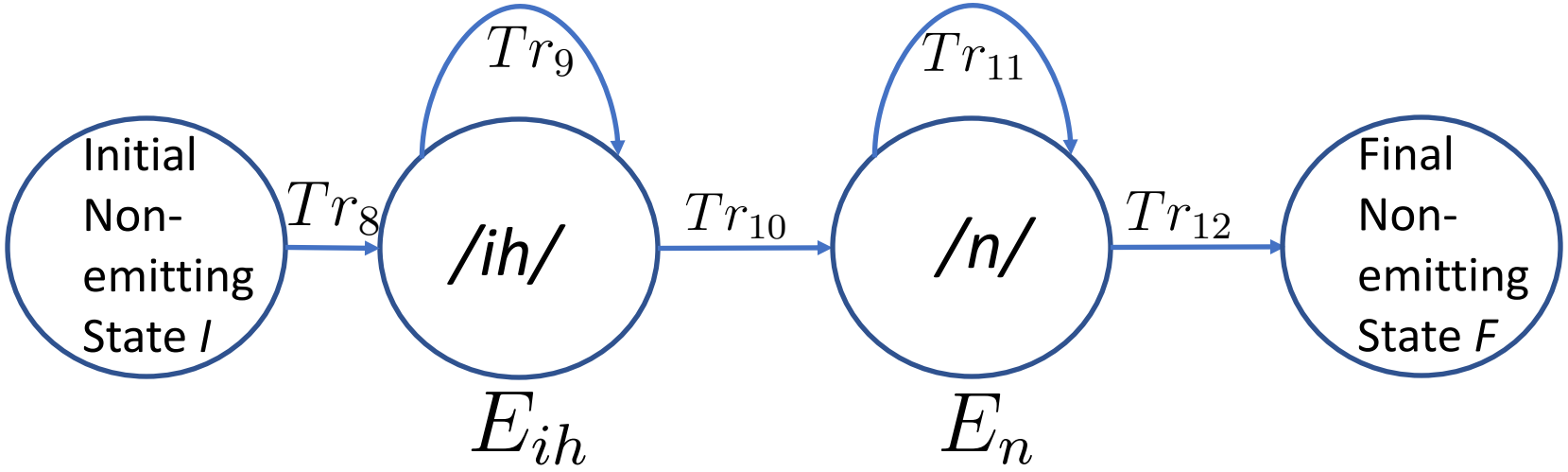
$$Tr_3 = P(q_m = /n/|q_{m-1} = /ae/) = 1.0 - Tr_2$$

$$Tr_4 = P(q_m = /n/|q_{m-1} = /n/)$$

$$Tr_5 = P(q_m = /d/|q_{m-1} = /n/) = 1.0 - Tr_4$$

$$Tr_6 = P(q_m = /d/|q_{m-1} = /d/)$$

$$Tr_7 = P(F|q_M = /d/) = 1.0 - Tr_6$$



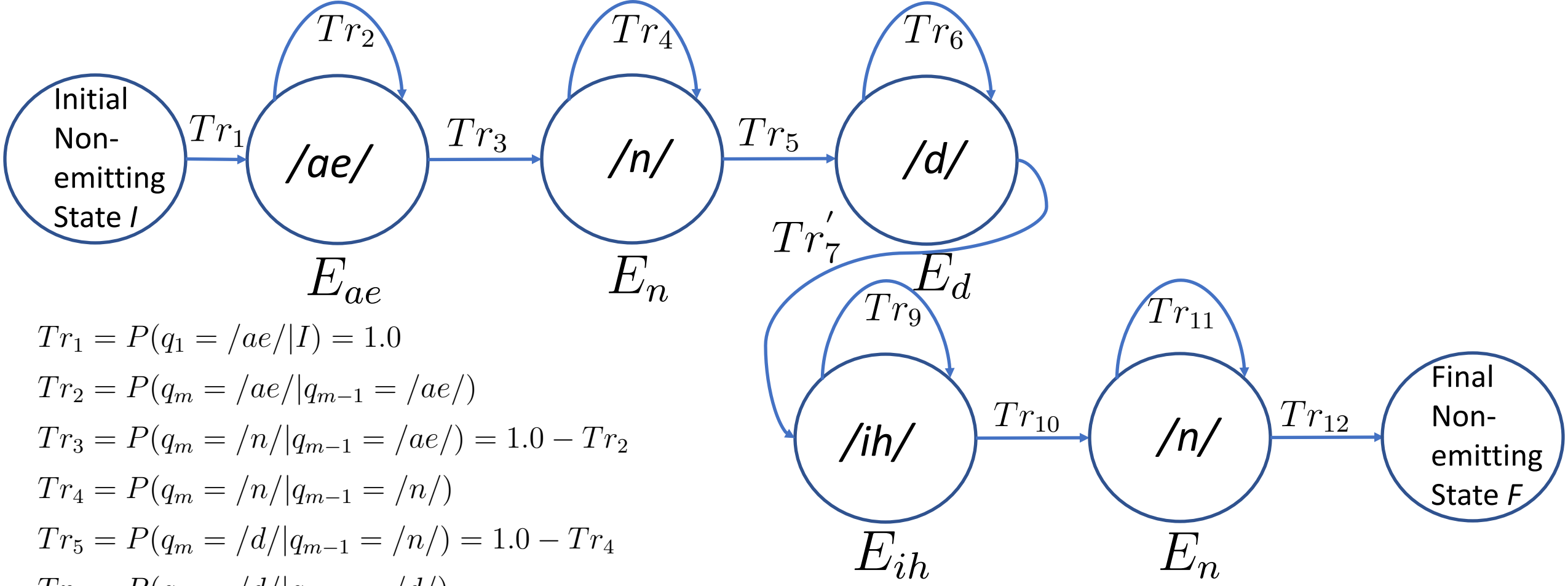
$$Tr_8 = P(q_1 = /ih/|I) = 1.0$$

$$Tr_9 = P(q_m = /ih/|q_{m-1} = /ih/)$$

$$Tr_{10} = P(q_m = /n/|q_{m-1} = /ih/) = 1.0 - Tr_9$$

$$Tr_{11} = P(q_m = /n/|q_{m-1} = /n/)$$

$$Tr_{12} = P(F|q_M = /n/) = 1.0 - Tr_{11}$$



$$Tr_1 = P(q_1 = /ae/|I) = 1.0$$

$$Tr_2 = P(q_m = /ae/|q_{m-1} = /ae/)$$

$$Tr_3 = P(q_m = /n/|q_{m-1} = /ae/) = 1.0 - Tr_2$$

$$Tr_4 = P(q_m = /n/|q_{m-1} = /n/)$$

$$Tr_5 = P(q_m = /d/|q_{m-1} = /n/) = 1.0 - Tr_4$$

$$Tr_6 = P(q_m = /d/|q_{m-1} = /d/)$$

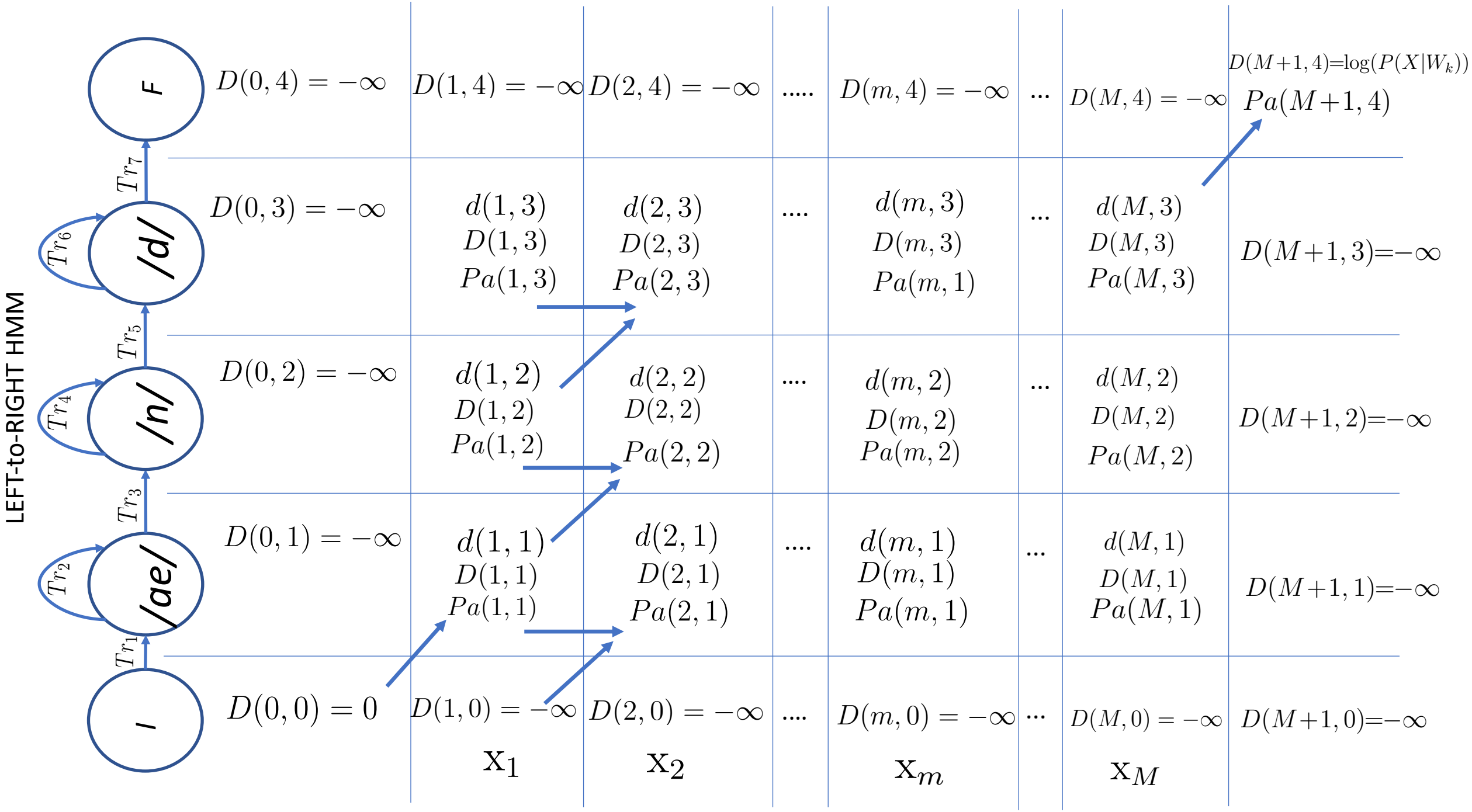
$$Tr_{7'} = P(q_m = /ih/|q_m - 1 = /d/) = Tr_7 \times Tr_8 = 1.0 - Tr_6$$

$$Tr_9 = P(q_m = /ih/|q_{m-1} = /ih/)$$

$$Tr_{10} = P(q_m = /n/|q_{m-1} = /ih/) = 1.0 - Tr_9$$

$$Tr_{11} = P(q_m = /n/|q_{m-1} = /n/)$$

$$Tr_{12} = P(F|q_M = /n/) = 1.0 - Tr_{11}$$


LEFT-to-RIGHT HMM
F
/d/
/n/
/ae/
I
Tr_7
Tr_6
Tr_5
Tr_4
Tr_3
Tr_2
Tr_1
$D(0,4) = -\infty$
$D(1,4) = -\infty$
$D(2,4) = -\infty$
.....
$D(m,4) = -\infty$
...
$D(M,4) = -\infty$
$D(M+1,4)=\log(P(X|W_k))$
$Pa(M+1,4)$
$D(0,3) = -\infty$
$d(1,3)$ $D(1,3)$ $Pa(1,3)$
$d(2,3)$ $D(2,3)$ $Pa(2,3)$
....
$d(m,3)$ $D(m,3)$ $Pa(m,1)$
...
$d(M,3)$ $D(M,3)$ $Pa(M,3)$
$D(M+1,3)=-\infty$
$D(0,2) = -\infty$
$d(1,2)$ $D(1,2)$ $Pa(1,2)$
$d(2,2)$ $D(2,2)$ $Pa(2,2)$
....
$d(m,2)$ $D(m,2)$ $Pa(m,2)$
...
$d(M,2)$ $D(M,2)$ $Pa(M,2)$
$D(M+1,2)=-\infty$
$D(0,1) = -\infty$
$d(1,1)$ $D(1,1)$ $Pa(1,1)$
$d(2,1)$ $D(2,1)$ $Pa(2,1)$
....
$d(m,1)$ $D(m,1)$ $Pa(m,1)$
...
$d(M,1)$ $D(M,1)$ $Pa(M,1)$
$D(M+1,1)=-\infty$
$D(0,0) = 0$
$D(1,0) = -\infty$
$D(2,0) = -\infty$
....
$D(m,0) = -\infty$
...
$D(M,0) = -\infty$
$D(M+1,0)=-\infty$
x_1
x_2
x_m
x_M

Let $n \in \{1, \cdots N\}$ denote the sequence of emitting states in the HMM

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

$$Pa(m,n) = \arg\max[(D(m-1,n) + \log(P(q_m = n|q_{m-1} = n)),$$
$$(D(m-1,n-1) + \log(P(q_m = n|q_{m-1} = n-1))]$$

LEFT-to-RIGHT HMM

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

In the example, there are $N = 3$ emitting states.

Let us consider $n = 2$, i.e. HMM state /n/

$$D(m,2) = d(m,2) + \max[(D(m-1,n) + \log(P(q_m = /n/|q_{m-1} = /n/)),$$
$$(D(m-1,n-1) + \log(P(q_m = /n/|q_{m-1} = /ae/))]$$

$$Pa(m,2) = \arg\max[(D(m-1,n) + \log(P(q_m = /n/|q_{m-1} = /n/)),$$
$$(D(m-1,n-1) + \log(P(q_m = /n/|q_{m-1} = /ae/))]$$

Local score :
$d(m,2) = \log(\mathbf{v}_m^{\mathrm{T}} \mathbf{y}_{/n/,k})$

HMM Training

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: emission distribution modeled by single multivariate 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: emission distribution modeled by 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: emission distribution modeled by 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

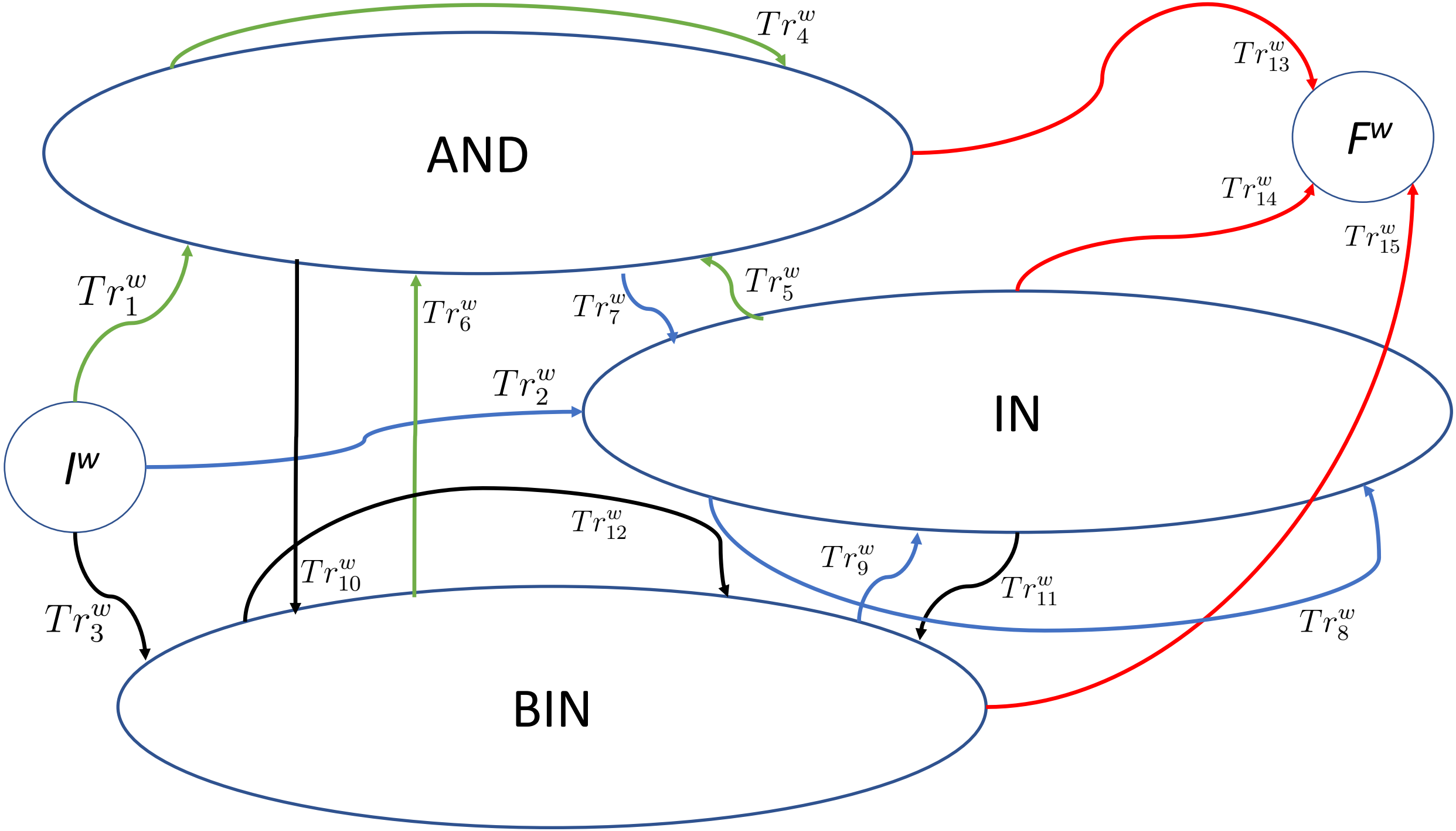
From the aligned sequence of states, estimate the self transition probabilities for each state by counting.

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.

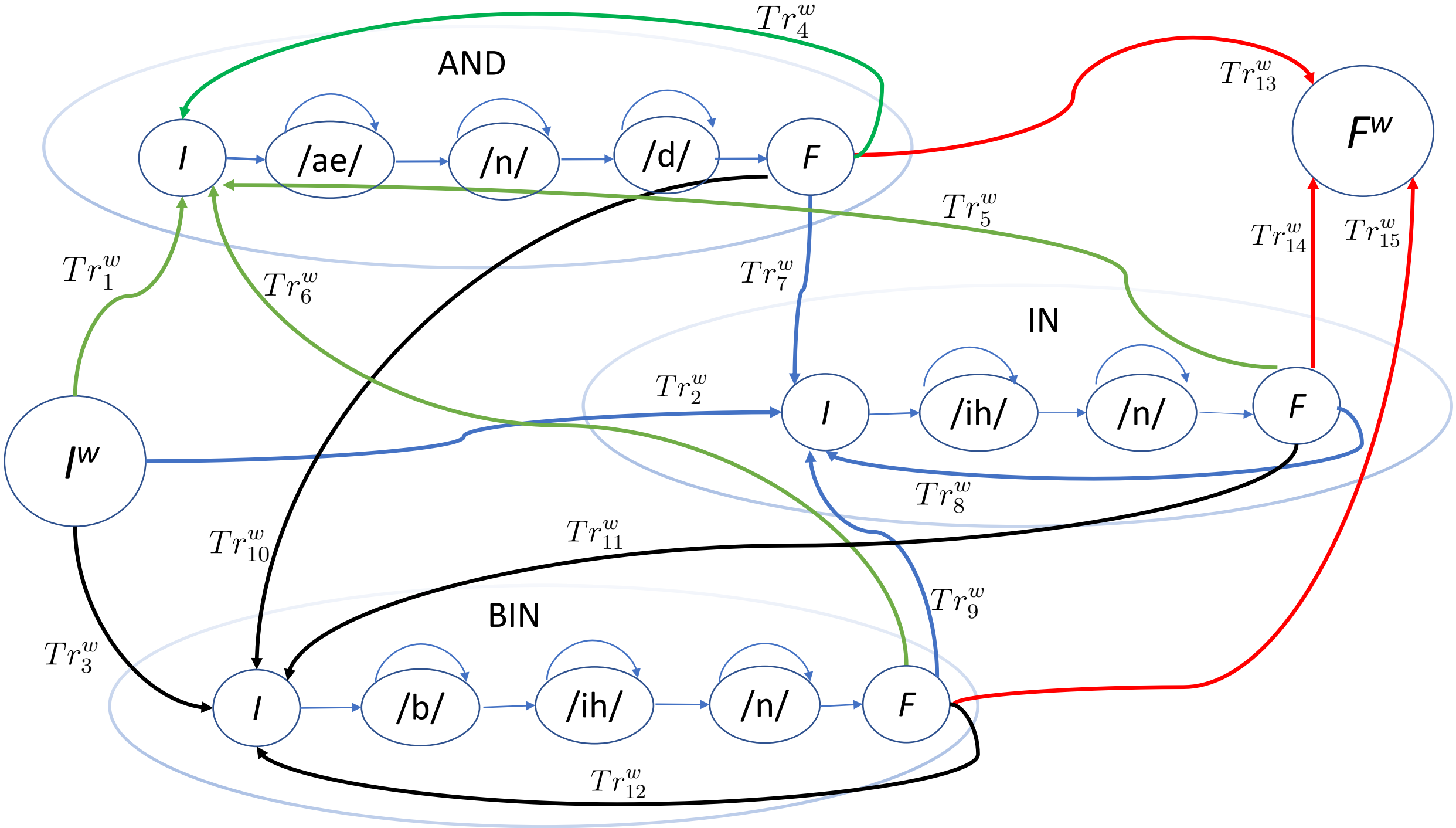
Decoding

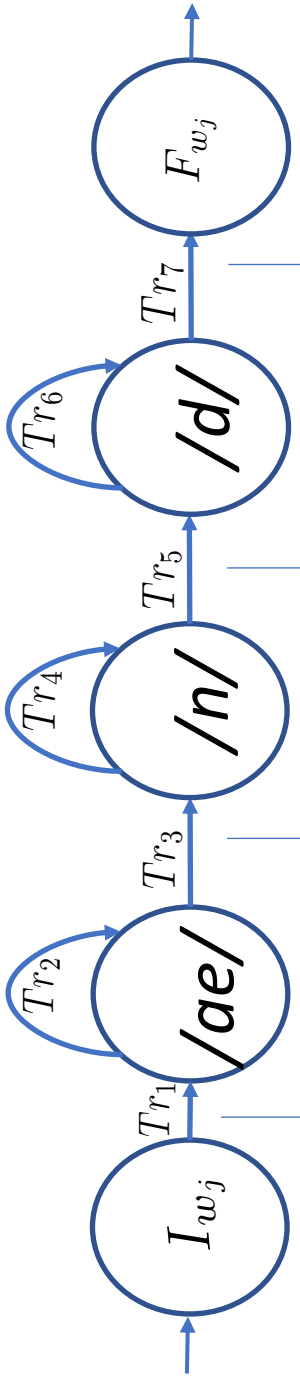

AND
IN
BIN
I^w
F^w
Tr_1^w
Tr_2^w
Tr_3^w
Tr_4^w
Tr_5^w
Tr_6^w
Tr_7^w
Tr_8^w
Tr_9^w
Tr_{10}^w
Tr_{11}^w
Tr_{12}^w
Tr_{13}^w
Tr_{14}^w
Tr_{15}^w

Discrete Markov model to estimate $P(W_k)$

$$Tr_1^w = P(\text{AND}|I^w)\ Tr_2^w = P(\text{IN}|I^w)\ Tr_3^w = P(\text{BID}|I^w)$$
$$Tr_4^w = P(\text{AND}|\text{AND})\ Tr_5^w = P(\text{AND}|\text{IN})\ Tr_6^w = P(\text{AND}|\text{BID})$$
$$Tr_7^w = P(\text{IN}|\text{AND})\ Tr_8^w = P(\text{IN}|\text{IN})\ Tr_9^w = P(\text{IN}|\text{BID})$$
$$Tr_{10}^w = P(\text{BID}|\text{AND})\ Tr_{11}^w = P(\text{BID}|\text{IN})\ Tr_{12}^w = P(\text{BID}|\text{BID})$$
$$Tr_{13}^w = P(F^w|\text{AND})\ Tr_{14}^w = P(F^w|\text{IN})\ Tr_{15}^w = P(F^w|\text{BID})$$

Parameters estimated using a large text database by counting.


AND
I
/ae/
/n/
/d/
F
IN
I
/ih/
/n/
F
BIN
I
/b/
/ih/
/n/
F
I^w
F^w
Tr_1^w
Tr_2^w
Tr_3^w
Tr_4^w
Tr_5^w
Tr_6^w
Tr_7^w
Tr_8^w
Tr_9^w
Tr_{10}^w
Tr_{11}^w
Tr_{12}^w
Tr_{13}^w
Tr_{14}^w
Tr_{15}^w



$D(0, F_{w_j}) = -\infty$	$D(1, F_{w_j})$ $Pa(1, F_{w_j})$	$D(2, F_{w_j})$ $Pa(2, F_{w_j})$	.....	$D(m, F_{w_j})$ $Pa(m, F_{w_j})$	...	$D(M, F_{w_j})$ $Pa(M, F_{w_j})$	$D(T+1, F_{w_j})$ $Pa(T+1, F_{w_j})$
$D(0, 3) = -\infty$	$d(1, 3)$ $D(1, 3)$ $Pa(1, 3)$	$d(2, 3)$ $D(2, 3)$ $Pa(2, 3)$	....	$d(m, 3)$ $D(m, 3)$ $Pa(m, 1)$	...	$d(M, 3)$ $D(M, 3)$ $Pa(M, 3)$	$D(M+1, 3) = -\infty$
$D(0, 2) = -\infty$	$d(1, 2)$ $D(1, 2)$ $Pa(1, 2)$	$d(2, 2)$ $D(2, 2)$ $Pa(2, 2)$	....	$d(m, 2)$ $D(m, 2)$ $Pa(m, 2)$	...	$d(M, 2)$ $D(M, 2)$ $Pa(M, 2)$	$D(M+1, 2) = -\infty$
$D(0, 1) = -\infty$	$d(1, 1)$ $D(1, 1)$ $Pa(1, 1)$	$d(2, 1)$ $D(2, 1)$ $Pa(2, 1)$	....	$d(m, 1)$ $D(m, 1)$ $Pa(m, 1)$	...	$d(M, 1)$ $D(M, 1)$ $Pa(M, 1)$	$D(M+1, 1) = -\infty$
$D(0, I_{w_j})$	$D(1, I_{w_j})$ $Pa(1, I_{w_j})$	$D(2, I_{w_j})$ $Pa(2, I_{w_j})$	....	$D(m, I_{w_j})$ $Pa(m, I_{w_j})$	...	$D(M, I_{w_j})$ $Pa(M, I_{w_j})$	$D(M+1, I_{w_j}) = -\infty$
	x_1	x_2		x_m		x_M	

Key changes in dynamic programming for each word w_j in the vocabulary

allow the possibility to begin and end any word at any time frame.

$$D(0, I_{w_j}) = \log(P(w_j|I^w))$$

$$D(0, I_{w_j}) = \log(P(w_j = \text{AND}|I^w)) = \log(Tr_1^w) \text{ (For the illustrated example)}$$

$$D(m, I_{w_j}) = \max_{j' \in \{1 \cdots J\}} [D(m-1, F_{w_{j'}}) + \log(P(w_j|w_{j'}))]$$

$$Pa(m, I_{w_j}) = \arg\max_{j' \in \{1 \cdots J\}} [D(m-1, F_{w_{j'}}) + \log(P(w_j|w_{j'}))]$$

I^w and F^w denote the initial and final states of the language model DMM

I_{w_j} and F_{w_j} denote the non-emitting initial and final states of the HMM of word w_j

J denotes the number of words in the vocabulary

After reaching end of utterance i.e. last time frame M :

$$\text{w}_{la} = \arg\max_{j' \in \{1, \cdots J\}} D(M+1, F_{w_{j'}}) + \log(P(F^w|w_{j'})) \text{(start back tracking using } Pa(,))$$

w_{la} denotes the recognized last word in the utterance.