Pre-midsem Revision

Lecture 11

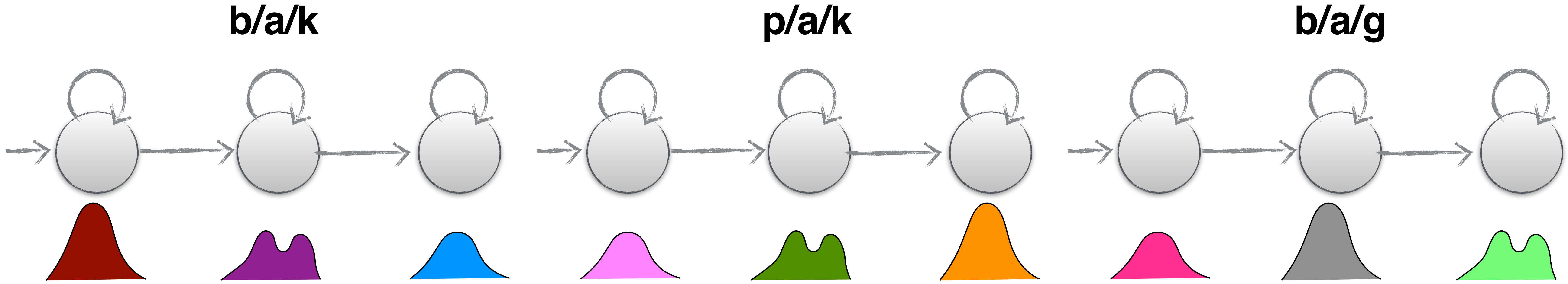
CS 753

Instructor: Preethi Jyothi

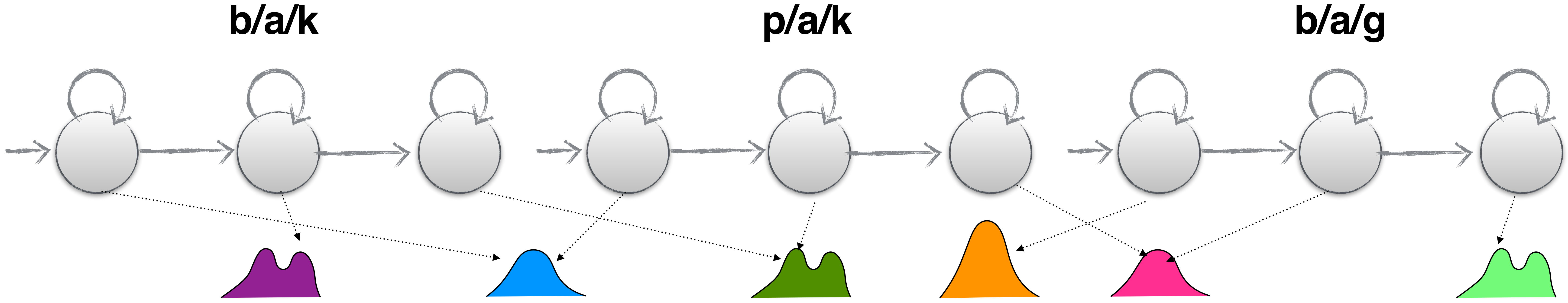
Tied-state Triphone Models

State Tying

- Observation probabilities are shared across triphone states which generate acoustically similar data

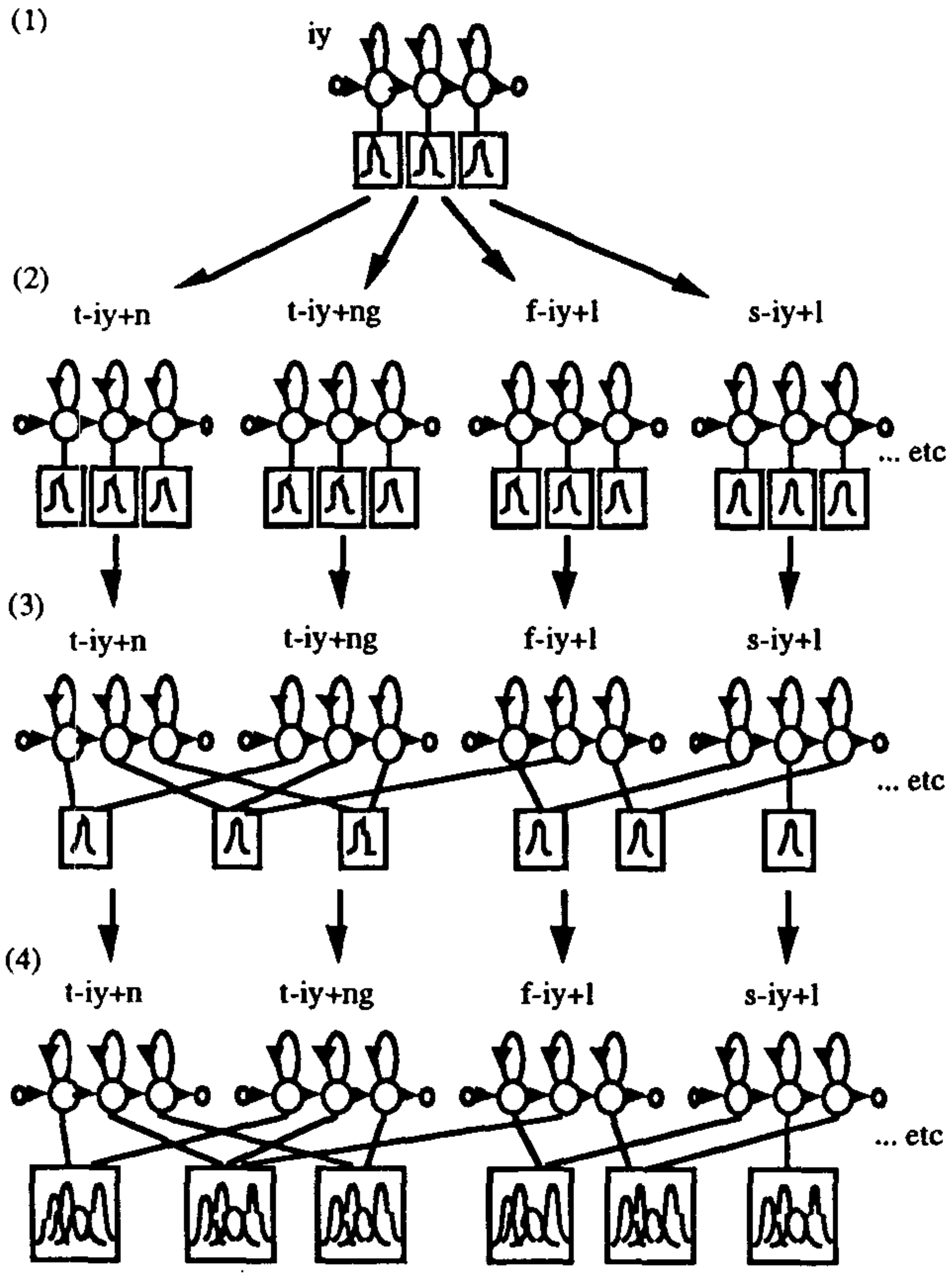


Triphone HMMs (No sharing)



Triphone HMMs (State Tying)

Tied state HMMs

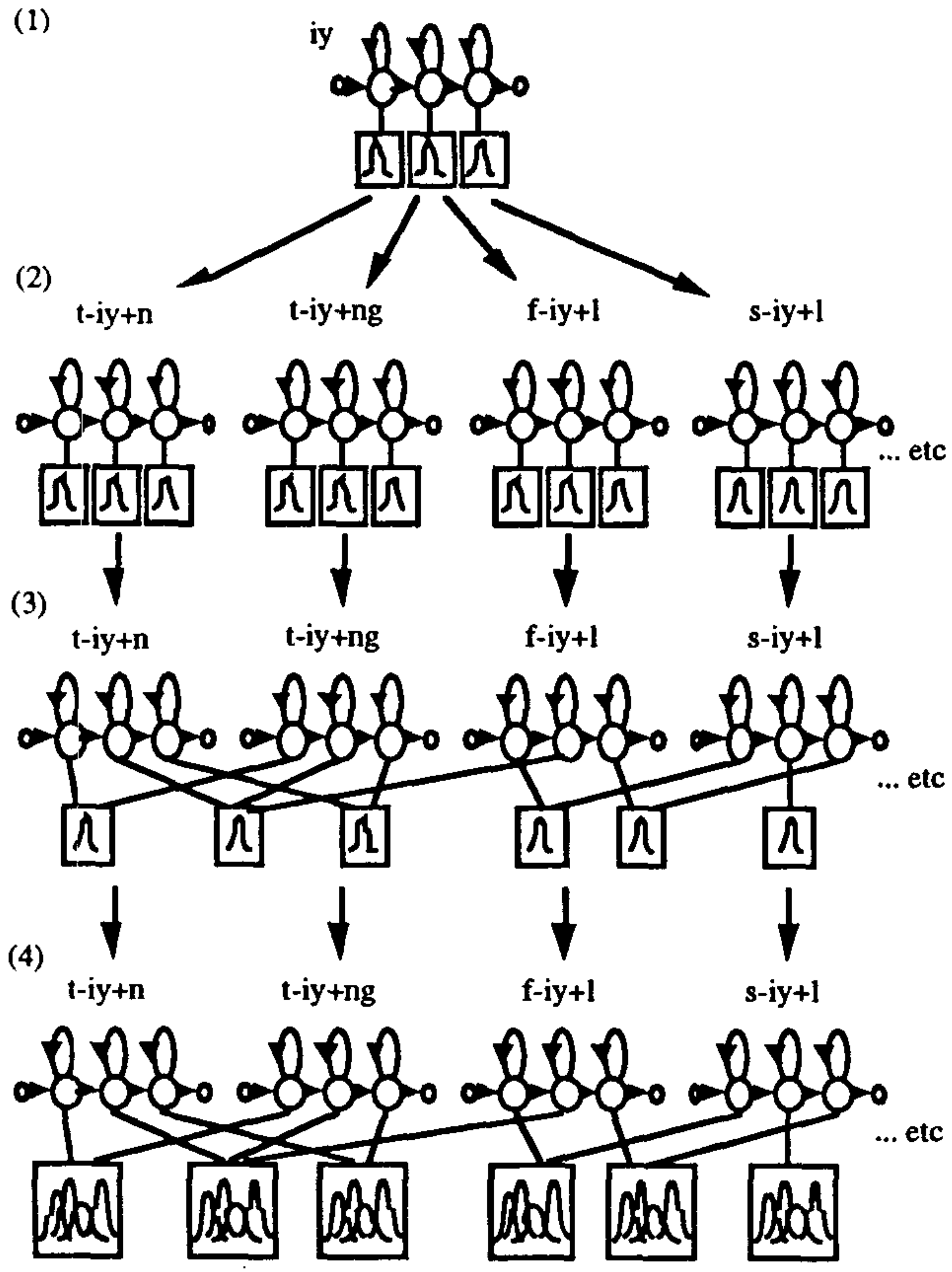


Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.
4. Number of mixture components in each tied state is increased and models re-estimated using BW

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs



Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.
4. Number of mixture components in each tied state is increased and models re-estimated using BW

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs: Step 2

Clone these monophone distributions to initialise a set of untied triphone models

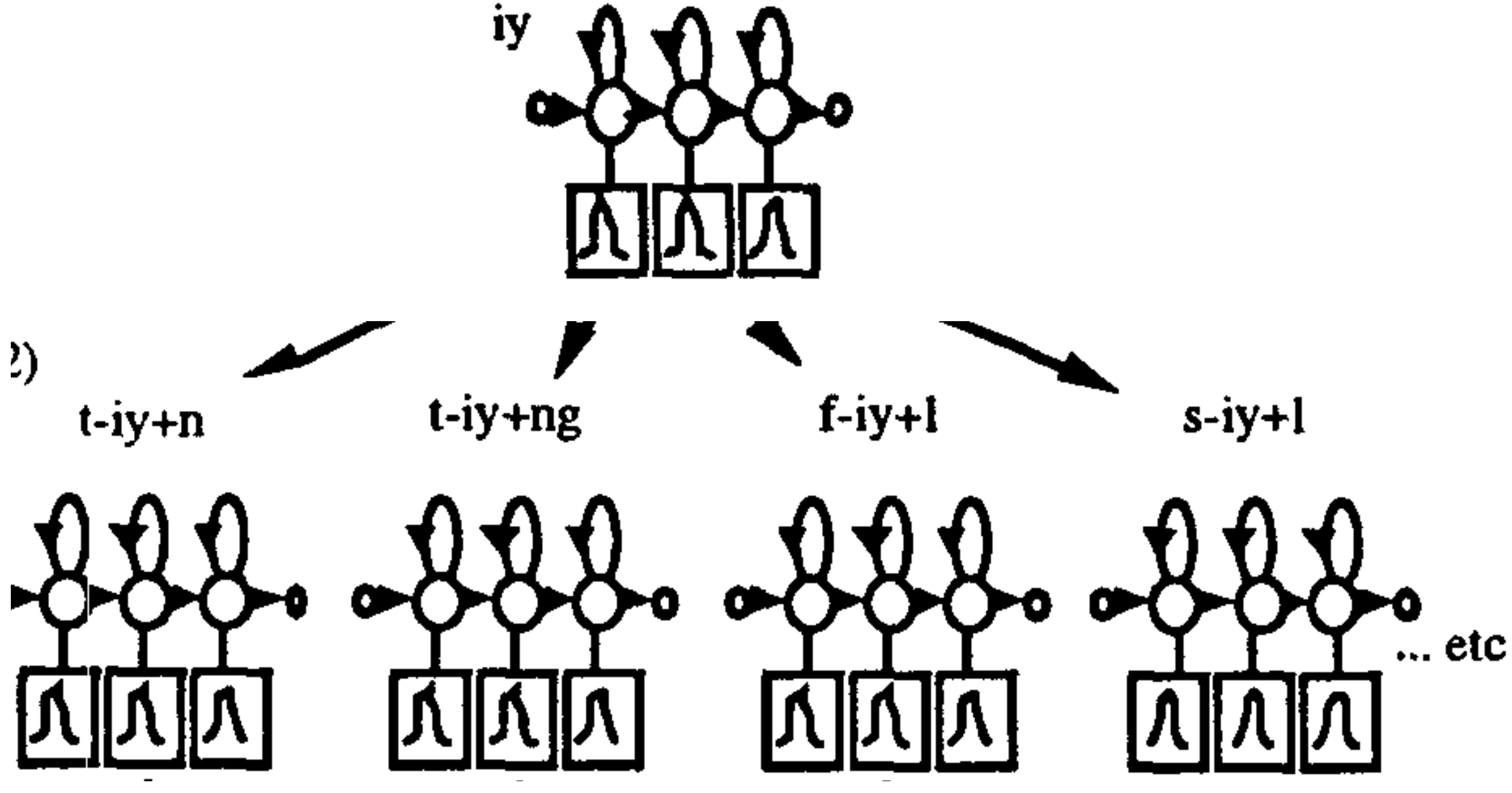
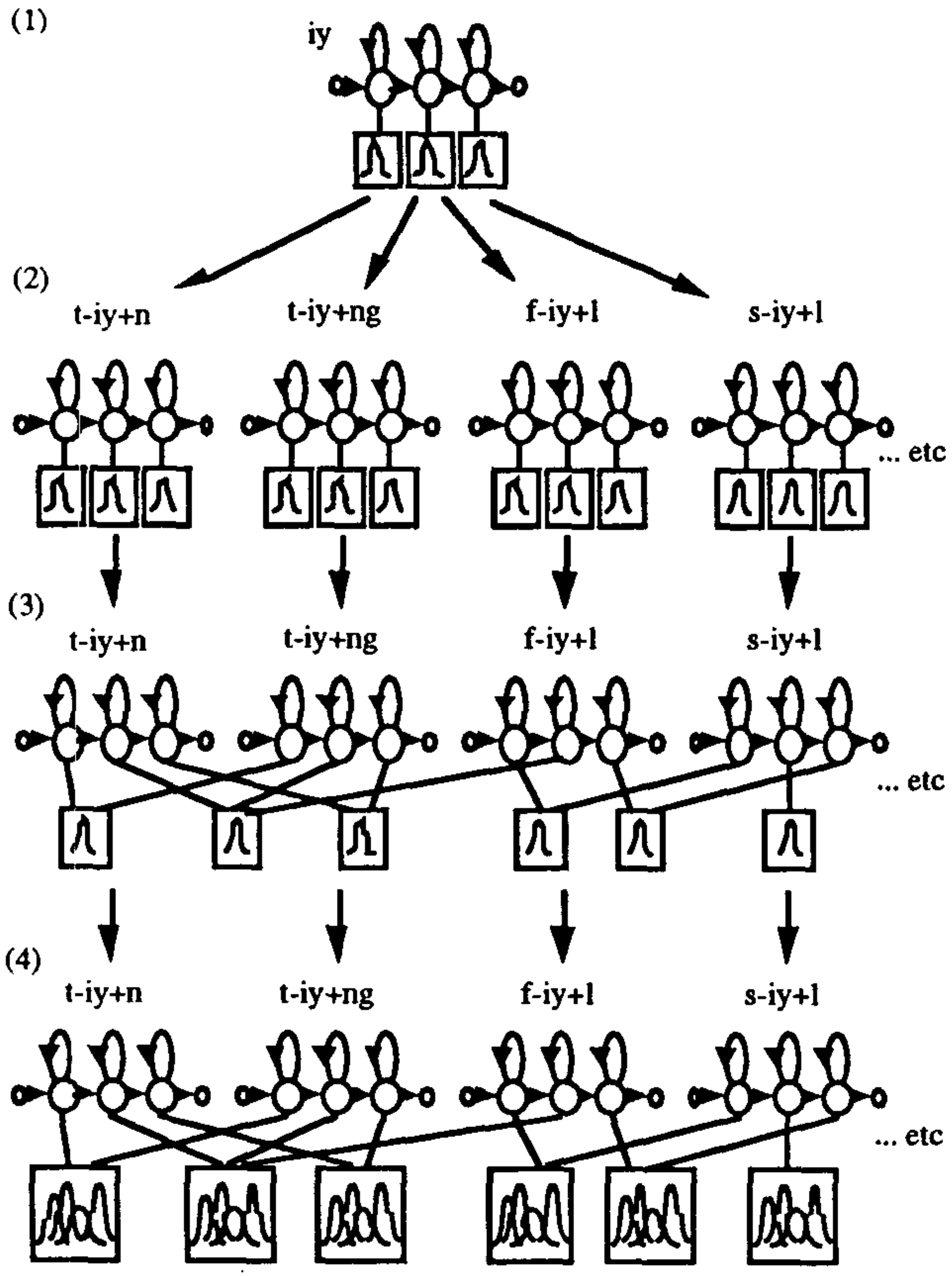


Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs

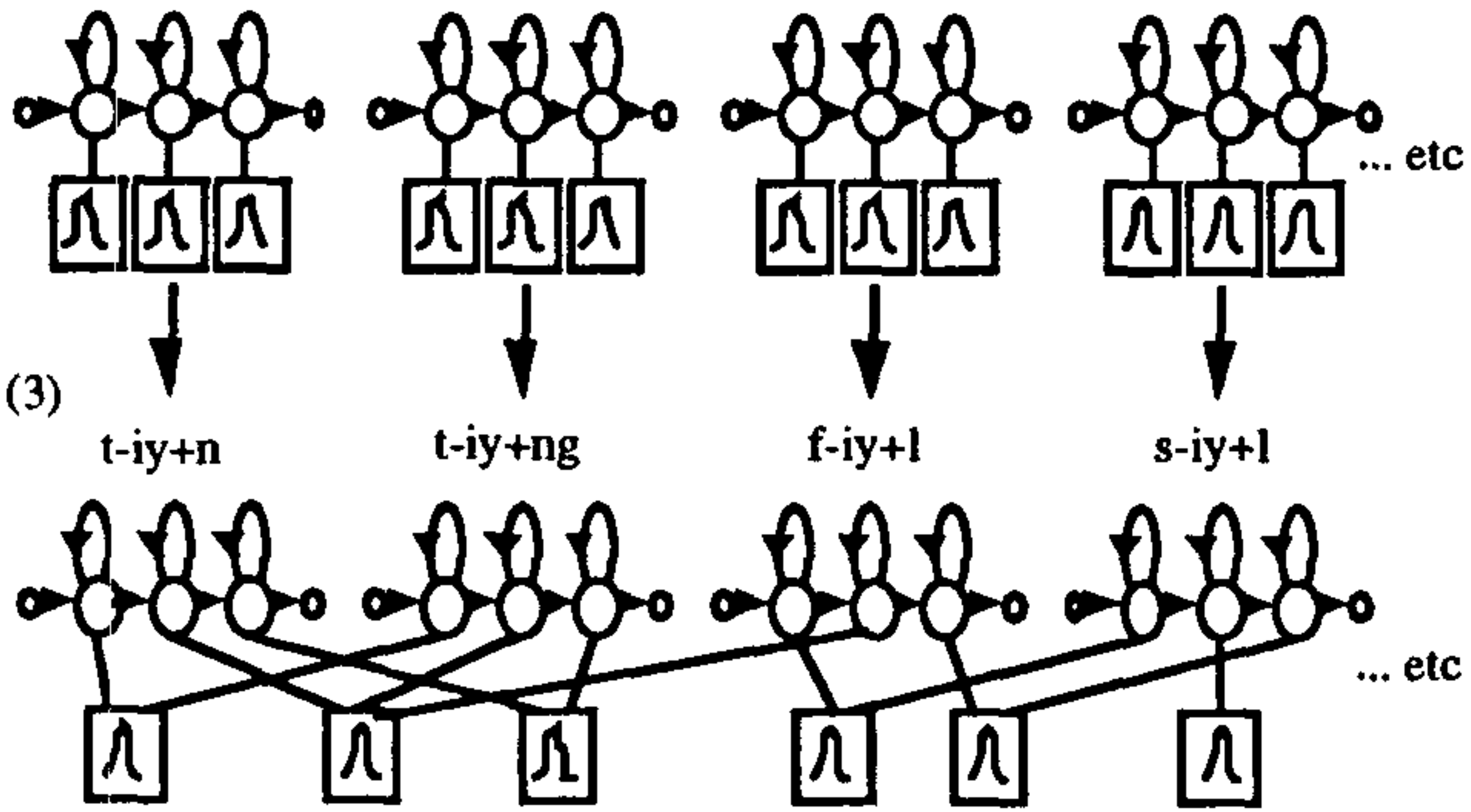


Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.
4. Number of mixture components in each tied state is increased and models re-estimated using BW

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Tied state HMMs: Step 3

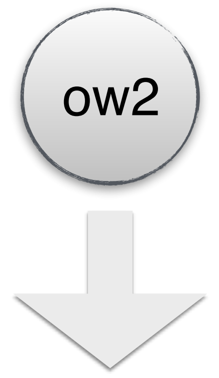


Use decision trees to determine which states should be tied together

Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states



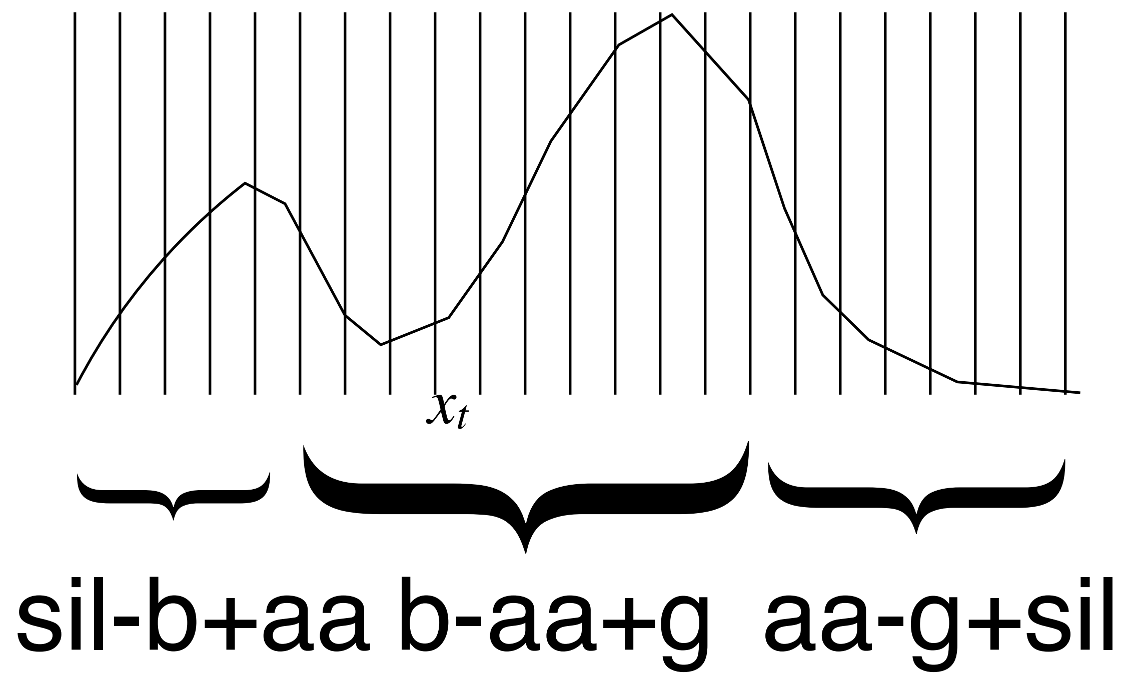
DT for center state of [ow]

Uses all training data tagged with *-ow_2+*

Head node
$aa_2/ow_2/f_2$, $aa_2/ow_2/s_2$,
$aa_2/ow_2/d_2$, $h_2/ow_2/p_2$,
$aa_2/ow_2/n_2$, $aa_2/ow_2/g_2$,
...

Training data for DT nodes

- Align training instance $x = (x_1, \ldots, x_T)$ where $x_i \in \mathbb{R}^d$ with a set of triphone HMMs
- Use Viterbi algorithm to find the best HMM triphone state sequence corresponding to each x
- Tag each x_t with ID of current phone along with left-context and right-context

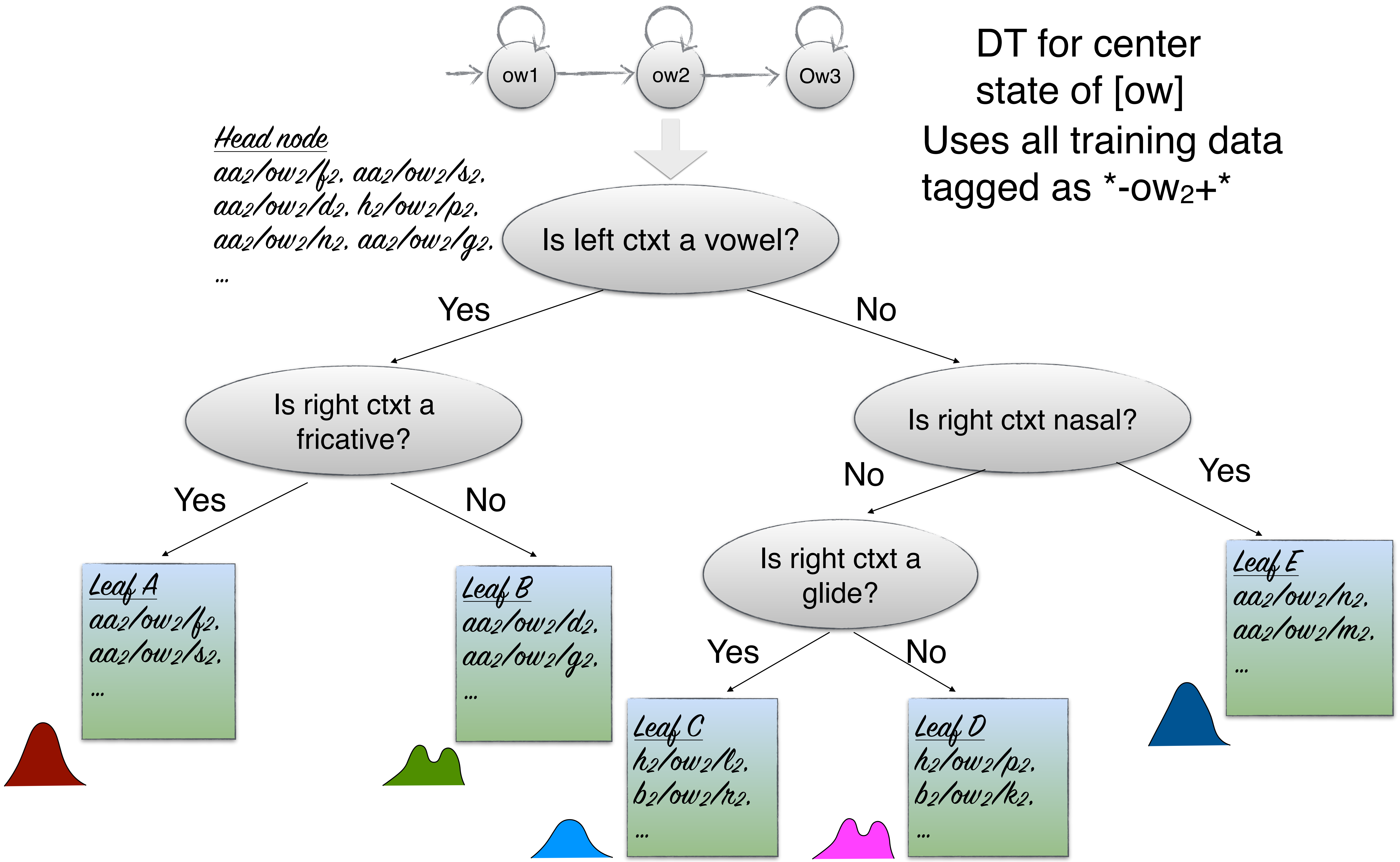


x_t is tagged with ID b_2-aa_2+g_2 i.e. x_t is aligned with the second state of the 3-state HMM corresponding to the triphone b-aa+g

- Training data corresponding to state j in phone p: Gather all x_t's that are tagged with ID *-p_j+*

Example: Phonetic Decision Tree (DT)

One tree is constructed for each state of each monophone to cluster all the corresponding triphone states



How do we build these phone DTs?

1. What questions are used?

   Linguistically-inspired binary questions: “Does the left or right phone come from a broad class of phones such as vowels, stops, etc.?” “Is the left or right phone [k] or [m]?”

2. What is the training data for each phone state, p_j? (root node of DT)

   All speech frames that align with the j^{th} state of every triphone HMM that has p as the middle phone

3. What criterion is used at each node to find the best question to split the data on?

   Find the question which partitions the states in the parent node so as to give the maximum increase in log likelihood

Likelihood of a cluster of states

- If a cluster of HMM states, $S = \{s_1, s_2, \ldots, s_M\}$ consists of M states and a total of K acoustic observation vectors are associated with S, $\{x_1, x_2 \ldots, x_K\}$, then the log likelihood associated with S is:

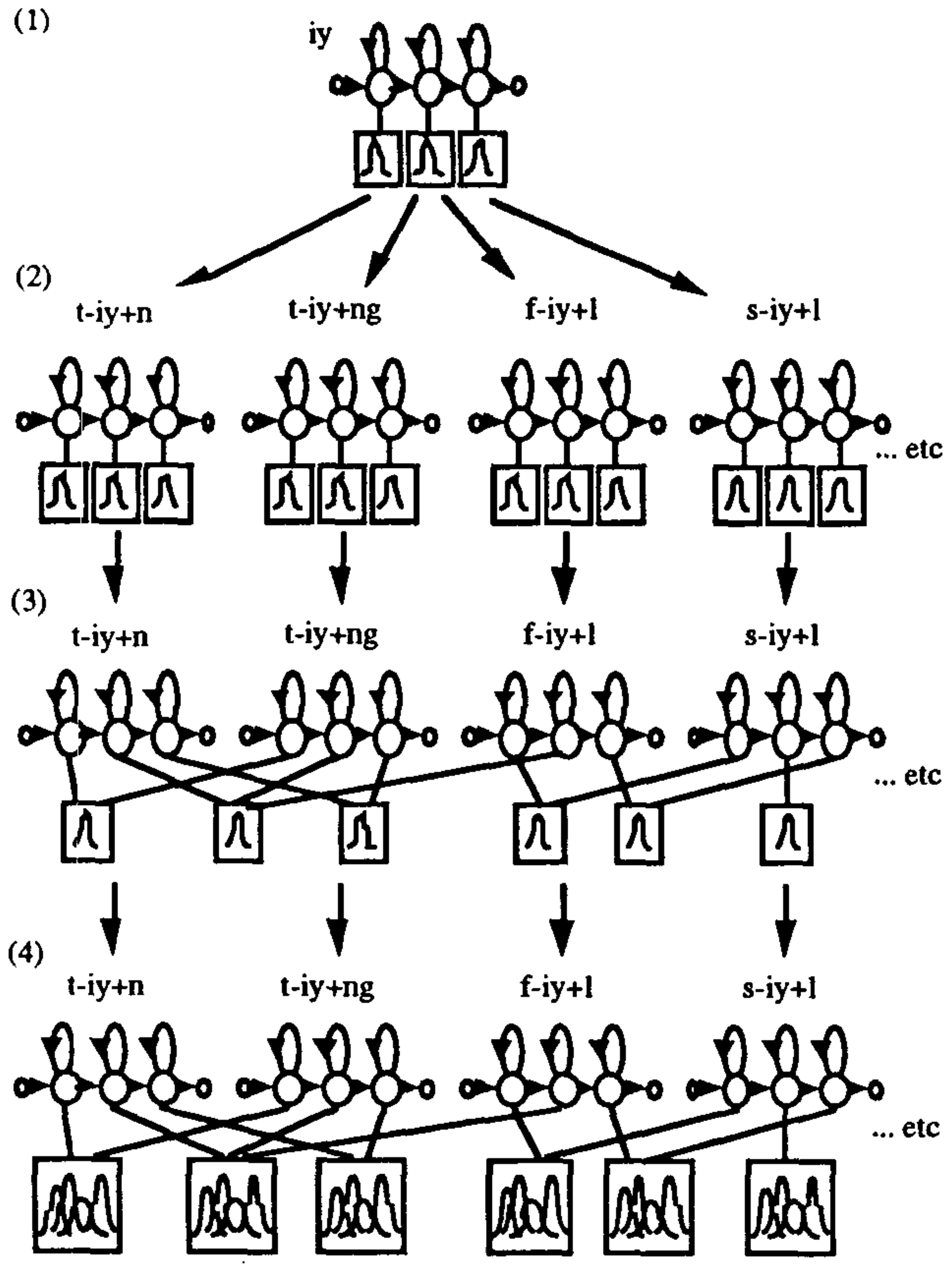
$$\mathcal{L}(S) = \sum_{i=1}^{K} \sum_{s \in S} \log \Pr(x_i; \mu_S, \Sigma_S) \gamma_s(x_i)$$

- For a question q that splits S into S_{yes} and S_{no}, compute the following quantity:

$$\Delta_q = \mathcal{L}(S^q_{\text{yes}}) + \mathcal{L}(S^q_{\text{no}}) - \mathcal{L}(S)$$

- Go through all questions, find Δ_q for each question q and choose the question for which Δ_q is the biggest
- Terminate when: Final Δ_q is below a threshold or data associated with a split falls below a threshold

Tied state HMMs



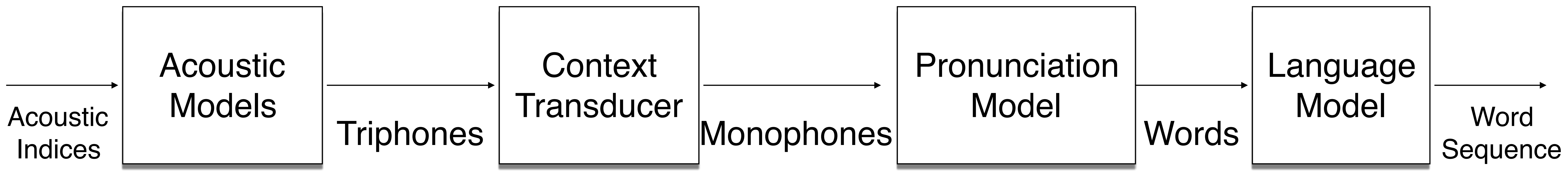
Four main steps in building a tied state HMM system:

1. Create and train 3-state monophone HMMs with single Gaussian observation probability densities
2. Clone these monophone distributions to initialise a set of untied triphone models. Train them using Baum-Welch estimation. Transition matrix remains common across all triphones of each phone.
3. For all triphones derived from the same monophone, cluster states whose parameters should be tied together.
4. Number of mixture components in each tied state is increased and models re-estimated using BW

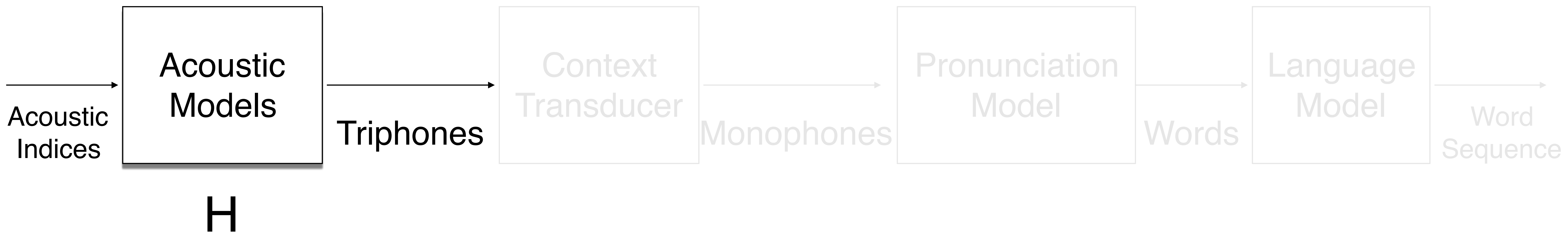
Image from: Young et al., “Tree-based state tying for high accuracy acoustic modeling”, ACL-HLT, 1994

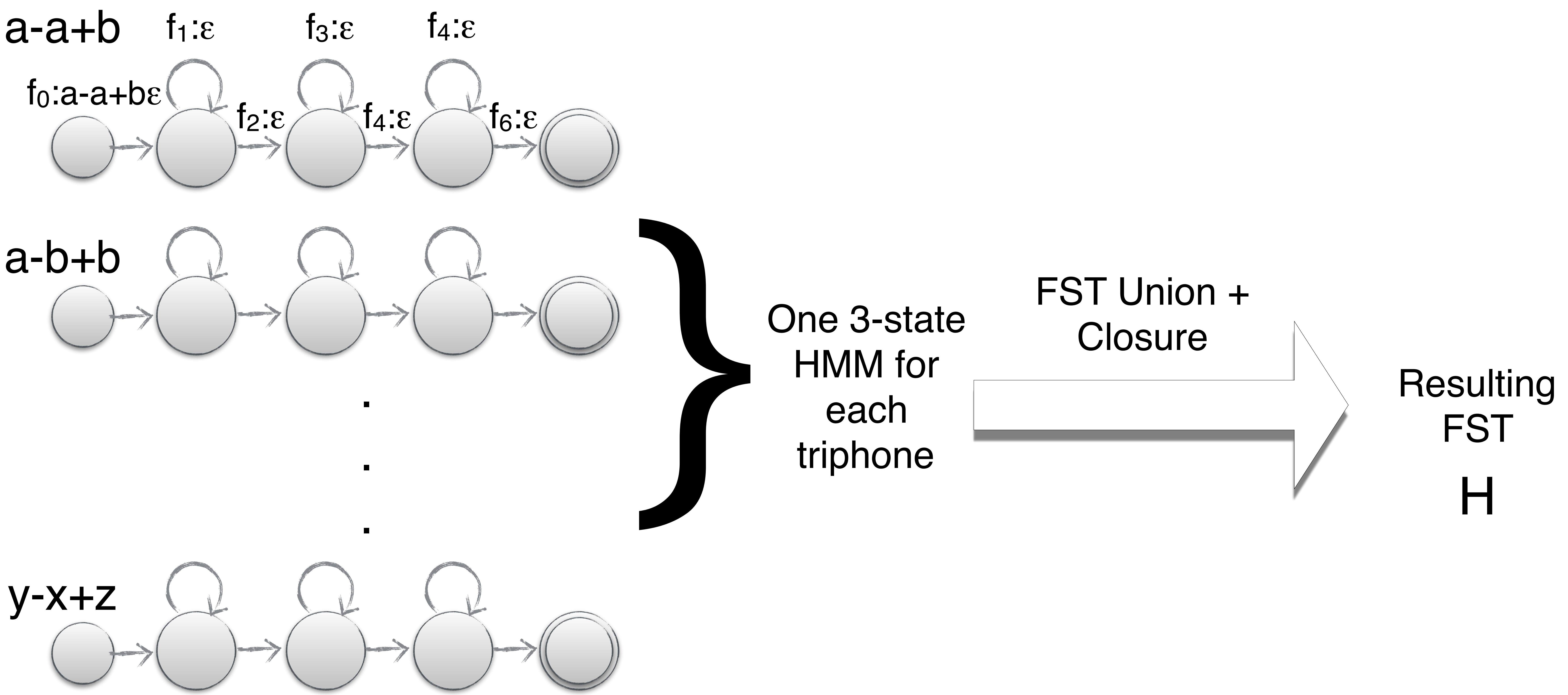
WFSTs for ASR

WFST-based ASR System

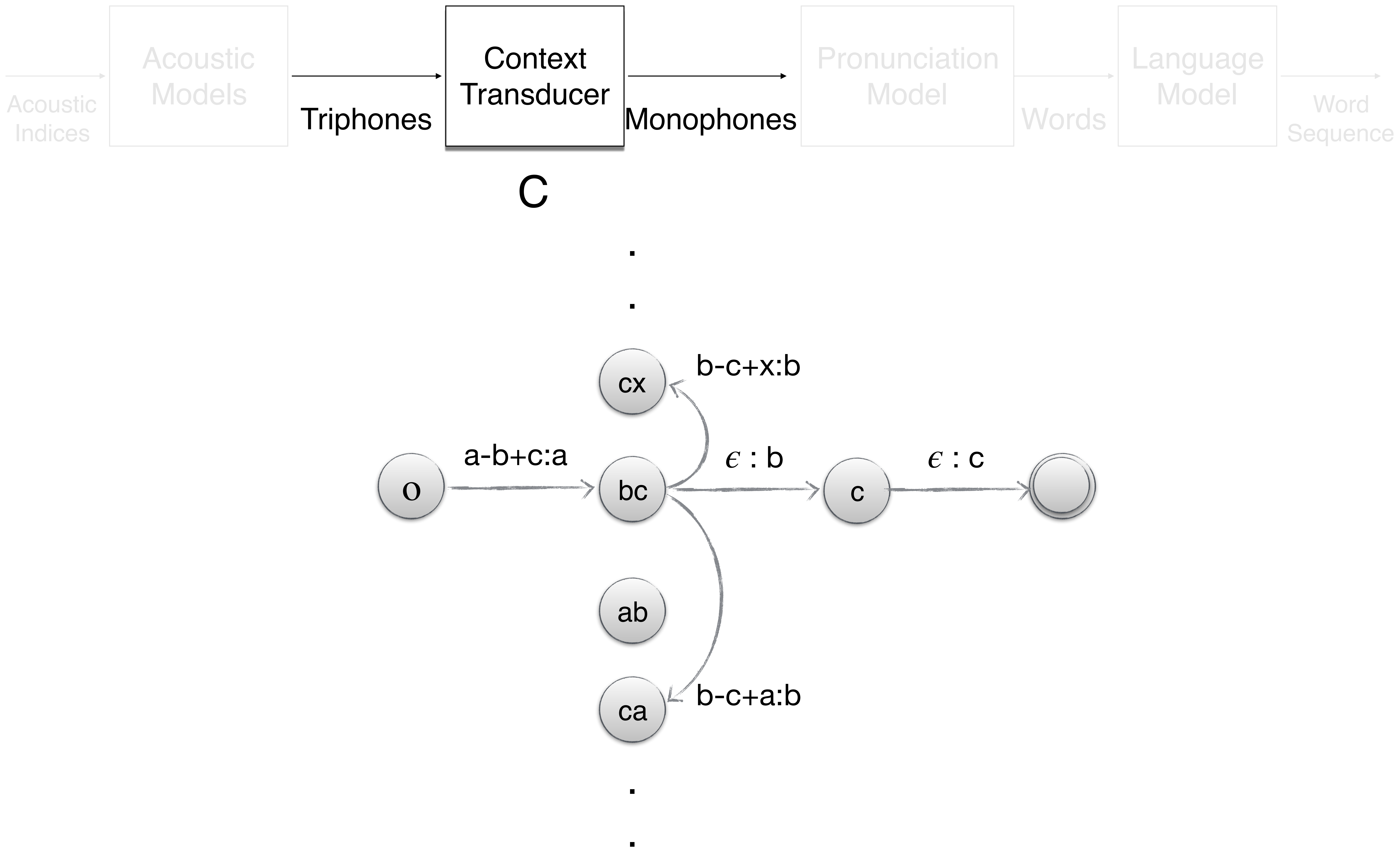


WFST-based ASR System

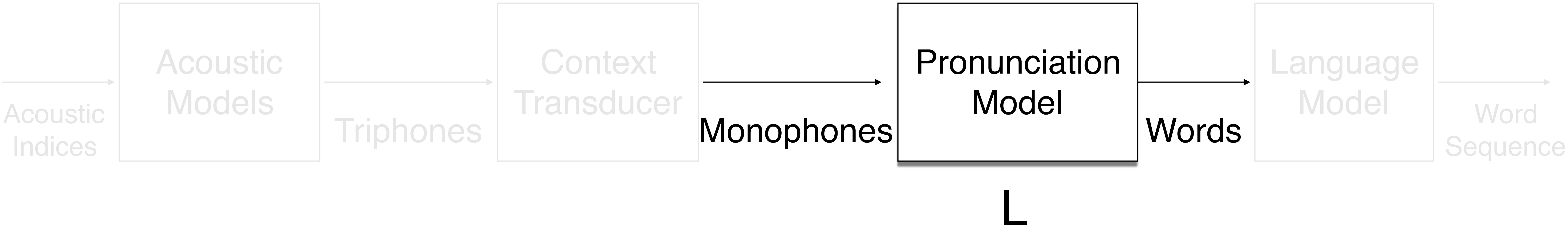




WFST-based ASR System



WFST-based ASR System



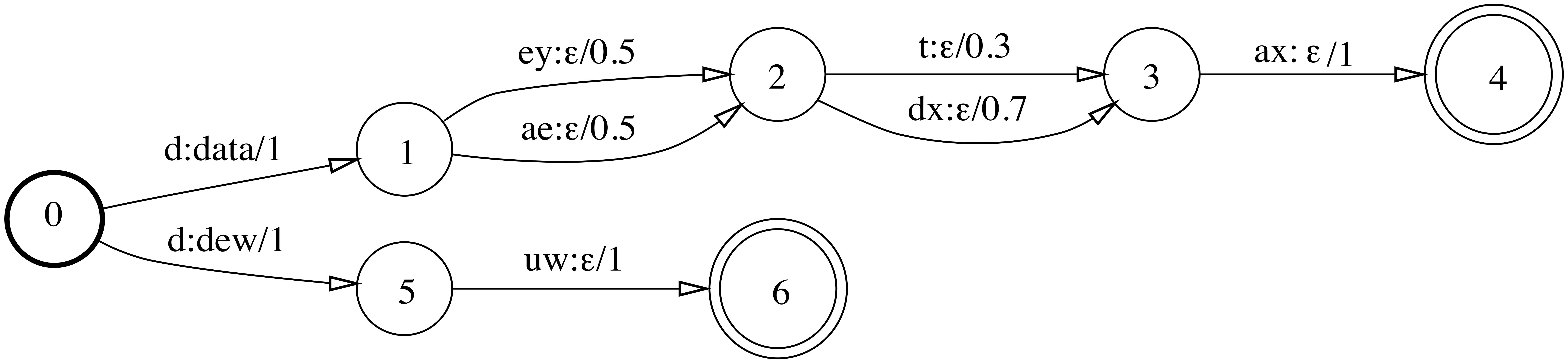
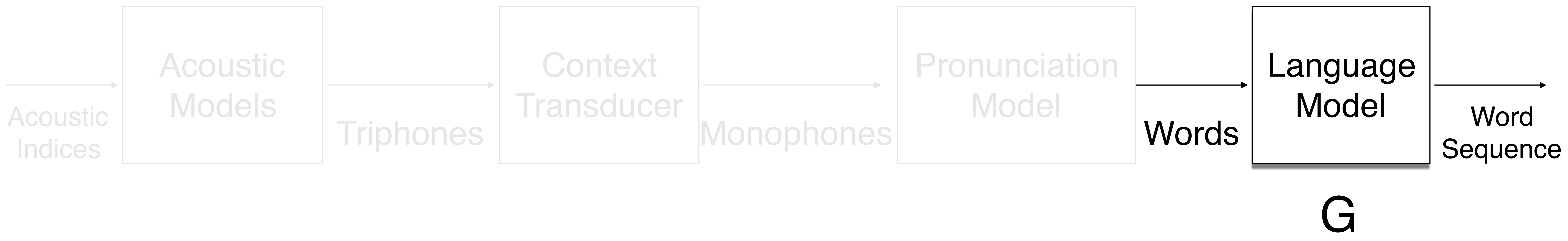
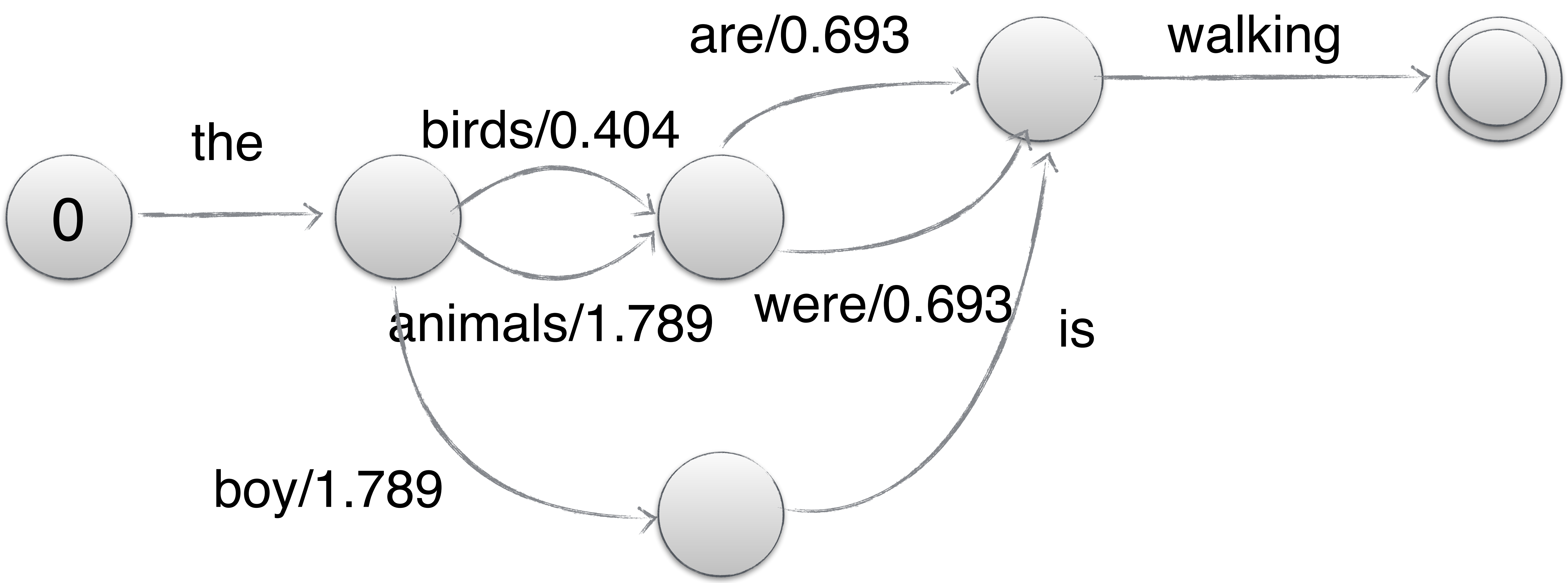


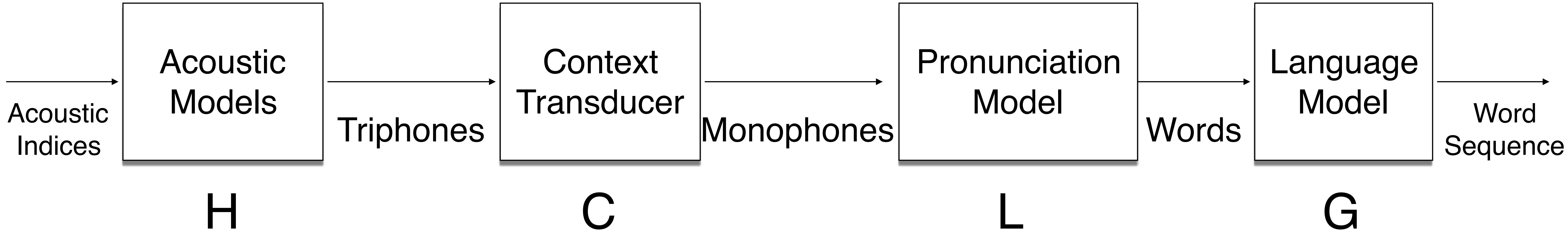
Figure reproduced from "Weighted Finite State Transducers in Speech Recognition", Mohri et al., 2002

WFST-based ASR System





Decoding

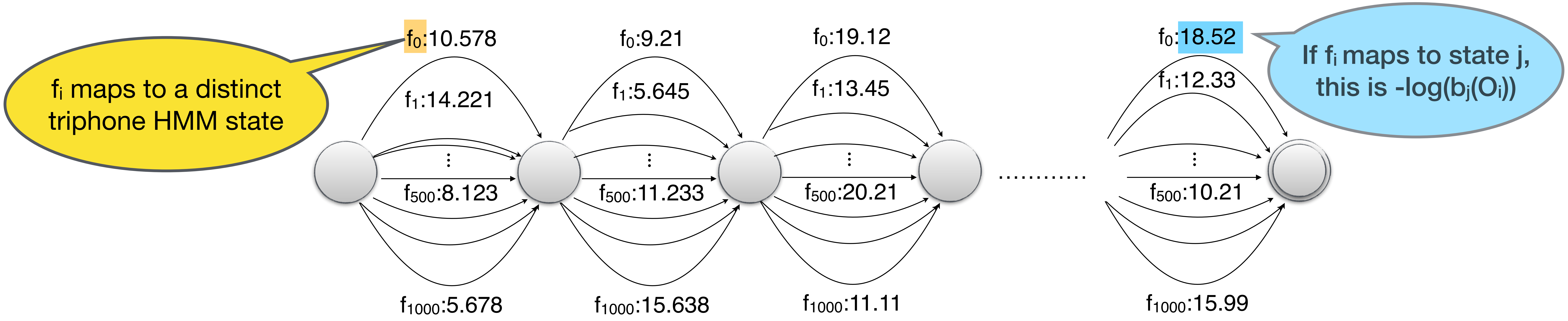


Carefully construct a decoding graph D using optimization algorithms:

$$D = \min(\det(H \circ \det(C \circ \det(L \circ G))))$$

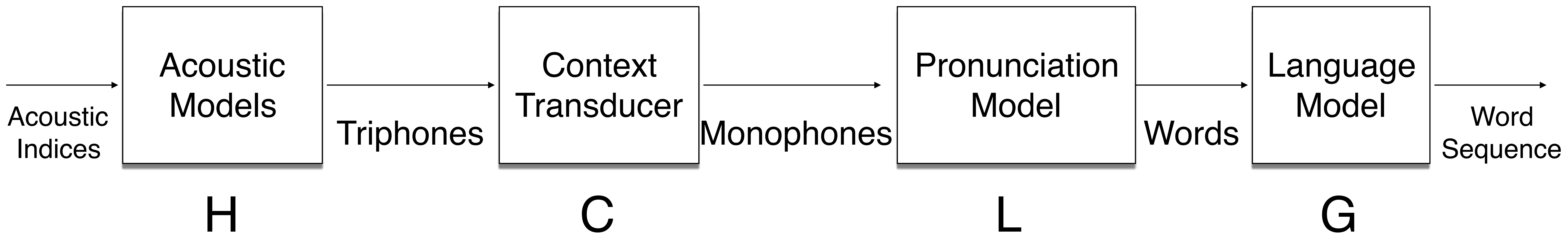
Given a test utterance O, how do I decode it?

Assuming ample compute, first construct the following machine X from O.



"Weighted Finite State Transducers in Speech Recognition", Mohri et al., Computer Speech & Language, 2002

Decoding

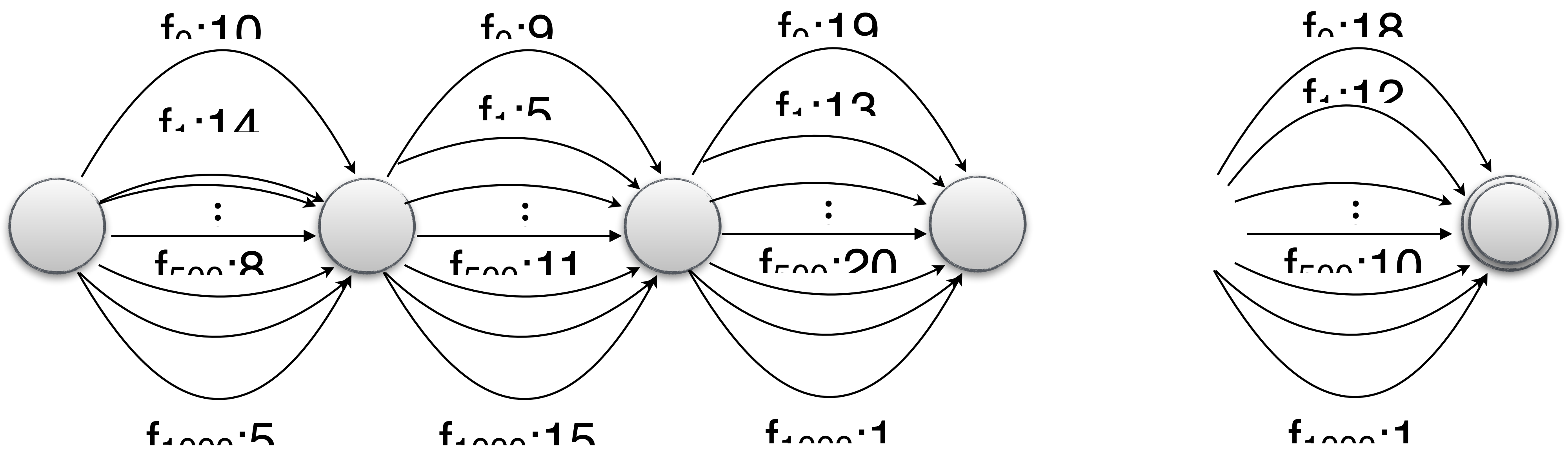


Carefully construct a decoding graph D using optimization algorithms:

$$D = \min(\det(H \circ \det(C \circ \det(L \circ G))))$$

Given a test utterance O, how do I decode it?

Assuming ample compute, first construct the following machine X from O.



X

$$W^* = \arg\min_{W = out[\pi]} X \circ D$$

where π is a path in the composed FST

out$[\pi]$ is the output label sequence of π

X is never typically constructed;

D is traversed dynamically using approximate search algorithms

(discussed later in the semester)

Ngram LM Smoothing

Good-Turing Discounting

- Good-Turing discounting states that for any token that occurs r times, we should use an adjusted count $r^* = \theta(r) = (r + 1)N_{r+1}/N_r$ where N_r is the number of tokens with r counts

- Good-Turing counts for unseen events: $\theta(0) = N_1/N_0$

- For large r, many instances of $N_{r+1} = 0$.
  A solution: Smooth N_r using a best-fit power law
  once counts start getting small

- Good Turing discounting always used in conjunction with backoff or interpolation

Katz Backoff Smoothing

- For a Katz bigram model, let us define:
  - $\Psi(w_{i\text{-}1}) = \{w\text{: } \pi(w_{i\text{-}1}, w) > 0\}$
- A bigram model with Katz smoothing can be written in terms of a unigram model as follows:

$$P_{\mathrm{Katz}}(w_i|w_{i-1}) = \begin{cases} \frac{\pi^*(w_{i-1}, w_i)}{\pi(w_{i-1})} & \text{if } w_i \in \Psi(w_{i-1}) \\ \alpha(w_{i-1}) P_{\mathrm{Katz}}(w_i) & \text{if } w_i \notin \Psi(w_{i-1}) \end{cases}$$

$$\text{where} \quad \alpha(w_{i-1}) = \frac{\left(1 - \sum_{w \in \Psi(w_{i-1})} \frac{\pi^*(w_{i-1}, w)}{\pi(w_{i-1})}\right)}{\sum_{w_i \notin \Psi(w_{i-1})} P_{\mathrm{Katz}}(w_i)}$$

Absolute Discounting Interpolation

- Absolute discounting motivated by Good-Turing estimation
- Just subtract a constant d from the non-zero counts to get the discounted count
- Also involves linear interpolation with lower-order models

$$\Pr_{\text{abs}}(w_i|w_{i-1}) = \frac{\max\{\pi(w_{i-1}, w_i) - d, 0\}}{\pi(w_{i-1})} + \lambda(w_{i-1})\Pr(w_i)$$

- However, interpolation with unigram probabilities has its limitations
- Cue in, Kneser-Ney smoothing that replaces unigram probabilities (how often does the word occur) with continuation probabilities (how often is the word a continuation)

Kneser-Ney discounting

$$\Pr_{\mathrm{KN}}(w_i|w_{i-1}) = \frac{\max\{\pi(w_{i-1}, w_i) - d, 0\}}{\pi(w_{i-1})} + \lambda_{\mathrm{KN}}(w_{i-1})\Pr_{\mathrm{cont}}(w_i)$$

Consider an example: "Today I cooked some yellow <u>curry</u>"

Suppose π(yellow, curry) = 0. $\Pr_{\text{abs}}$[w | yellow] = λ(yellow)Pr(w)

Now, say Pr[Francisco] >> Pr[curry], as San Francisco is very common in our corpus.

But Francisco is not as common a "continuation" (follows only San) as curry is (red curry, chicken curry, potato curry, …)

Moral: Should use probability of being a continuation!

c.f., absolute discounting

$$\Pr_{\mathrm{abs}}(w_i|w_{i-1}) = \frac{\max\{\pi(w_{i-1}, w_i) - d, 0\}}{\pi(w_{i-1})} + \lambda(w_{i-1})\Pr(w_i)$$

Kneser-Ney discounting

$$\Pr_{\mathrm{KN}}(w_i|w_{i-1}) = \frac{\max\{\pi(w_{i-1}, w_i) - d, 0\}}{\pi(w_{i-1})} + \lambda_{\mathrm{KN}}(w_{i-1})\Pr_{\mathrm{cont}}(w_i)$$

$$\Pr_{\mathrm{cont}}(w_i) = \frac{|\Phi(w_i)|}{|B|}$$ and $$\lambda_{\mathrm{KN}}(w_{i-1}) = \frac{d}{\pi(w_{i-1})}|\Psi(w_{i-1})|$$

$$\frac{d \cdot |\Psi(w_{i-1})| \cdot |\Phi(w_i)|}{\pi(w_{i-1}) \cdot |B|}$$

where
$$\Phi(w_i) = \{w_{i-1} : \pi(w_{i-1}, w_i) > 0\}$$
$$B = \{(w_{i-1}, w_i) : \pi(w_{i-1}, w_i) > 0\}$$

c.f., absolute discounting

$$\Pr_{\mathrm{abs}}(w_i|w_{i-1}) = \frac{\max\{\pi(w_{i-1}, w_i) - d, 0\}}{\pi(w_{i-1})} + \lambda(w_{i-1})\Pr(w_i)$$

Midsem Exam

- September 17th, 2019 (Tuesday)
- Time: 8.30 am to 10.30 am
- Venue: CC 101, 103 and 105
- Closed book exam. Will allow 1 A4 (two-sided) sheet of notes.
- Can bring calculators to the exam hall.

Midsem Syllabus

- HMMs (Forward/Viterbi/Baum-Welch (EM) algorithms)
- Tied-state HMM models
- WFST algorithms
- WFSTs in ASR
- Feedforward NN-based acoustic models (Hybrid/Tandem/TDNNs)
- Language modeling (Ngram models + Smoothing techniques)
- There could be (no more than) one question on basic probability

- Topics covered in class that won't appear in the exam:
  - Basics of speech production
  - Role of epsilon filters in composition
  - RNN-based models

Question 1: Phone recogniser

Suppose you are building a simple ASR system which recognizes only four words bowl, bore, pour, poll involving five phones p, b, ow, l, r (with obvious pronunciations for the words). We are given a phone recognizer which converts a spoken word into a sequence of phones, which is known to have the following behaviour:

Phone	p	ow	r	b	l
p	0.8	0	0	0.2	0
ow	0	1	0	0	0
r	0	0	0.6	0	0.4
b	0.2	0	0	0.8	0
l	0	0	0.4	0	0.6

The probability of recognizing a spoken phone x as a phone y is given in the row labeled by x and the column labeled by y. Let us assume a simple language model for our task: Pr(bowl) = 0.1, Pr(bore) = 0.4, Pr(pour) = 0.3 and Pr(poll) = 0.2. Determine the most likely word (and the corresponding probability) given that the output from the phone recognizer is “p ow l”.

Question 2: WFSTs for ASR

Recall the WFST-based framework for ASR that was described in class. Given a test utterance x, let D_x be a WFST over the tropical semiring (with weights specialized to the given utterance) such that decoding the utterance corresponds to finding the shortest path in D_x. Suppose we modify D_x by adding γ (> 0) to each arc in Dx that emits a word. Let's call the resulting WFST D'_x.

A) Describe informally, what effect increasing γ would have on the word sequence obtained by decoding D'_x.

B) Recall that decoding D_x was used as an approximation for \argmax Pr(x|W) Pr(W). What would be the analogous expression for decoding from D'_x?

Question 3: FSTs in ASR

Words in a language can be composed of sub-word units called morphemes. For simplicity, in this problem, we consider there to be three sets of morphemes, V_{pre},V_{stem} and V_{suf} – corresponding to prefixes, stems and suffixes. Further, we will assume that every word consists of a single stem, and zero or more prefixes and suffixes. That is, a word is of the form $w=p_1 \cdots p_k \sigma s_1 \cdots s_l$ where $k,l \geq 0$, and $p_i \in V_{pre}$, $s_i \in V_{suf}$ and $\sigma \in V_{stem}$. For example, a word like fair consists of a single morpheme (a stem), where as the word unfairness is composed of three morphemes, un + fair + ness, which are a prefix, a stem and a suffix, respectively.

A) Suppose we want to build an ASR system for a language using morphemes instead of words as the basic units of language. Which WFST(s) in the $H \circ C \circ L \circ G$ framework should be modified in order to utilize morphemes?

B) Draw an FSA over morphemes ($V_{pre} \cup V_{stem} \cup V_{suf}$) that accepts only words with at most four morphemes. Your FSA should not have more than 15 states. You may draw a single arc labeled with a set to indicate a collection of arcs, each labeled with an element in the set.

Question 4: Probabilities in HMMs

	a	b	c
q_1	0.5	0.3	0.2
q_2	0.3	0.4	0.3
q_3	0.2	0.1	0.7
q_4	0.4	0.5	0.1
q_5	0.3	0.3	0.4
q_6	0.9	0	0.1

Consider the HMM shown in the figure. (The transition probabilities are shown in the finite-state machine and the observation probabilities corresponding to each state are shown on the left.) This model generates hidden state sequences and observation sequences of length 4. If S1, S2, S3, S4 represent the hidden states and O1, O2, O3, O4 represent the observations, then $S_i \in \{q1,...,q6\}$ and $O_i \in \{a,b,c\}$. Pr(S1 = q1) = 1 i.e. the state sequence starts in q1.

State whether the following three statements are true or false and justify your responses. If the statement is false, then state how the left expression is related to the right expression, using either =,< or > operators. (We use the following shorthand in the statements below: Pr(O = abbc) denotes Pr(O1 = a,O2 = b,O3 = b,O4 = c)

A) Pr(O = bbca, S1 = q1, S4 = q6) = Pr(O = bbca | S1 = q1, S4 = q6)

B) Pr(O = acac, S2 = q2, S3 = q5) > Pr(O = acac, S2 = q4, S3 = q3)

C) Pr(O = cbcb | S2 = q2, S3 = q5) = Pr(O = baac, S2 = q4, S3 = q5)

Question 5: HMM training

Suppose we are given N observation sequences, X_i, i = 1 to N where each X_i is a sequence $(x_i^1, \ldots, x_i^{T_i})$ of length T_i where x_i^t is an acoustic vector $\in R^d$. To estimate the parameters of an HMM with Gaussian output probabilities from this data, the Baum-Welch EM algorithm uses empirical estimates $\xi_{i,t}(s, s')$ for the probability of being in state s at time t and s' at time t + 1 given the observation sequence X_i and $\gamma_{i,t}(s)$ for the probability of occupying state s at time t given X_i.

In a variant of EM known as Viterbi training, for each i, one computes the single most likely state sequence $S_i^1, \ldots, S_i^{T_i}$ for X_i by Viterbi decoding, and defines $\xi_{i,t}$ and $\gamma_{i,t}$ assuming that X_i was produced deterministically by this path. Give the expressions for $\xi_{i,t}(s, s')$ and $\gamma_{i,t}(s)$ in this case.