

1ª parte: Diseño de sistemas de ASR basados en HMM

2ª parte: Entrenamiento de modelos HMM de densidad continua

**Rita Singh**

Facultad de Informática  
Carnegie Mellon University

Diseño de estructuras gráficas para el lenguaje HMM:  
Comprensión y tratamiento de la complejidad  
introducida por el uso de unidades de subpalabra

# Creación de modelos HMM para secuencias de palabras : unidades

---

- ◆ Los sistemas de enormes vocabularios no emplean palabras como unidades de sonido
  - El vocabulario consiste en decenas de miles de palabras
  - Es difícil hallar suficientes ejemplos de cada palabra incluso en enormes corpus de entrenamiento
  - Las palabras no tratadas durante el entrenamiento nunca serán aprendidas ni reconocidas
  
- ◆ En cambio, las palabras se descomponen en unidades de subpalabra
  - En un corpus existen muchos más ejemplos de unidades de subpalabra que de palabras, los parámetros del HMM se pueden estimar mejor
  - Las unidades de subpalabra se pueden combinar de varios modos para formar nuevas palabras que pueden reconocerse
    - ▶ No es necesario ver todas las palabras del vocabulario durante el entrenamiento
  - Normalmente motivadas fonéticamente y conocidas por tanto como fonos

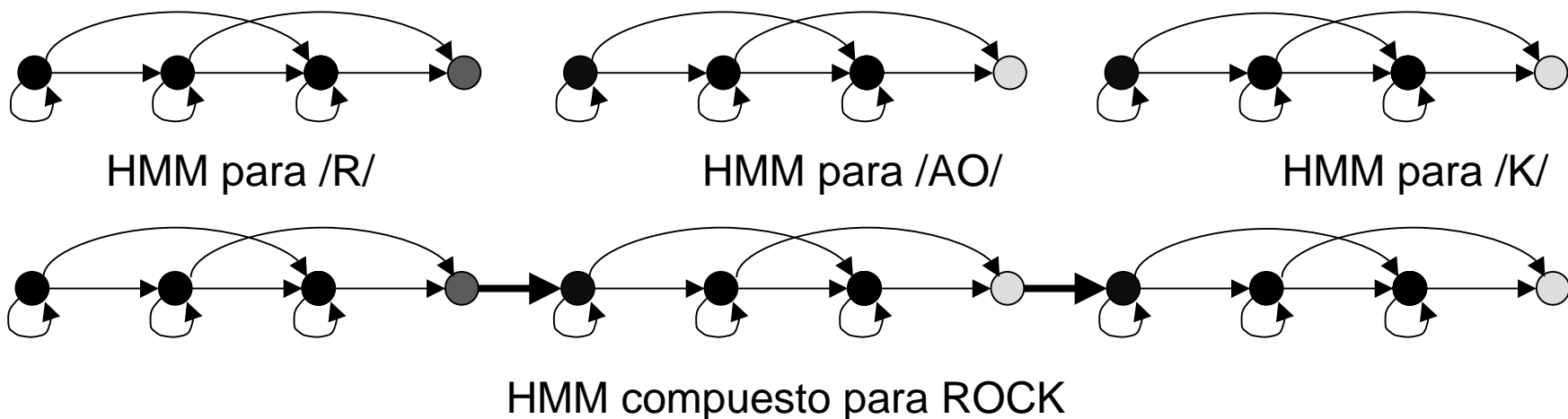
# Creación de modelos HMM para secuencias de palabras: Unidades independientes del contexto

*Ejemplo:*

<u>Palabra</u>	<u>Fonos</u>
Rock	R AO K

*Las unidades son independientes del contexto porque la identidad de los fonemas no depende de sus vecinos*

- ◆ Cada palabra se expresa como una secuencia de unidades de subpalabra
- ◆ Cada unidad de subpalabra es modelada por un HMM
- ◆ Los HMM de las palabras se construyen al concatenar los HMM de unidades de subpalabra
- ◆ El componer los HMM de la palabra con unidades independientes del contexto no aumenta la complejidad del lenguaje HMM



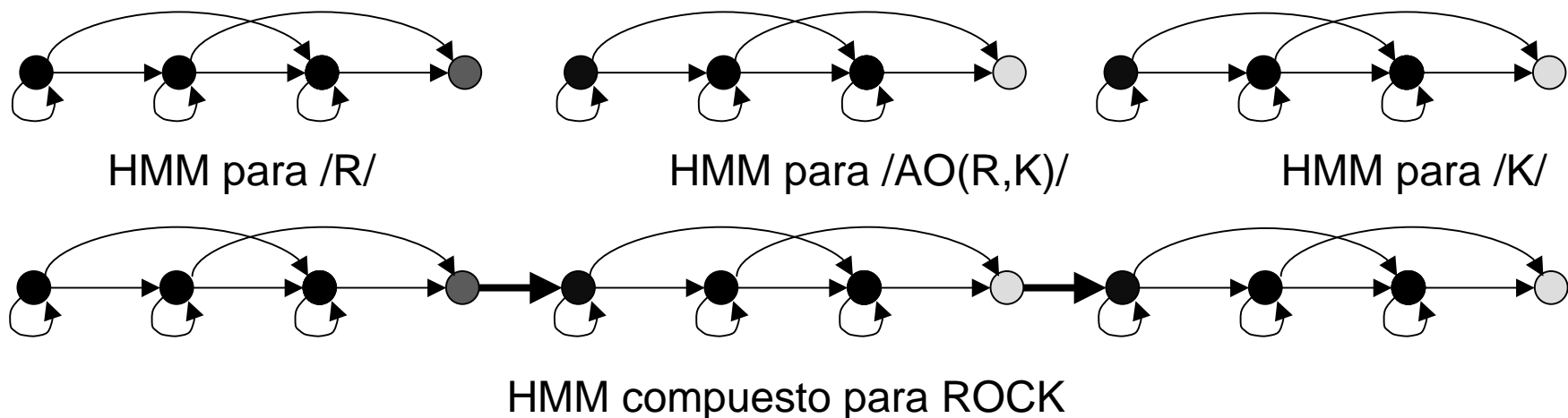
# Creación de HMM para secuencias de palabras: Unidades dependientes del contexto internas a la palabra

*Ejemplo:*

Palabra	Fonos	Trifonos
Rock	R AO K	R,AO(R,K),K

*La unidad de subpalabra A, O (R,K) está relacionada con los contextos R y K.  
Es dependiente del contexto*

- ◆ Los fonemas son unidades ordinarias
  - Cuando /AO/ va precedida por R y seguida por K, desde un punto de vista espectrográfico, es distinta a cuando va precedida por /B/ y seguida por /L/
- ◆ Los trifonos son unidades fonéticas *en contexto*
- ◆ Si los trifonos se utilizaran sólo dentro de palabras, y las unidades utilizadas al final de palabra fuesen independientes del contexto, la complejidad del lenguaje HMM sería la misma que cuando todas las unidades son independientes del contexto

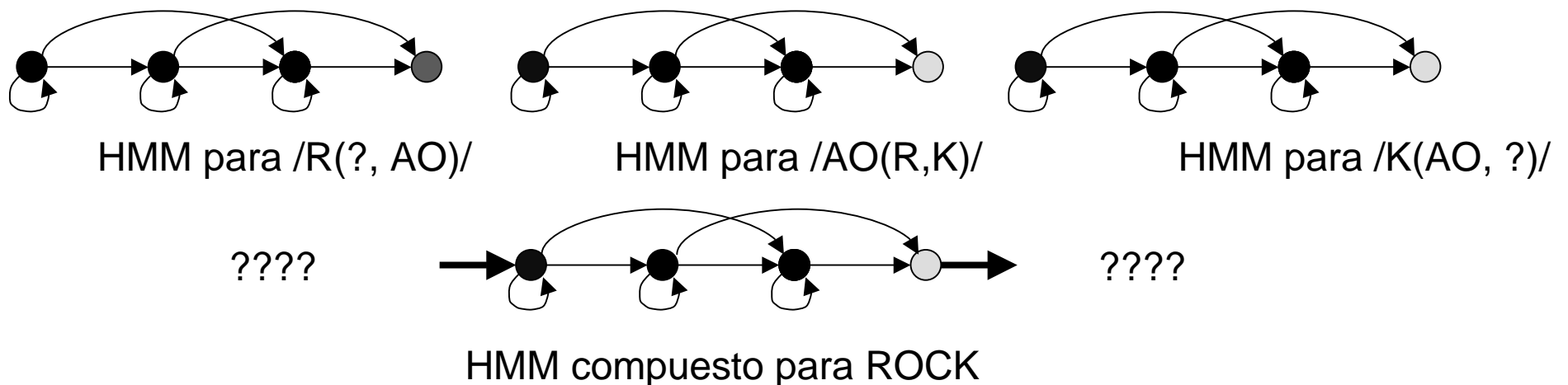


# Creación de modelos HMM para secuencias de palabra: Unidades dependientes del contexto entre palabras

*Ejemplo:*

Palabra	Fonos	Trifonos
Rock	R AO K	R(*,AO), AO(R,K),K(AO, *)

- ◆ Cuando los trifonos se utilizan en límites de palabra, los HMM empleados para componer la palabra se convierten en dependientes de palabras adyacentes
  - Si “Rock” fuera seguido por “STAR S T AA R”, el trifono final para ROCK sería K(AO,S)
  - Si “Rock” fuera seguido por “MUSIC M Y UW Z I K”, el trifono final en ROCK sería K(AO, M)



# Construcción del HMM de una oración utilizando unidades de subpalabra

---

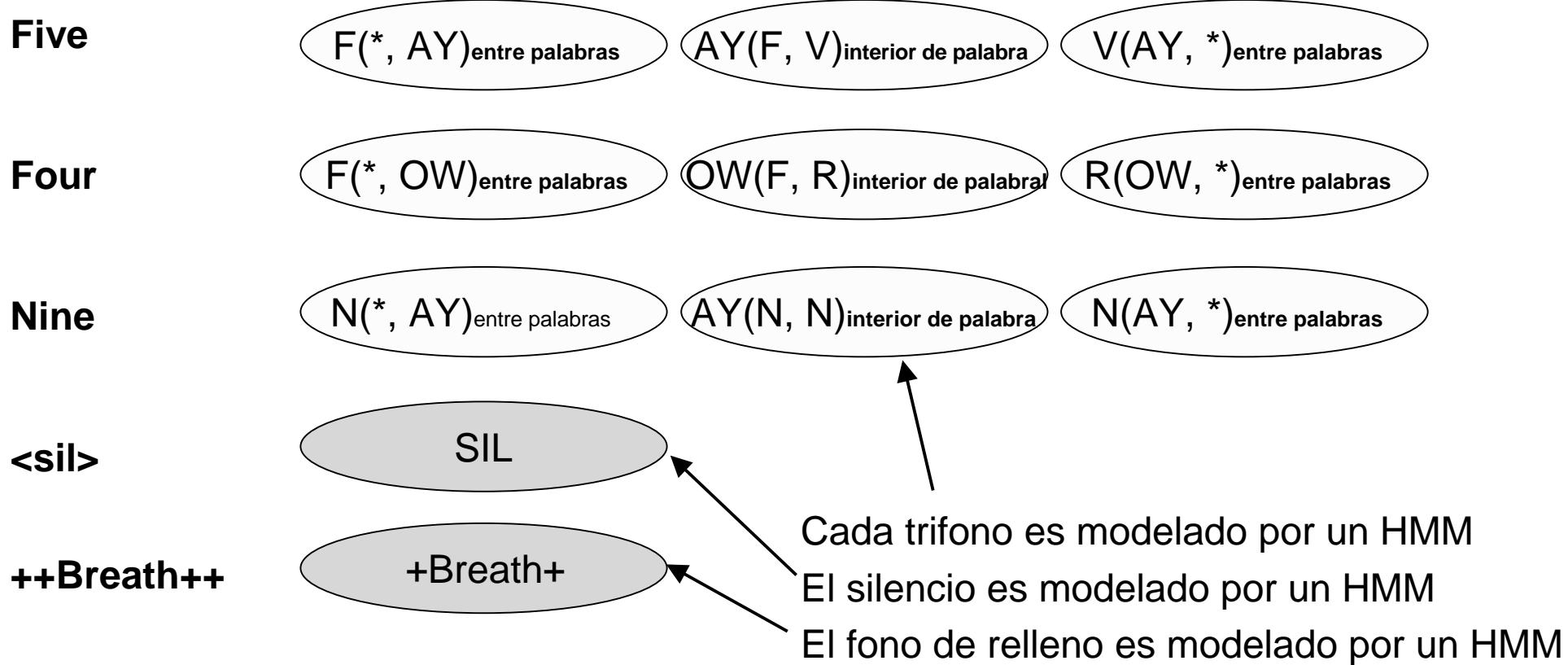
## Diccionario

Five:	<b>F AY V</b>
Four:	<b>F OW R</b>
Nine:	<b>N AY N</b>
<sil>:	<b>SIL (silencio)</b>
++breath++:	<b>+breath+ (respiración)</b>

Tenemos aquí enumeradas cinco “palabras” y sus pronunciaciones en cuanto a "fonos". Supongamos que estas son las únicas palabras del discurso actual que van a ser reconocidas. Digamos entonces que el vocabulario de reconocimiento consta de cinco palabras. El sistema utiliza un diccionario como referencia para estas asociaciones.

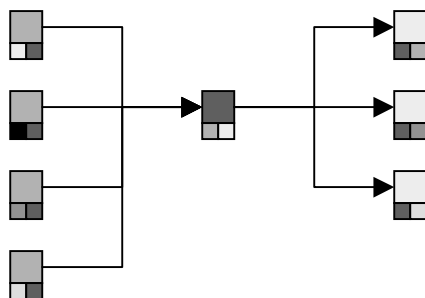
# Construcción del HMM de una oración utilizando unidades de subpalabra

Utilizando el diccionario como referencia, el sistema primero asocia cada palabra con pronunciaci3nes basadas en trifonos. Cada trifono adem1s presenta una etiqueta o tipo caracter1stico, seg1n el lugar de aparici3n en la palabra. El contexto no es inicialmente conocido para los trifonos que se dan entre palabras.



# Construcción del HMM de una oración para un UNIGRAMA de trifonos

HMM para "Four"  
Esto se compone de 8 HMM.



Cada recuadro triple representa un trifono. Cada modelo de trifono es de hecho un HMM de izquierda a derecha (podría tener cualquier número de estados. Cada estado es un senone)

## Léxico

Four			
Five			
Nine			
<sil>			
++Breath++			

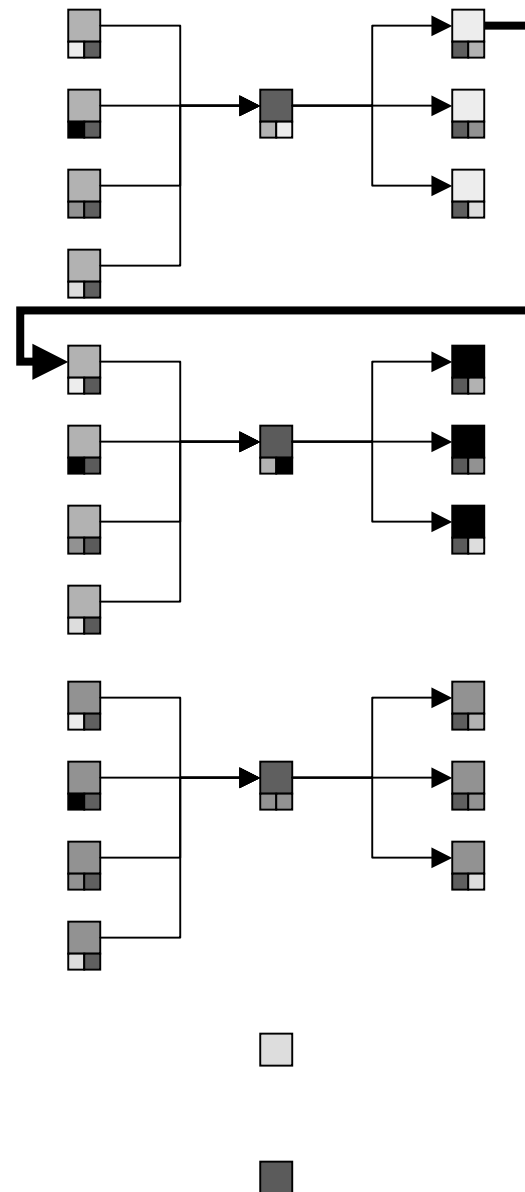
	= F
	= OW
	= R
	= AY
	= V
	= N
	= SIL
	= +breath

Un trifono es un fono simple con **INFORMACIÓN** de contexto. No es una secuencia literal de 3 fonos.

## Expandir la palabra *Four*

- Todos los últimos fonos (excepto el de relleno) se convierten en contextos a la izquierda para el primer fono de *Four*
- Todos los fonos primeros (excepto el de relleno) se convierten en contextos derechos para el último fono de *Four*
- El silencio puede formar contextos, pero por sí mismo no presenta ninguna dependencia de contexto
- Los fonos de relleno (ej. +breath+) son tratados como silencio al construir contextos. Como el silencio, ellos por sí mismos no presentan ninguna dependencia de contexto

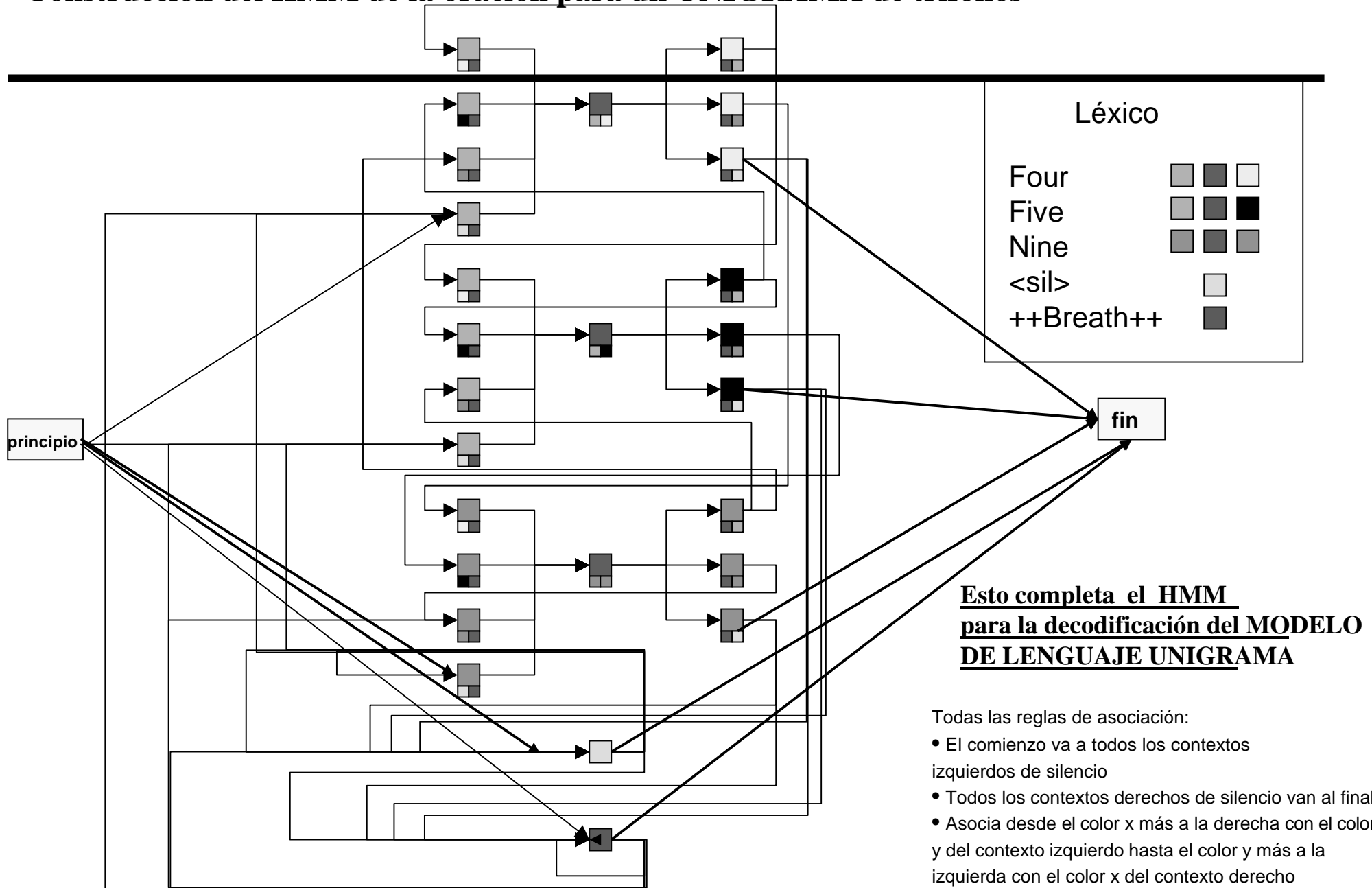
# Construcción del HMM de una oración para un UNIGRAMA de trifonos



Léxico	
Four	
Five	
Nine	
<sil>	
++breath++	

Regla de asociación:  
 Asocia desde el color x más a la derecha con el color de contexto derecho y, hasta el color más a la izquierda y con el color de contexto derecho x

# Construcción del HMM de la oración para un UNIGRAMA de trifonos



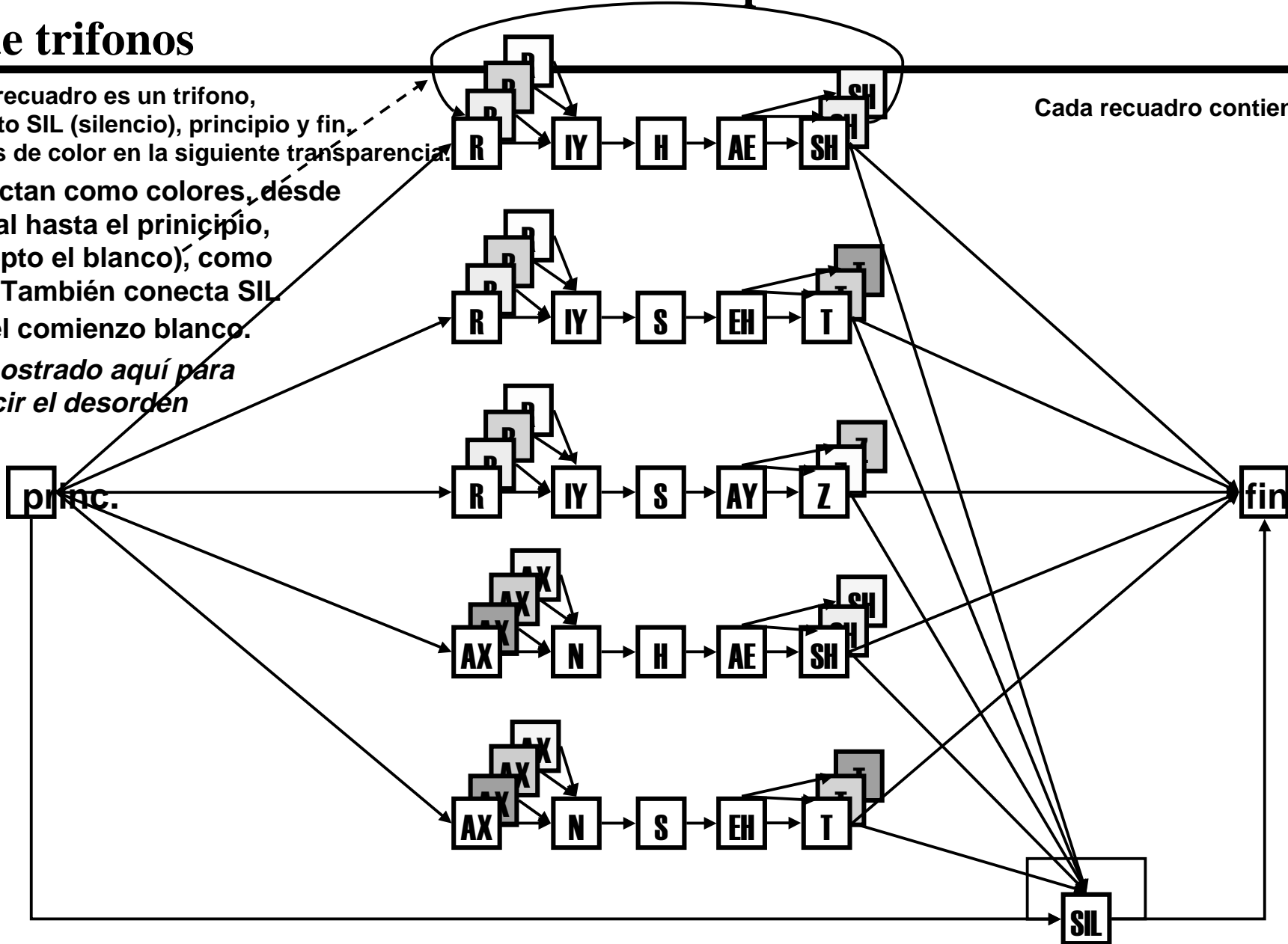
# Construcción del HMM de la oración para un UNIGRAMA PLANO de trifonos

Cada recuadro es un trifono, excepto SIL (silencio), principio y fin. Claves de color en la siguiente transparencia.

Conectan como colores, desde el final hasta el principio, (excepto el blanco), como este. También conecta SIL con el comienzo blanco.

*No mostrado aquí para reducir el desorden*

Cada recuadro contiene un HMM



# Construcción del HMM de la oración para un UNIGRAMA de trifonos: clave de color de la transparencia anterior

---

AX(SH,N)	R(SH,IY)	T(AE,R)	Z(AE,R)	SH(AE,R)
AX(T,N)	R(T,IY)	T(AE,AX)	Z(AE,AX)	SH(AE,AX)
AX(Z,N)	R(Z,IY)	T(AE,SIL)	Z(AE,SIL)	SH(AE,SIL)
AX(SIL,N)	R(SIL,IY)			

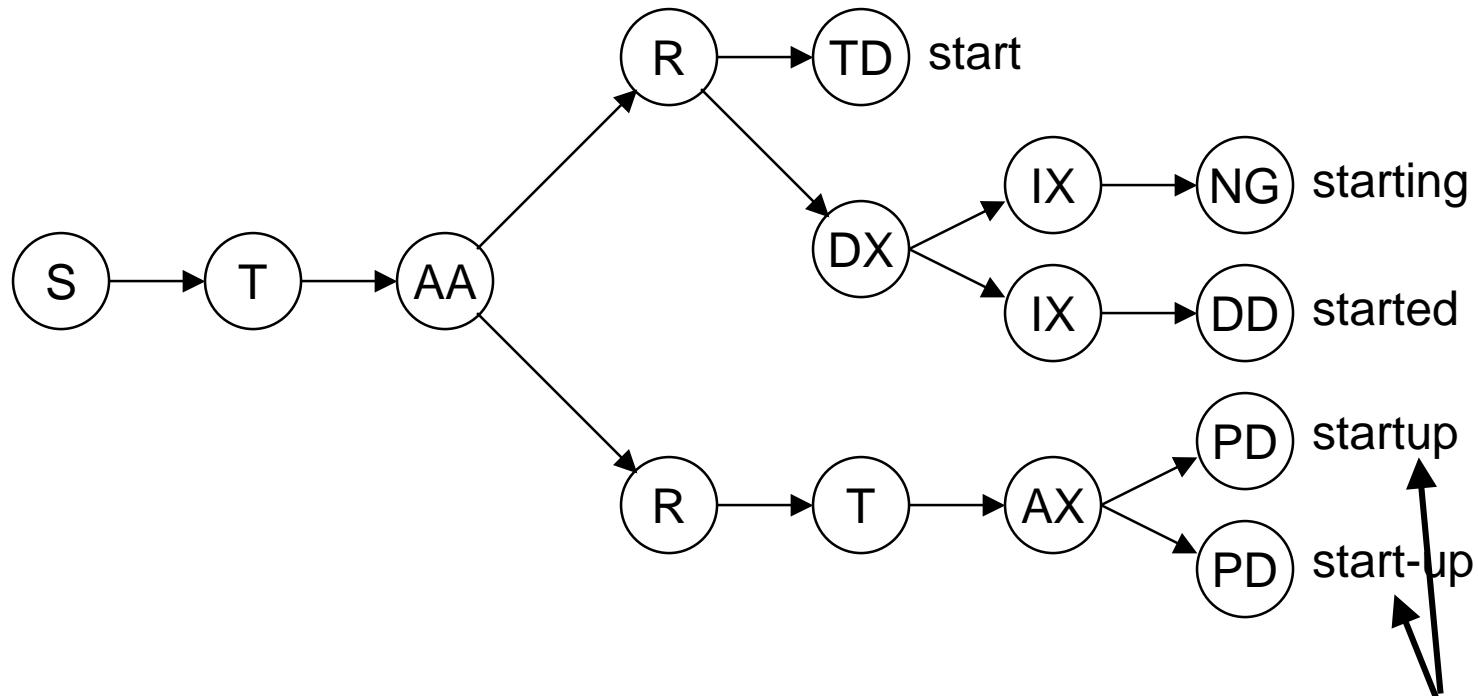
Rehash	R IY H AE SH
Reset	R IY S EH T
Resize	R IY S AY Z
Unhash	AX N H AE SH
Unset	AX N S EH T
<sil>	SIL

# Simplificación de la decodificación

---

- ◆ El ejemplo que hemos visto es de decodificación de búsqueda PLANA con estructura de lenguaje unigrama
  - La estructura del vocabulario es plano: cada palabra posee su propia representación
- ◆ El HMM de la oración para grafos de tipo de búsqueda plana, basados en el modelo de lenguaje bigrama y trigrama, puede llegar a ser grande y complicado
- ◆ Reducir el tamaño del HMM de la oración es una cuestión importante de ingeniería, a la hora de diseñar un decodificador
- ◆ La búsqueda PLANA es exacta, pero de memoria intensiva y lenta

# Árbol léxico



**Palabras distintas  
con pronunciaci3nes id3nticas  
deben tener distintos nodos  
terminales**

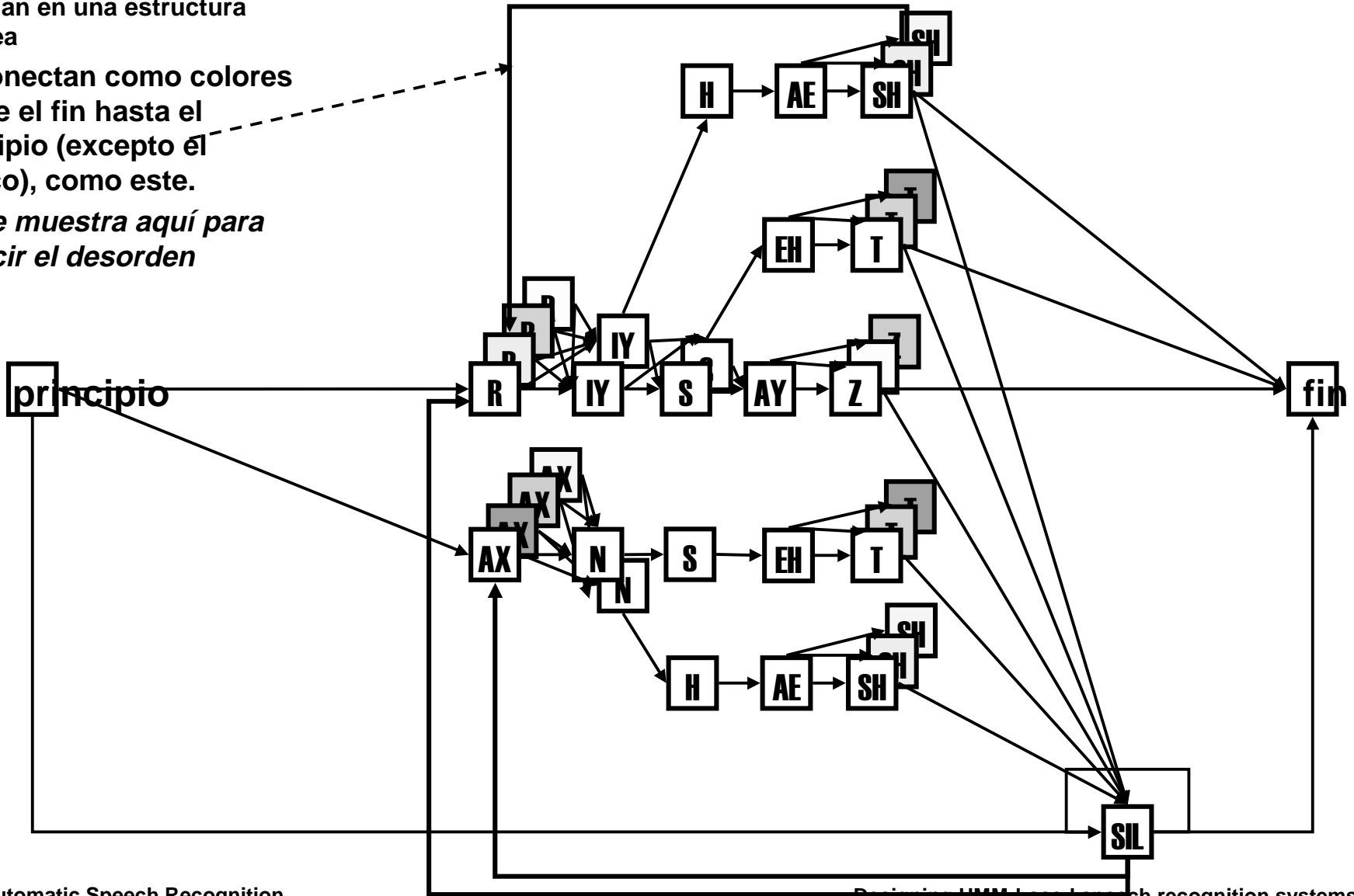
- ◆ Las palabras comparten HMM de fonos (o trifonos). Se emplea la similitud fon3tica para reducir el tama1o y las necesidades de memoria, y disminuir la computaci3n para aumentar la velocidad de decodificaci3n

# Construcción del HMM de la oración para un *ÁRBOL LÉXICO UNIGRAMA* de trifonos

En un árbol léxico, los fonos se fusionan en una estructura arbórea

Se conectan como colores desde el fin hasta el principio (excepto el blanco), como este.

*No se muestra aquí para reducir el desorden*

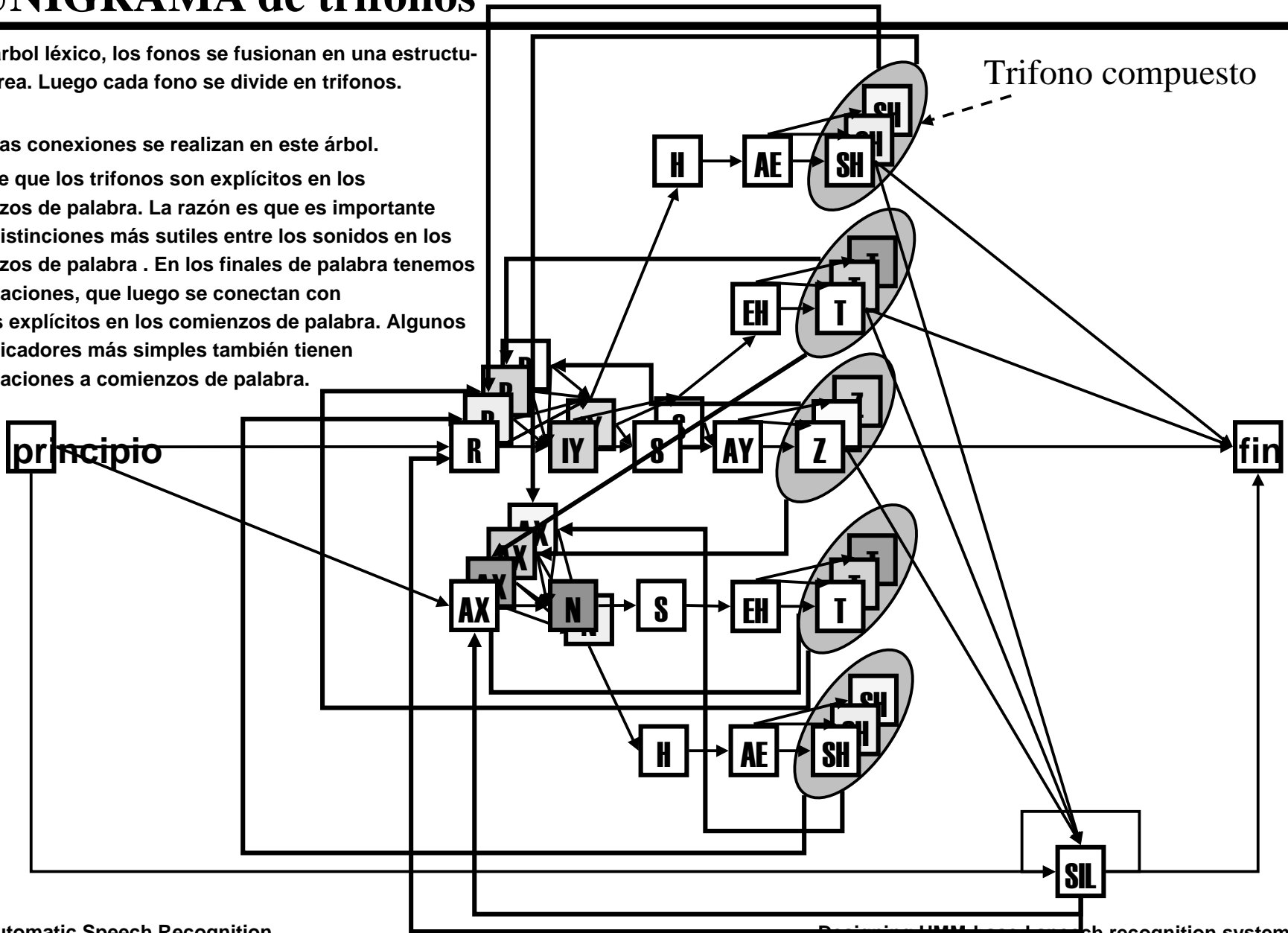


# Construcción del HMM de la oración para un *ÁRBOL LÉXICO UNIGRAMA* de trifonos

En un árbol léxico, los fonos se fusionan en una estructura arbórea. Luego cada fono se divide en trifonos.

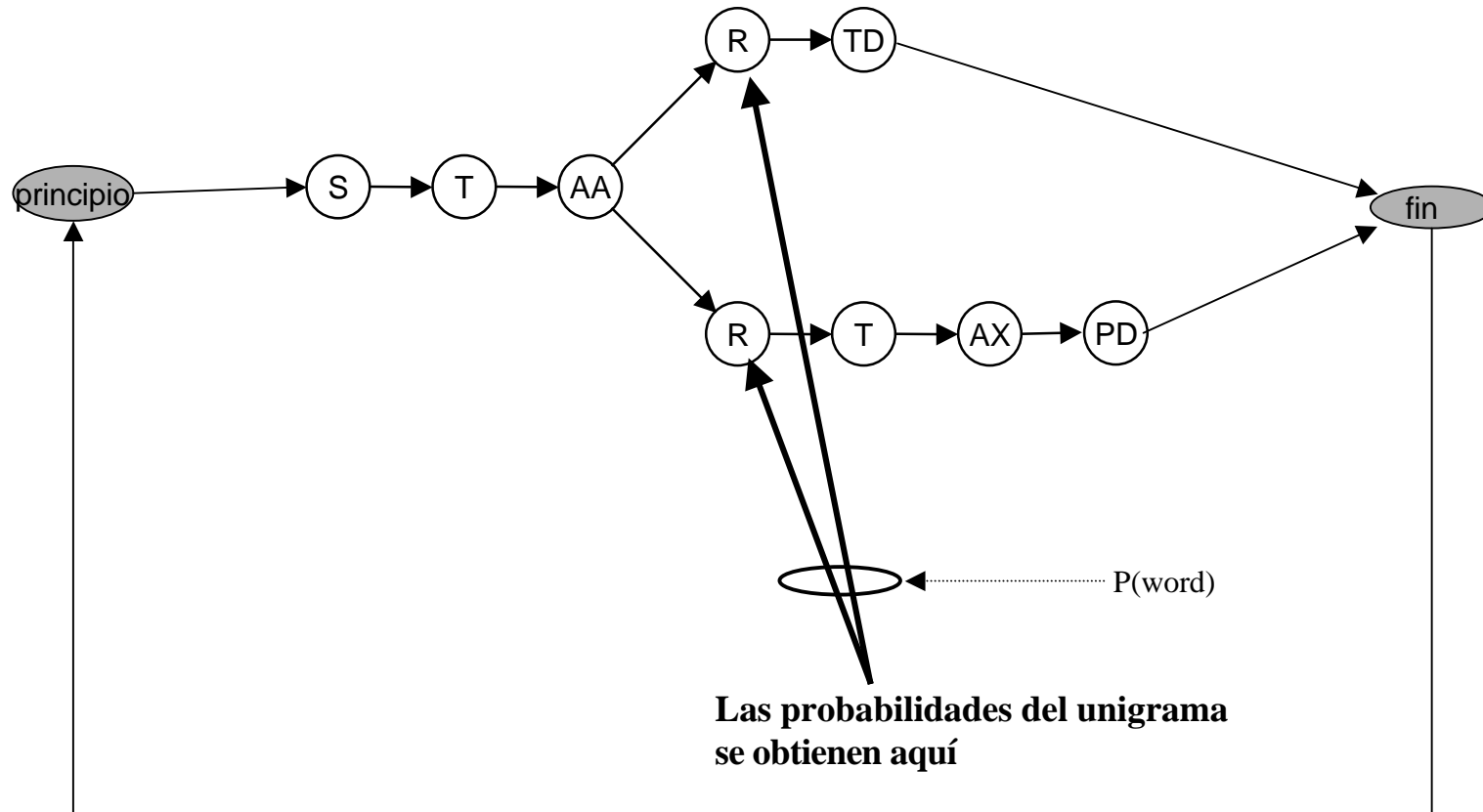
Todas las conexiones se realizan en este árbol.

Observe que los trifonos son explícitos en los comienzos de palabra. La razón es que es importante hacer distinciones más sutiles entre los sonidos en los comienzos de palabra. En los finales de palabra tenemos combinaciones, que luego se conectan con trifonos explícitos en los comienzos de palabra. Algunos decodificadores más simples también tienen combinaciones a comienzos de palabra.



# Decodificación de un árbol léxico unigrama

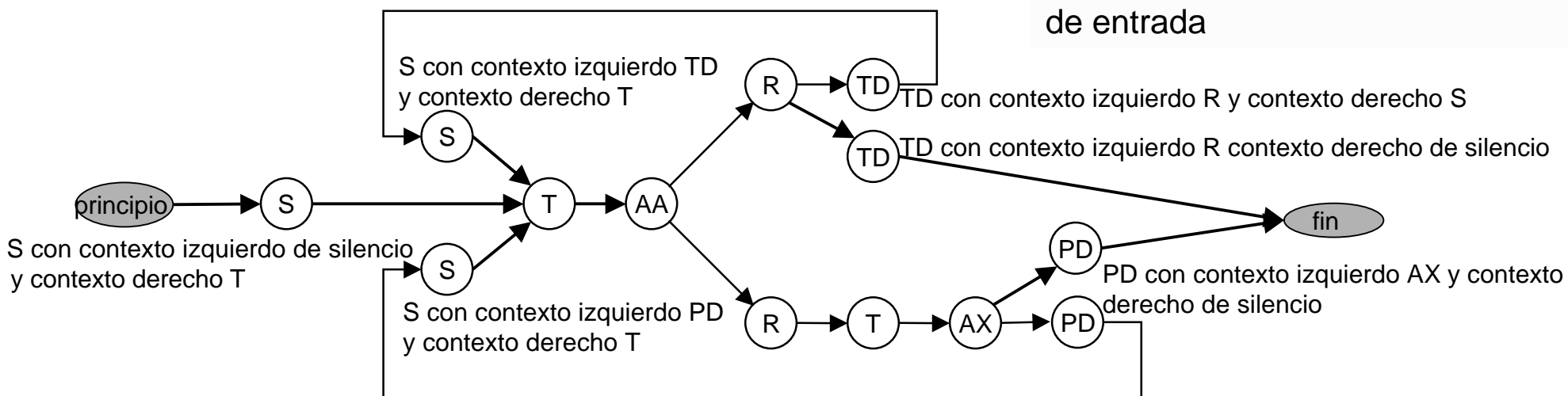
La figura es conceptual. Una figura más exacta para árboles léxicos basados en trifonos tendría en cuenta los contextos trifonéticos



# Decodificación de un árbol léxico unigrama (Más precisión)

Árbol léxico detallado basado en trifono para este ejemplo

Si hubieran múltiples fonos de entrada, existirían múltiples copias de TD y PD en la salida, una para cada posible fono de entrada



Un árbol léxico unigrama posee varios puntos de entrada, uno por cada posible fono precedente

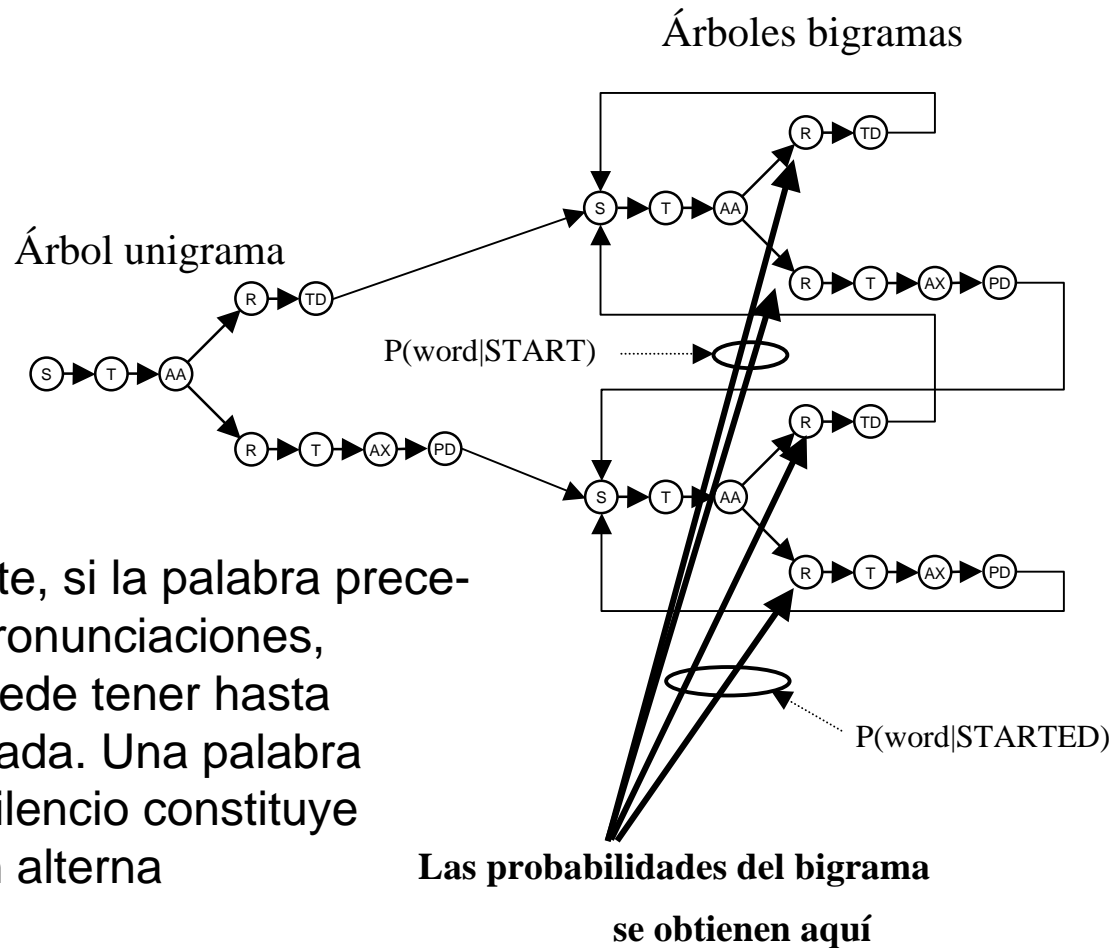
Aquí los posibles fonos precedentes son TD, PD y silencio (al "comienzo")

Existen dos modelos de trifonos para S, uno con contexto izquierdo TD, y el otro con contexto izquierdo PD

Observe que el modelo es un árbol sólo a partir del segundo fono

# Decodificación del árbol léxico bigrama

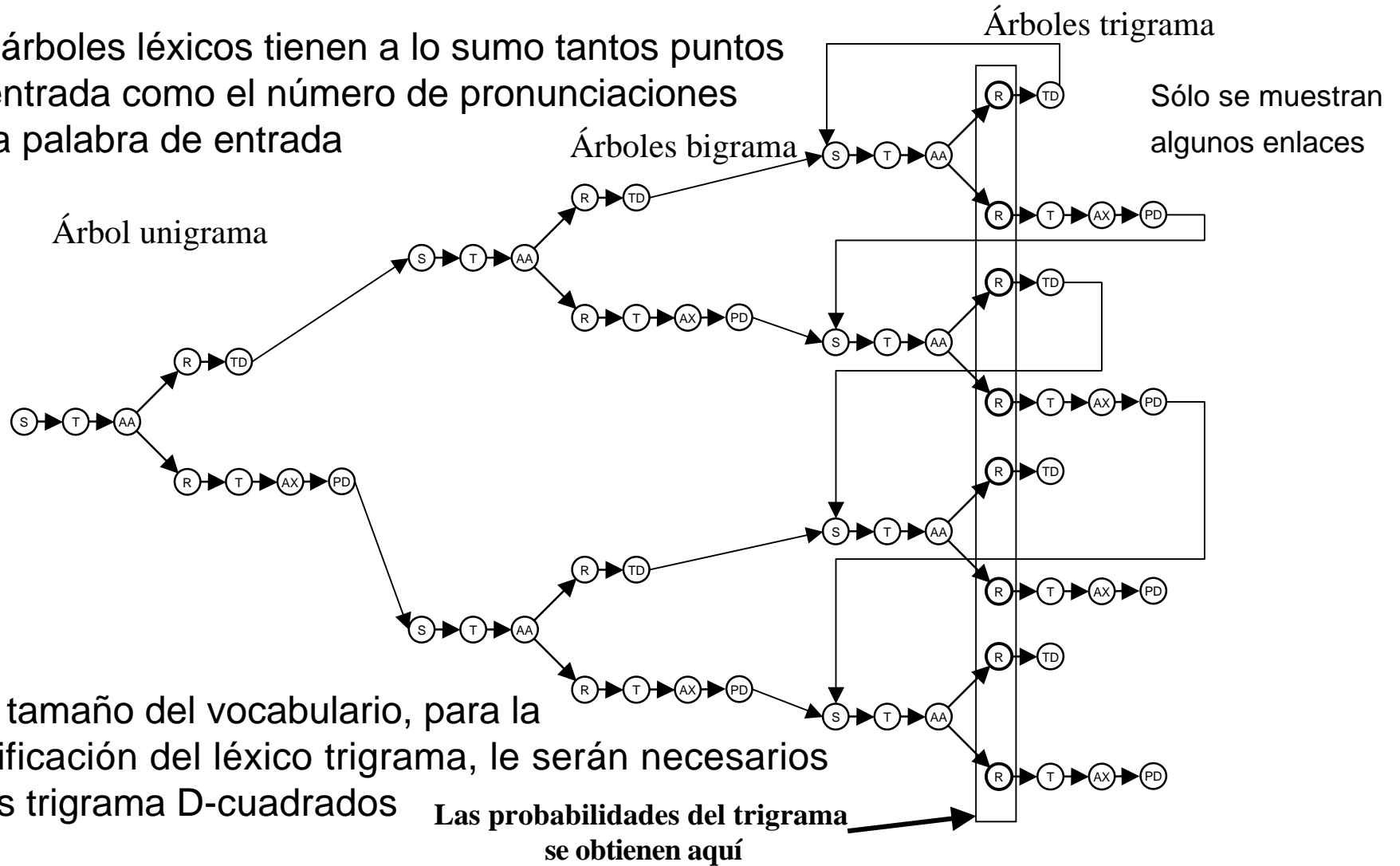
Si todas las palabras tienen sólo una única pronunciación, todos los árboles léxicos presentan sólo un único punto de entrada, ya que únicamente pueden introducirse desde una palabra específica



Más generalmente, si la palabra precedente posee  $N$  pronunciaciones, el árbol léxico puede tener hasta  $N$  puntos de entrada. Una palabra seguida por un silencio constituye una pronunciación alterna para la palabra

# Decodificación del árbol léxico trigrama

Los árboles léxicos tienen a lo sumo tantos puntos de entrada como el número de pronunciaciones de la palabra de entrada



# Cuestiones sobre árboles léxicos

---

- ◆ Las identidades de la palabra no se conocen en la entrada. Esto complica la estructura del lenguaje HMM incluso más que en la búsqueda plana
  - Los lenguajes HMM basados en árbol léxico *crecen* en realidad mucho más que los correspondientes a los HMM planos, en todos excepto en el caso del unigrama
    - Un HMM plano que incorpora probabilidades Ngrama y posee un vocabulario de  $D$  palabras, requiere los HMM de  $D^{N-1} + D^{N-2} + \dots + D$  palabras. Un HMM de árbol léxico para el mismo vocabulario, requiere  $D^{N-1} + D^{N-2} + \dots + D$  árboles léxicos
  - El número de transiciones entre el estado del HMM de la oración es proporcionalmente más grande
  - Se proponen varias heurísticas para corregir este problema

# Reducción del tamaño de árboles léxicos mediante la incorporación de N-gramas: Estructuras de decodificación aproximadas

---

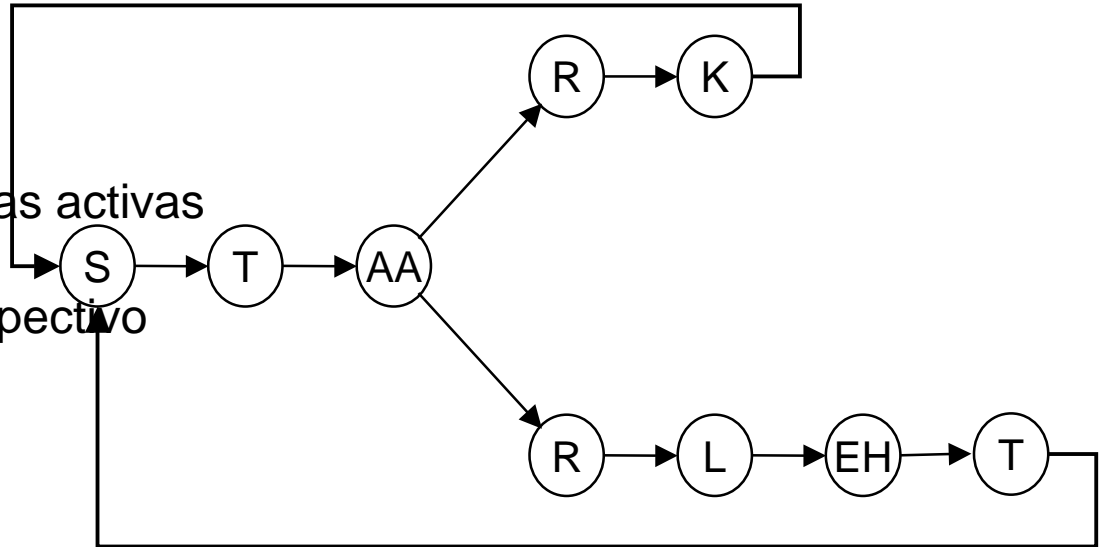
- ◆ Árboles léxicos de tamaño reducido
  - Decodificación de N-grama con un árbol léxico simple
  - Decodificación de N-grama con árboles léxicos cambiantes
  
- ◆ Efectos sobre el reconocimiento
  - La estructura HMM soporta restricciones lingüísticas más débiles
  - El reconocimiento es subóptimo, pero menos intensivo en cuanto a memoria

# Estructuras de decodificación aproximadas

## Decodificación de árbol léxico simple

---

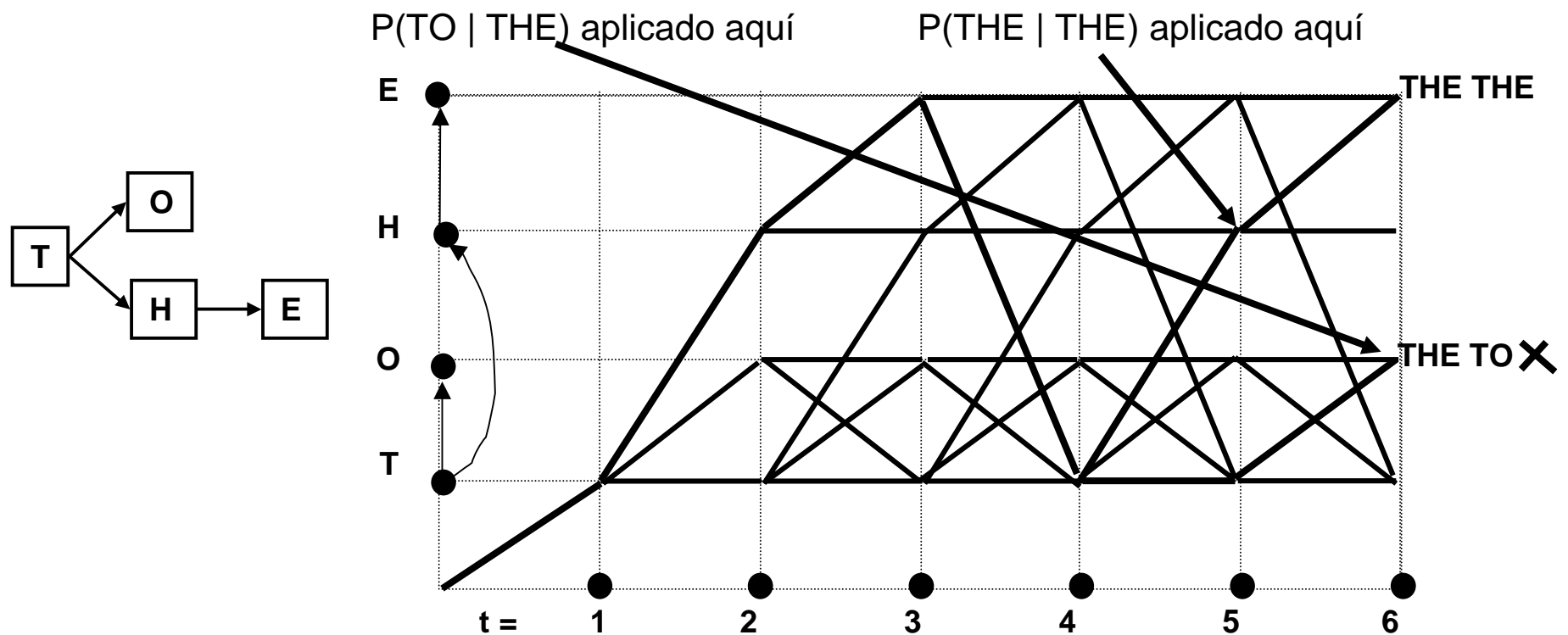
Las historias de la palabra para palabras activas en cada instante en el tiempo almacenado en una tabla con indicador retrospectivo



# La decodificación con un árbol léxico simple introduce errores

## ◆ Ejemplo con los HMM de dos estados simplificados

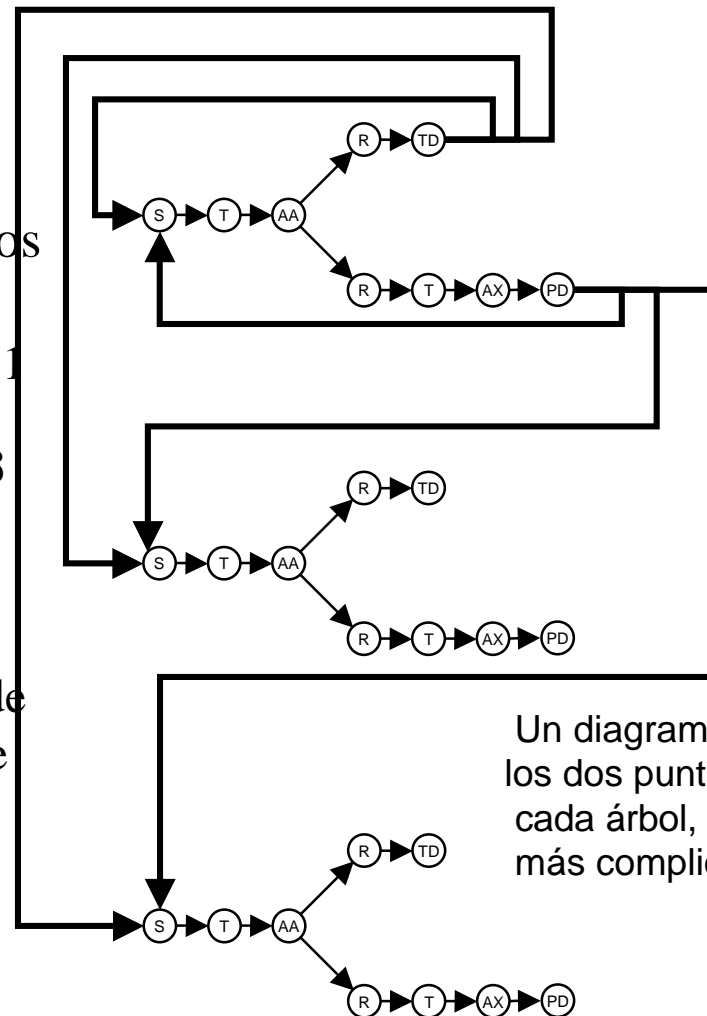
- ◆  $P(x1..x3, THE) > P(x1..x3, TO)$
- ◆  $P(x1..x3,THE)*P(x4..x6,THE | THE) < P(x1..x3,TO)*P(x4..x6,THE | TO)$
- ◆ Sin embargo,  $P(x4..x6,THE | TO) = P(x4..x6 | THE)*P(THE|TO)$  nunca puede computarse ya que TO nunca se tiene en cuenta como contexto
- ◆ Aunque matemáticamente TO THE debe ganar, aquí sóloTHE THE puede plantearse como hipótesis



# Estructuras de decodificación aproximadas

## Árbol léxico cambiante con 3 árboles léxicos: Multiplexado en tiempo

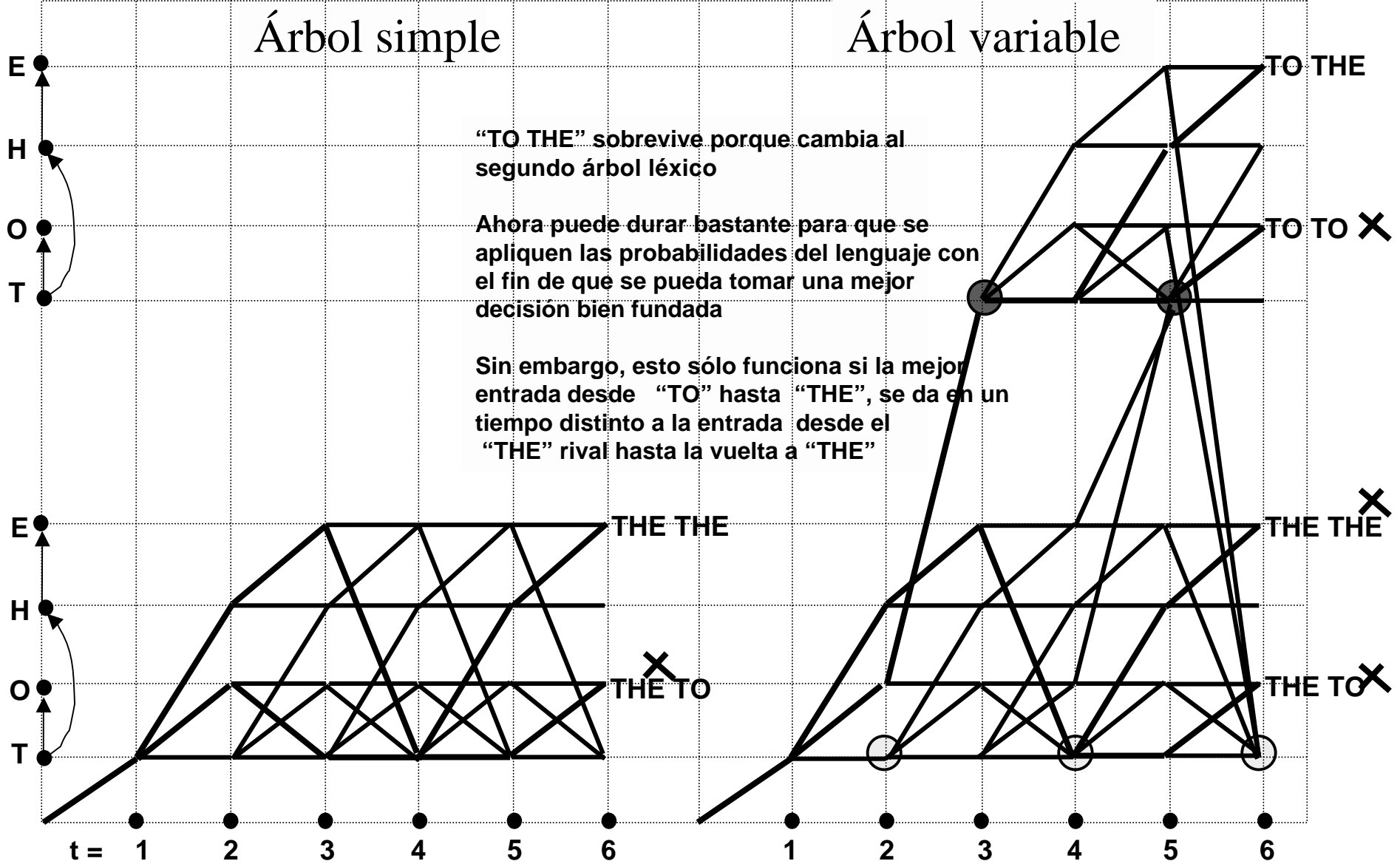
- ◆ Los tres árboles léxicos conectados de forma parecida
- ◆ Puntos de entrada al árbol escalonados en el tiempo
  - Ej. se puede cambiar al árbol léxico 1 sólo en  $t=1,4,7,\dots$ , al árbol léxico 2 sólo en  $t=2,5,8,\dots$ , y al árbol léxico 3 sólo en  $t=3,6,9,\dots$
- ◆ Contextos de N-grama necesarios para las probabilidades de la palabra a partir de una tabla con indicador retrospectivo que mantiene la historia de Viterbi de cualquier camino



Un diagrama detallado mostraría los dos puntos de entrada para cada árbol, además de una figura más complicada

Cada árbol léxico puede tener muchos puntos de entrada

# Árboles léxicos variables



# Reducción del tamaño de los HMM planos: Estructuras de decodificación aproximadas

---

- ◆ Emplean estructuras de HMM de Ngrama de orden inferior para realizar el reconocimiento mediante el uso de probabilidades de Ngrama de orden superior
  - Decodificación de Ngrama a partir de estructuras de unigrama
  - Decodificación de Ngrama a partir de estructuras de bigrama (búsqueda pseudo-trigrama)
  - Empleo de una tabla con indicador retrospectivo para obtener la historia de la palabra
  
- ◆ Efecto sobre el reconocimiento
  - Aplicación imprecisa de probabilidades de Ngrama de orden superior
  - Reducción de los requisitos de memoria

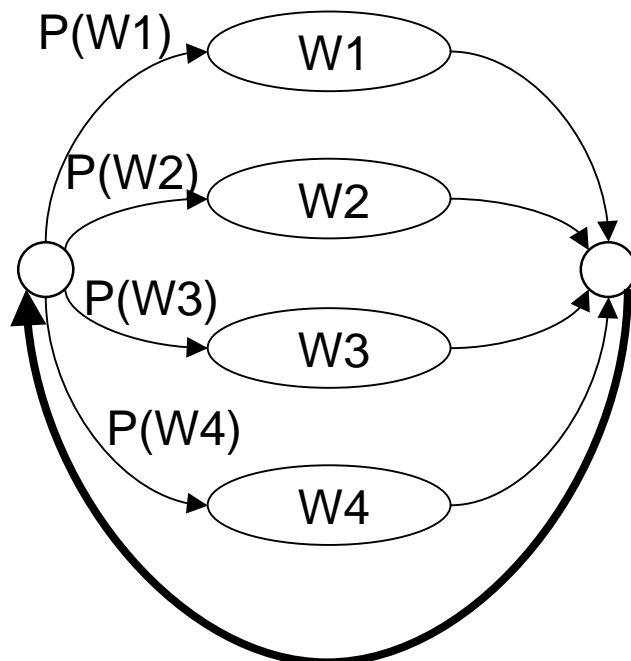
# Estructuras de decodificación aproximadas

Decodificación de pseudo-bigrama a partir de estructura de unigrama en búsqueda plana

---

- ◆ Emplean una estructura de unigrama simple
- ◆ Aplican probabilidades de bigrama
  - El contexto para el bigrama se obtiene a partir de la historia de la palabra
  - La estructura más simple requiere menos memoria y computación
  - Imprecisión

## Unigrama convencional

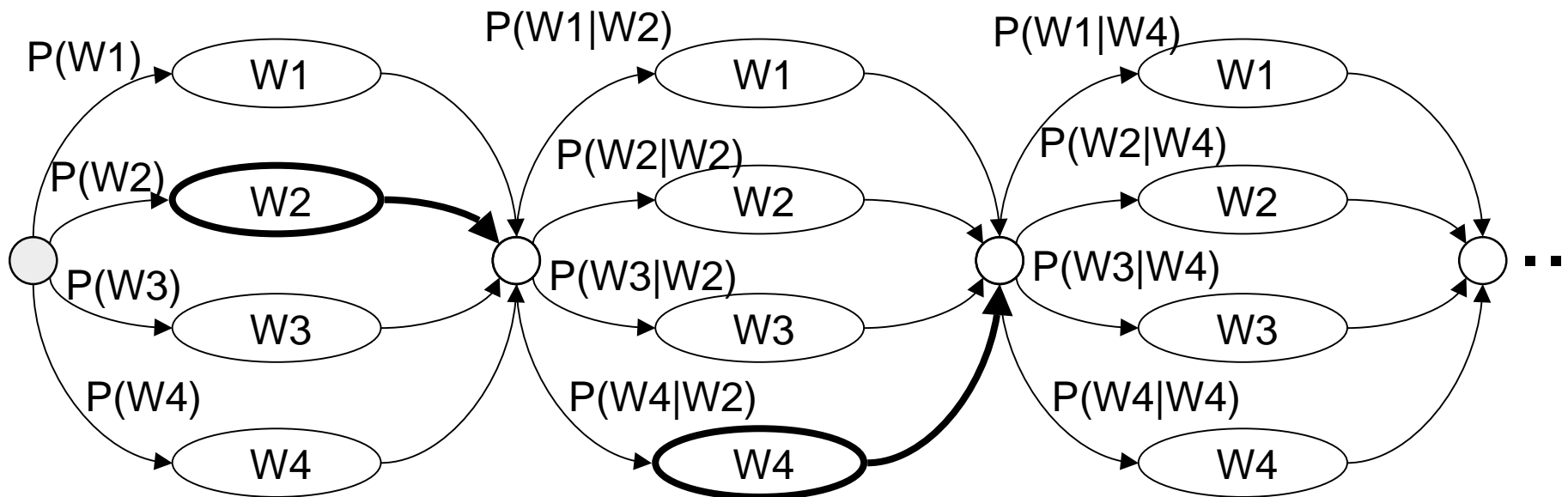


# Estructuras de decodificación aproximadas

## Decodificación de pseudo-bigrama a partir de estructura de unigrama en búsqueda plana

- ◆ Empleo de una estructura de unigrama simple
- ◆ Aplicación de probabilidades de bigrama
  - El contexto para el bigrama se obtiene a partir de la historia de la palabra
  - La estructura más simple requiere menos memoria y computación
  - Imprecisión

### Pseudo-bigrama a partir de estructura de unigrama

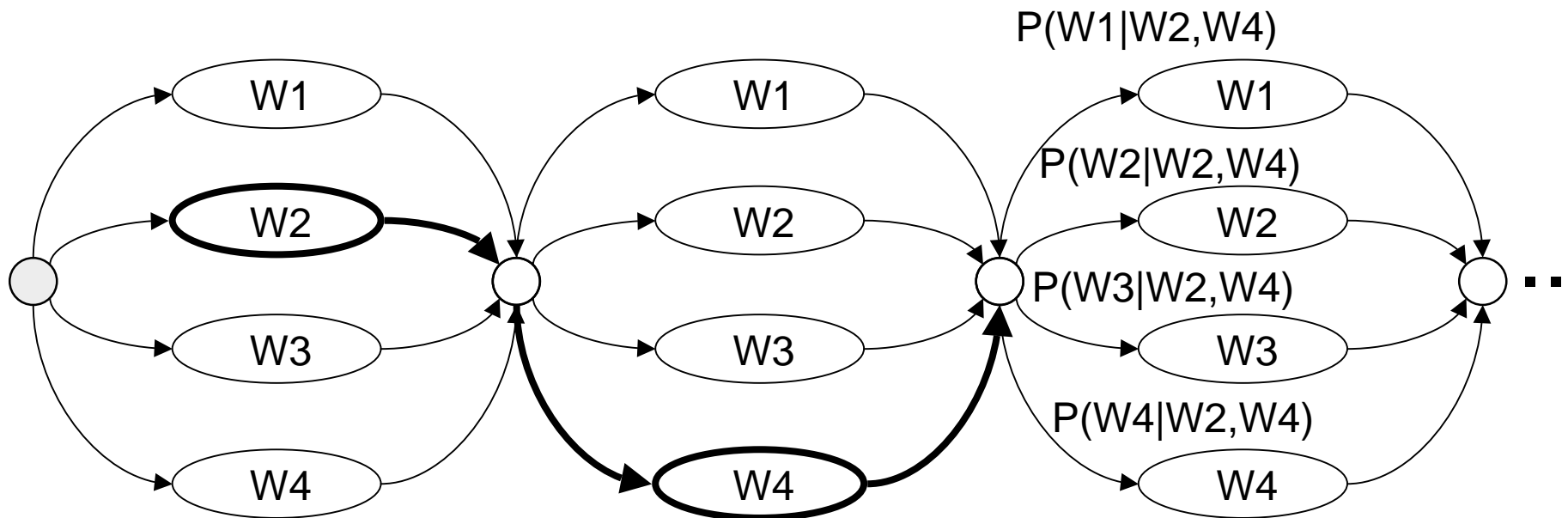


# Estructuras de decodificación aproximadas

## Decodificación de pseudo-trigrama a partir de estructura de unigrama en búsqueda plana

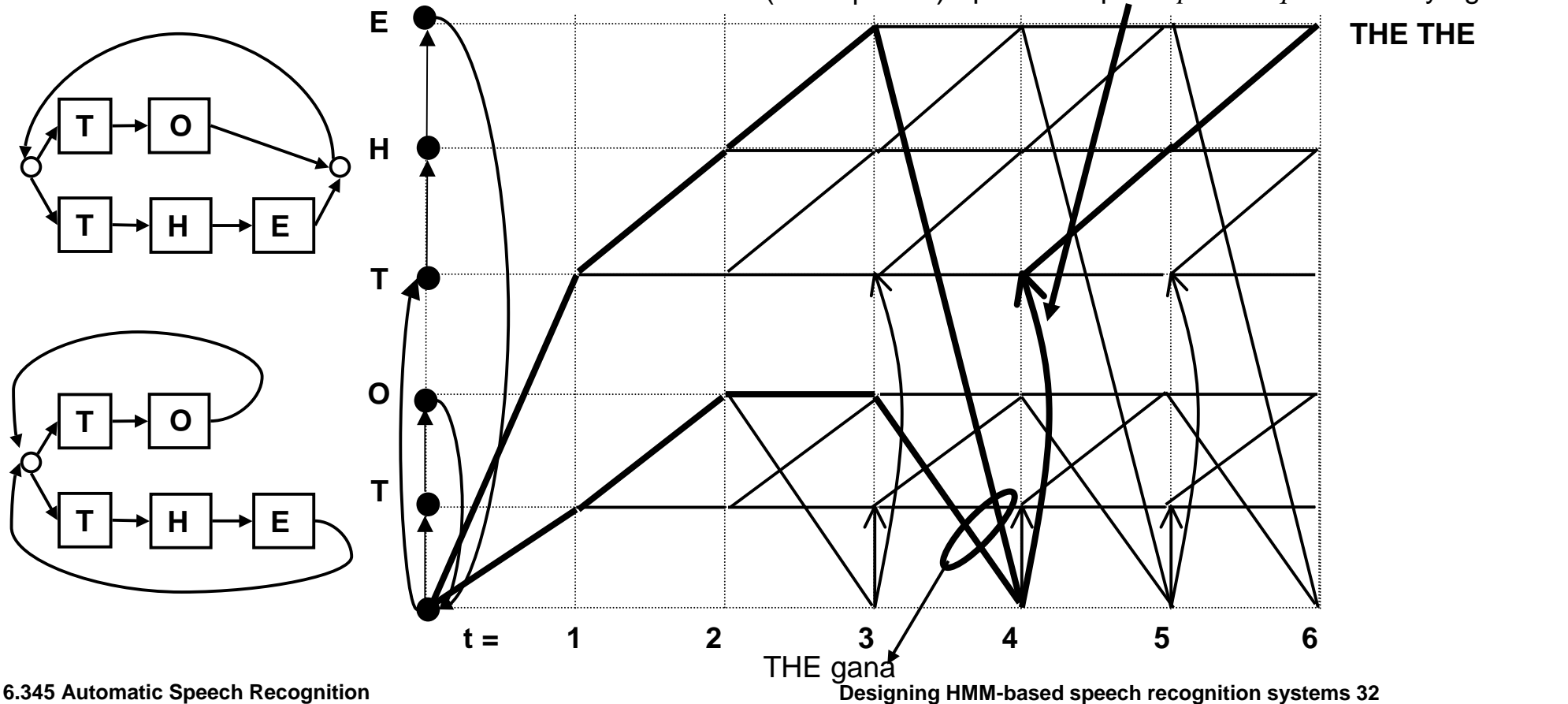
- ◆ Utilización de una estructura de unigrama simple
- ◆ Aplicación de probabilidades de bigrama
  - El contexto para el bigrama se obtiene a partir de la historia de la palabra
  - La estructura más simple requiere menos memoria y computación
  - Imprecisión

### Pseudo-trigrama a partir de estructura de unigrama



# Decodificar con estructura de unigrama presenta errores

- ◆ Ejemplo con HMM simplificados
  - ◆ En  $t=4$ , THE compite con TO y gana. TO no se considera ya una primera palabra candidata en este camino
  - ◆ La competición entre THE y TO se da antes de que la probabilidad  $P$  del bigrama (THE|contexto) sea aplicada
  - ◆  $P(\text{THE}|\text{TO})$  puede haber sido más alta que  $P(\text{THE}|\text{THE})$  y cambió de decisión en  $t=4$
  - ◆ No obstante, la palabra futura no se conoce en el nodo no emite en  $t=4$ . Las probabilidades del bigrama no podrían ser aplicadas

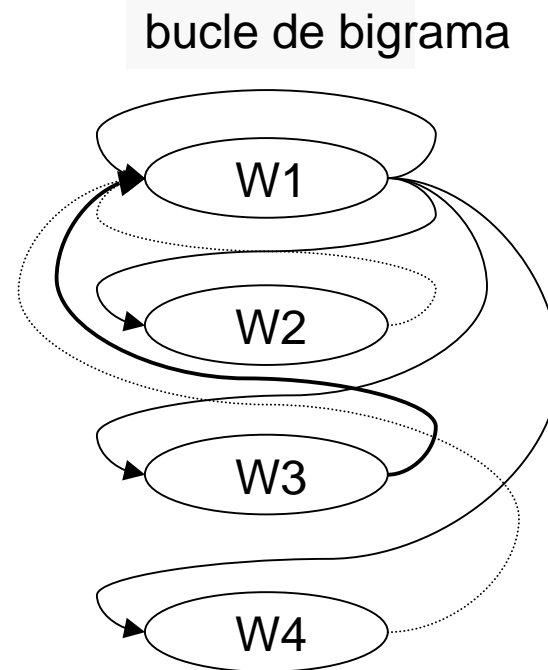


# Estructuras de decodificación aproximada

## Decodificación mediante la estructura del bigrama

---

En vez de una estructura de unigrama, se utiliza una estructura de bigrama. Esto es preciso para los bigramas pero aproximado para la decodificación con trigramas.

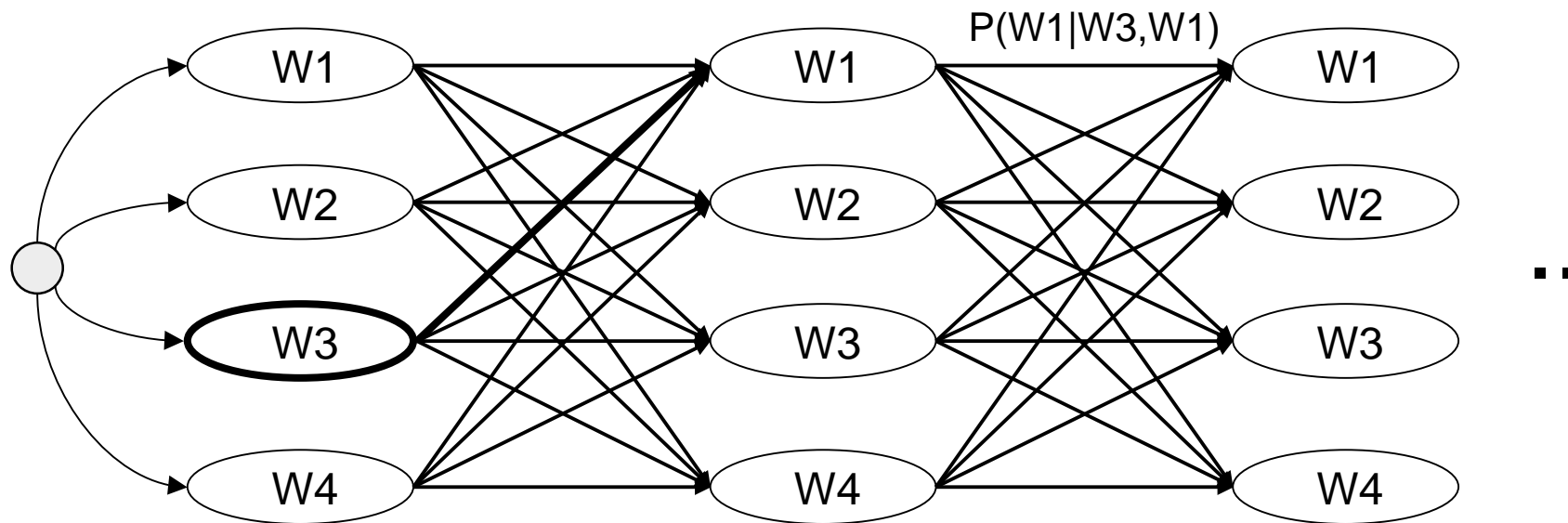


# Estructuras de decodificación aproximada

## Decodificación de pseudo-trigrama para la estructura del bigrama en búsqueda plana

- ◆ Utilización de una estructura de bigrama
- ◆ Aplicación de probabilidades de trigrama
  - El contexto para el trigrama se obtiene a partir de la historia de la palabra
  - Nuevamente, esto es poco preciso

### Bigrama convencional desplegado



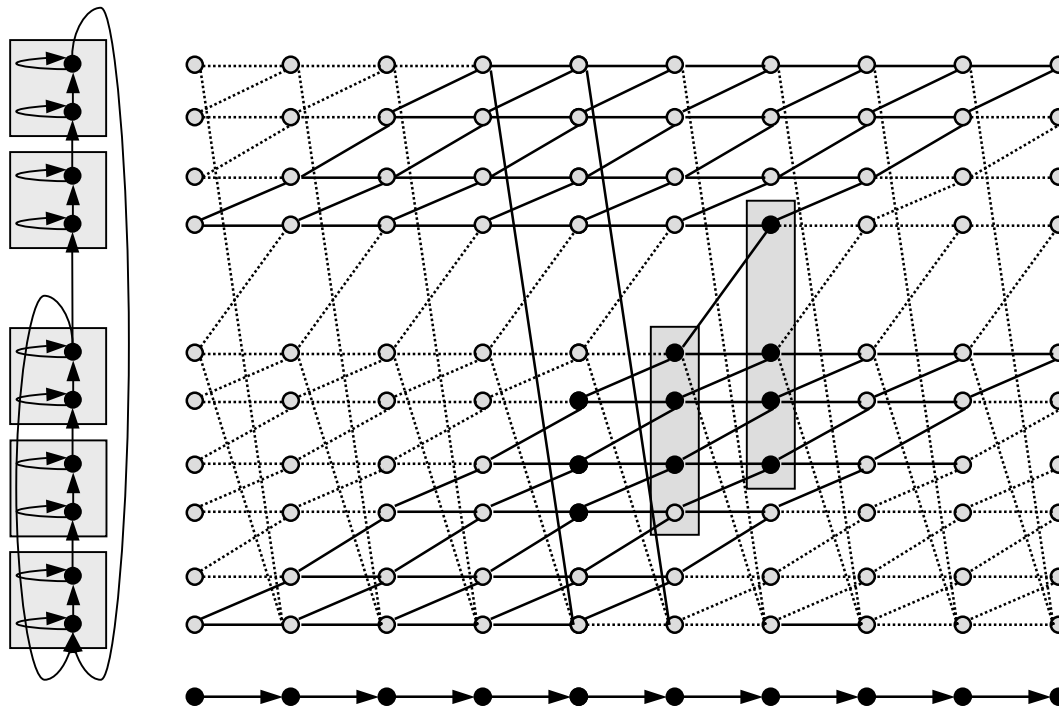
# Reducción máxima de la computación

---

- ◆ Las estructuras aproximadas aún son grandes
  - La búsqueda exhaustiva de todos los caminos a través de ellas es prohibitiva
- ◆ La búsqueda debe ser más restringida
  - Búsqueda en haz (Técnica de recorte de caminos)
- ◆ La computación debe restringirse
  - Selección gaussiana

# Restricción de la búsqueda por memoria y velocidad: Búsqueda en haz

- ◆ En cualquier instante en el tiempo, los caminos que logran más del umbral de puntuación, sobreviven. El umbral de puntuación puede ser fijo (búsqueda en haz fija), o con relación al camino de puntuación más alta en ese instante en el tiempo (búsqueda en haz relativa). Por tanto, la búsqueda en haz implica el recorte de los caminos de puntuación baja.
- ◆ Los nodos que pueden sobrevivir en cualquier tiempo componen la lista activa. Observe que cada nodo es un estado HMM. Las listas activas no se generan siempre por comparaciones de puntuación directas (que resulta lento). Se utilizan muchos otros métodos



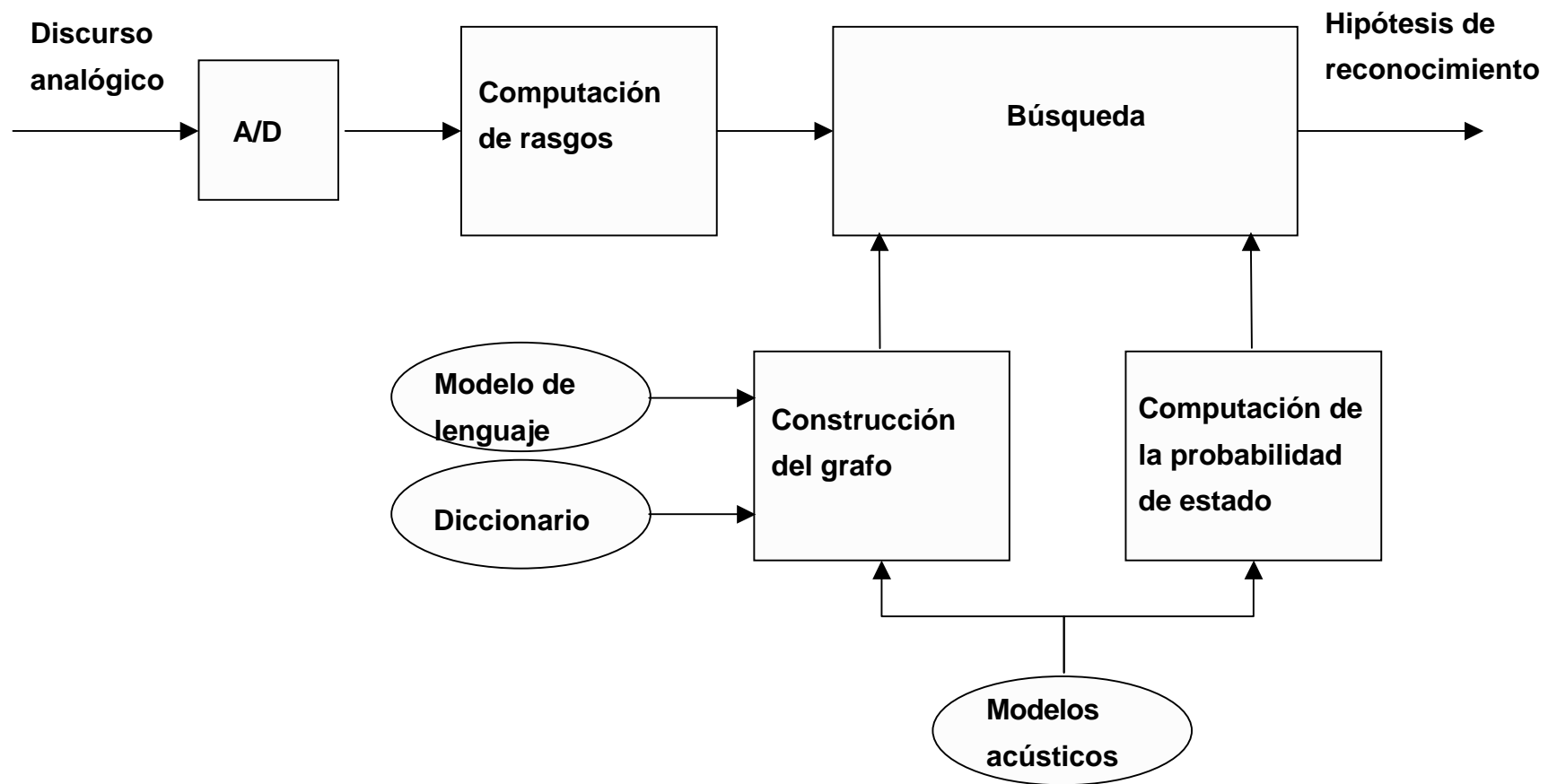
Búsqueda en haz  
relativa

# Restricción de la computación: Selección gaussiana

---

- ◆ Las densidades de probabilidad de estado son típicamente mezclas de gaussianas
- ◆ La computación explícita de probabilidades desde todas las gaussianas presentes en la lista activa es costosa. Se selecciona en primer lugar un subconjunto de estas gaussianas, basado en algún algoritmo de selección gaussiano, y sólo esas gaussianas se computan entonces explícitamente
- ◆ Los algoritmos de selección gaussiana se basan en:
  - Predicción
  - Distribución de densidades de estado
    - agrupamiento
  - Pre-cálculo de puntuaciones aproximadas a partir de un libro de código para la identificación rápida de las mejores gaussianas
    - Cuantización subvectorial

# Arquitectura global del decodificador



# Resumen y conclusiones

---

- ◆ Hemos tratado cuestiones básicas de decodificación
- ◆ Hemos tratado la construcción de los lenguajes HMM para la decodificación
  - La dependencia del grafo del lenguaje esperado
  - Modelos estadísticos de Ngrama y gramáticas de estado finito
- ◆ Hemos discutido algunas cuestiones relativas al tamaño del grafo, necesidades de memoria y requisitos computacionales
- ◆ Esto debería bastarle para comprender el funcionamiento de la mayoría de los sistemas de reconocimiento de voz basados en HMM
  - Y puede que hasta usted se atreva a escribir uno sencillo por sí mismo

# PARTE II

Entrenamiento de modelos HMM de densidad continua

# Índice de contenidos

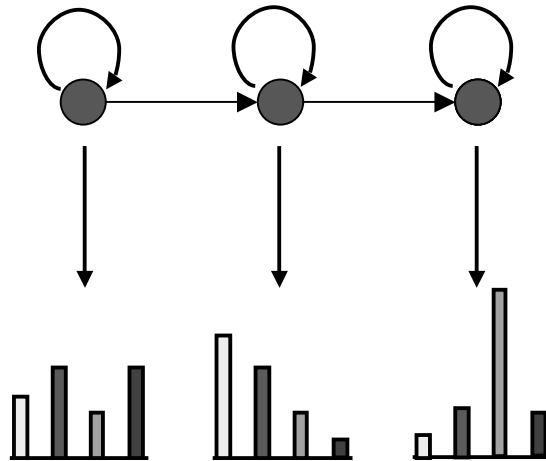
---

- ◆ Repaso de los HMM de densidad continua
- ◆ Entrenamiento de unidades de subpalabras independientes del contexto
  - Esquema
  - Entrenamiento de Viterbi
  - Entrenamiento de Baum-Welch
- ◆ Entrenamiento de unidades de subpalabras dependientes del contexto
  - Enlace de estados
  - Baum-Welch para parámetros compartidos

# HMM discreto

---

- ◆ Los datos sólo pueden tomar un conjunto finito de valores
  - Bolas de una urna
  - Las caras de un dado
  - Valores de un libro de código
- ◆ La distribución de salida del estado de cualquier estado es un histograma normalizado
- ◆ Cada estado tiene su propia distribución



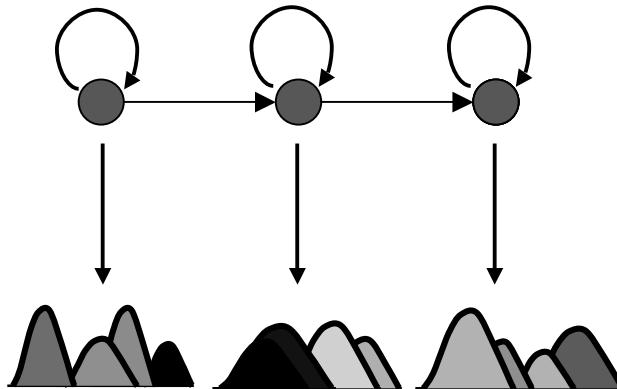
HMM que produce uno de los cuatro colores a cada instante. Cada uno de los tres estados tiene una distribución de probabilidad distinta para los colores.

Dado que el número de colores es discreto, las distribuciones de salida del estado son multinomiales.

# HMM de densidad continua

---

- ◆ Estos datos pueden tomar un continuo de valores
  - *ej.* vectores cepstrales
- ◆ Cada estado posee una *densidad* de salida de estado
- ◆ Cuando el proceso visita un estado, traza un vector desde la densidad de salida del estado para ese estado



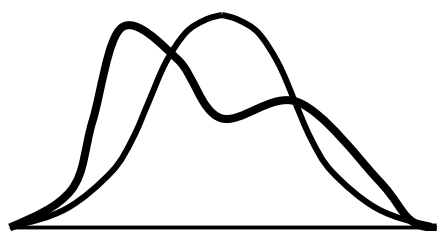
El HMM produce una variable aleatoria continua valorada en cada estado.

Las densidades de salida del estado son mezclas de gaussianas. La salida en cada estado viene trazada desde esta mezcla.

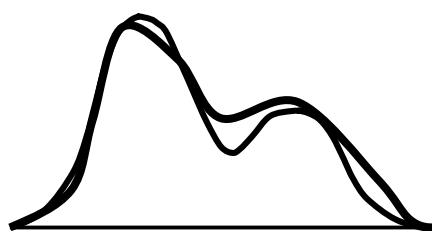
# Modelado de las densidades de salida del estado

---

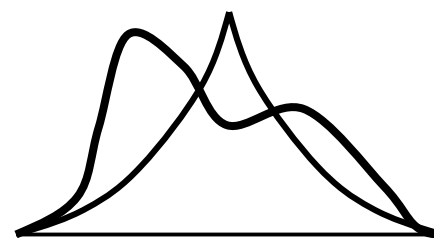
- ◆ Las distribuciones de salida del estado podrían ser algo en la realidad
- ◆ Modelamos estas distribuciones de salida del estado empleando varias densidades simples
  - Los modelos se seleccionan de forma que sus parámetros puedan estimarse con facilidad
  - Gaussiana
  - Mezcla de gaussians
  - Otras densidades exponenciales
- ◆ Si el modelo de densidad no es apropiado para los datos, el HMM será un modelo estadístico pobre
  - Las gaussianas son modelos pobres para la distribución de espectros de potencia



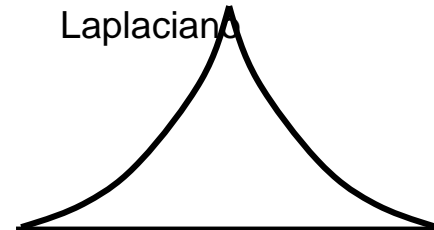
Gaussiana



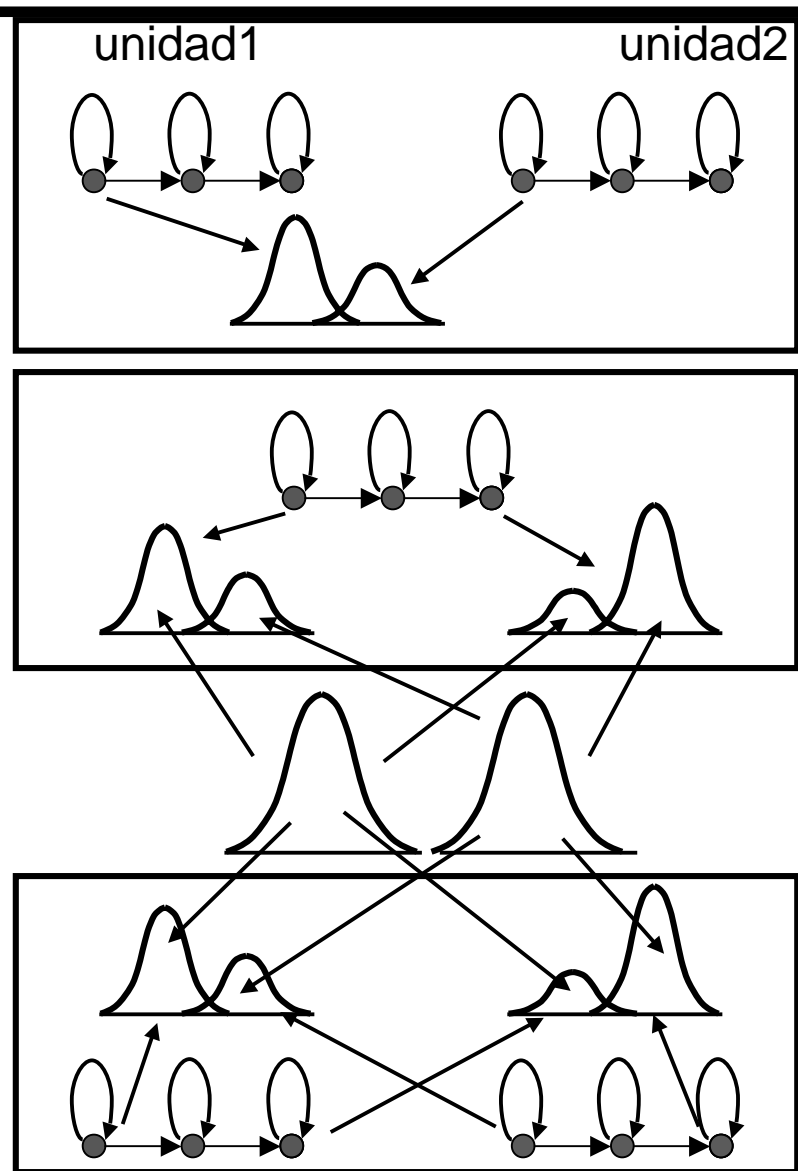
Mezcla de gaussianas



Laplaciano

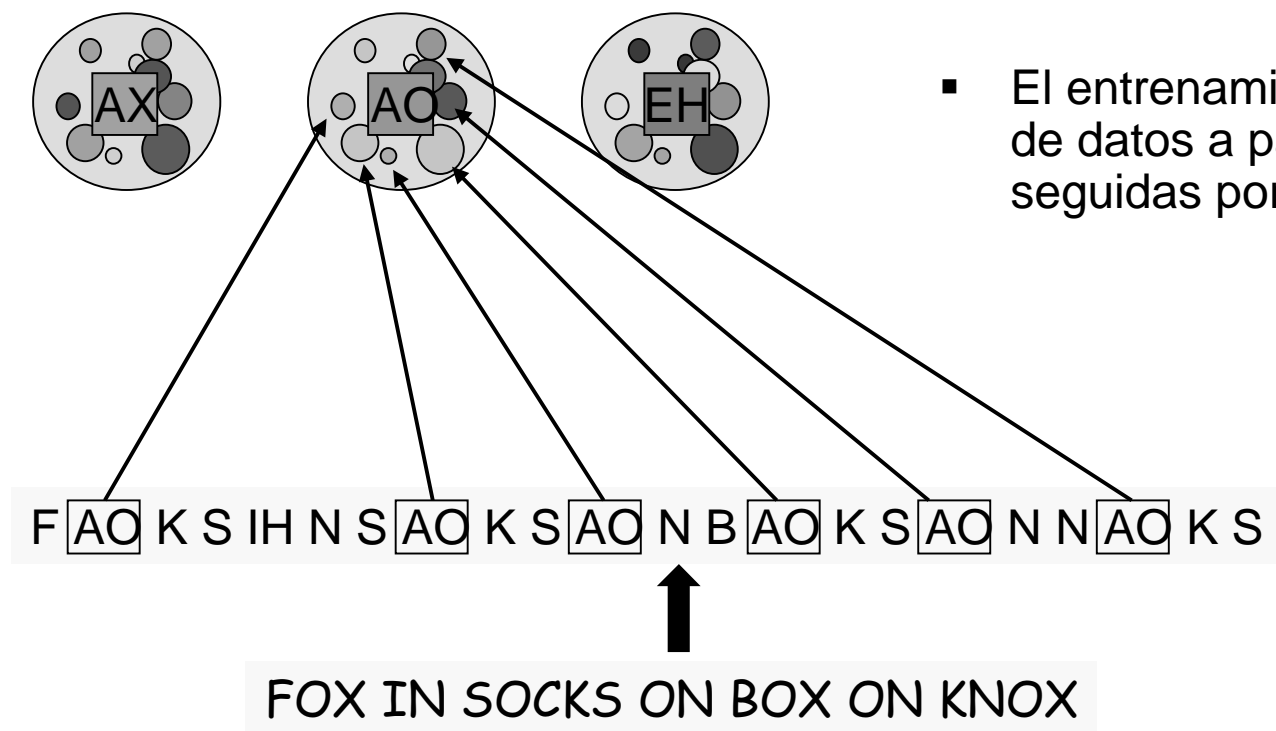


# Distribución de parámetros



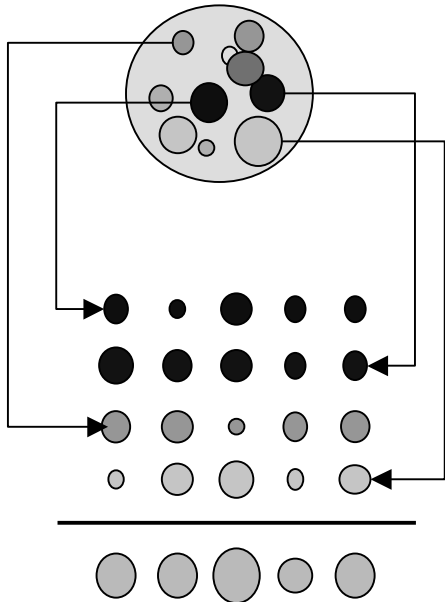
- ◆ Datos insuficientes para estimar todos los parámetros de todas las gaussianas
- ◆ Suponga que los estados de distintos HMM presentan la misma distribución de salida de estado
  - HMM de enlace de estados
- ◆ Suponga que todos los estados poseen distintas mezclas de las mismas gaussianas
  - HMM semicontinuo
- ◆ Suponga que todos los estados poseen distintas mezclas de las mismas gaussianas y algunos estados tienen las mismas mezclas
  - HMM semicontinuos con enlace de estados
- ◆ Se pueden dar otras combinaciones

# Entrenamiento de modelos para una unidad de sonido



- El entrenamiento implica el agrupamiento de de datos a partir de unidades de subpalabras seguidas por una estimación del parámetro

# Entrenamiento de modelos para una unidad de sonido



Para un HMM con 5 estados, los datos del segmento de cada instancia de unidad de subpalabra de hasta 5 partes, suman todos los datos de las partes correspondientes, y hallan los parámetros estadísticos de cada uno de los agregados

- El entrenamiento implica el agrupamiento de datos a partir de unidades de subpalabra seguidas por la estimación del parámetro
- El agrupamiento indiscriminado de vectores de una unidad desde diferentes lugares del corpus, da como resultado modelos independientes del contexto (CI)
- Los límites explícitos (segmentación) de unidades de subpalabra no están disponibles
  - No sabemos donde empieza o acaba cada unidad de subpalabra
  - Los límites deben ser estimados

# Aprendizaje de parámetros de HMM

---

- ◆ Entrenamiento de Viterbi
  - Algoritmo K-medias segmental
  - Cada punto de datos va asociado a un único estado
  
- ◆ Baum-Welch
  - Algoritmo de Expectación-maximización
  - Cada punto de datos va asociado con cada estado, con una probabilidad
    - Un (punto de datos, probabilidad) par va asociado con cada estado

# Entrenamiento de Viterbi

---

- ◆ 1. Inicializar todos los parámetros HMM
- ◆ 2. Para cada enunciado de entrenamiento, hallar la mejor secuencia de estado mediante el algoritmo de Viterbi
- ◆ 3. Tirar cada vector de datos del enunciado en el depósito correspondiente al estado, según la mejor secuencia de estado
- ◆ 4. Actualizar los cálculos de los vectores de datos en cada estado y número de transiciones fuera de cada estado
- ◆ 5. Volver a estimar los parámetros HMM
  - Parámetros de densidad de salida de estado
  - Matrices de transición
  - Probabilidades del estado inicial
- ◆ 6. Si las probabilidades no han convergido, volver al paso 2.

# Entrenamiento de Viterbi: Estimación de los parámetros del modelo

---

## ◆ Probabilidad del estado inicial

- La probabilidad del estado inicial  $\pi(s)$  para cualquier estado  $s$  es la proporción del número de enunciados para el que la secuencia de estado comenzó con  $s$ , hasta el número total de enunciados

$$\pi(s) = \frac{\sum_{\text{enunciado}} \delta(\text{estado}(1) = s)}{\text{N}^\circ \text{ de enunciados}}$$

## ◆ Probabilidades de transición

- La probabilidad de transición  $a(s, s')$  de transitar desde el estado  $s$  a  $s'$  es la proporción del número de observación desde el estado  $s$ , para el que la observación posterior era desde el estado  $s'$ , hasta el número de observaciones que estaban en  $s$

$$a(s, s') = \frac{\sum_{\text{enunciado}} \sum_t \delta(\text{estado}(t) = s, \text{estado}(t+1) = s')}{\sum_{\text{enunciado}} \sum_t \delta(\text{estado}(t) = s)}$$

# Entrenamiento de Viterbi: Estimación de los parámetros del modelo

## ◆ Parámetros de densidad de salida de estado

- Utilice todos los vectores del depósito para que un estado compute su densidad de salida de estado
- Para densidades gaussianas de salida de estado, sólo es necesario computar las medias y varianzas de los depósitos
- Para mezclas de gaussianas, la estimación de EM iterativa de los parámetros es necesaria dentro de cada iteración de Viterbi

$P(k   x) = \frac{P(k)P(x   k)}{\sum_j P(j)P(x   j)}$		
Probabilidad posterior de $K_{th}$ gaussiana dada la observación $x$ de que $P(x k)$ es una gaussiana	Media de gaussiana $k_{th}$	$\mu_k = \frac{\sum_x P(k   x)x}{\sum_x P(k   x)}$
	Covarianza de $K_{th}$ gaussiana	$C_k = \frac{\sum_x P(k   x)(x - \mu_k)(x - \mu_k)^T}{\sum_x P(k   x)}$
	Peso de mezcla de $K_{th}$ gaussiana	$P(k) = \frac{\sum_x P(k   x)}{\text{N}^\circ. \text{ de vectores del cubo}}$

# Entrenamiento de Baum-Welch

---

- ◆ 1. Inicializar los parámetros HMM
- ◆ 2. En cada enunciado, vaya hacia delante y hacia atrás para computar los siguientes términos:
  - $\gamma_{utt}(s,t)$  = probabilidad *a posteriori* dado el enunciado, de que el proceso estaba en el estado  $s$  en tiempo  $t$
  - $\gamma_{utt}(s,t,s',t+1)$  = probabilidad *a posteriori* dado el enunciado, de que el proceso estaba en el estado  $s$  en tiempo  $t$ , y posteriormente en el estado  $s'$  en tiempo  $t+1$
- ◆ 3. Re-estimar los parámetros HMM mediante términos gamma
- ◆ 4. Si la probabilidad del grupo de entrenamiento no ha convergido, vuelva al paso 2

# Baum-Welch: Computación de probabilidades de un estado *A Posteriori* y otros cálculos

---

- ◆ Computar los términos  $\alpha$  y  $\beta$  mediante el algoritmo de avance y retroceso

$$\alpha(s, t | word) = \sum_{s'} \alpha(s', t-1 | word) P(s | s') P(X_t | s)$$

$$\beta(s, t | word) = \sum_{s'} \beta(s', t+1 | word) P(s' | s) P(X_{t+1} | s')$$

- ◆ Computar las probabilidades *a posteriori* de estados y transiciones de estado mediante valores  $\alpha$  y  $\beta$

$$\gamma(s, t | word) = \frac{\alpha(s, t) \beta(s, t)}{\sum_{s'} \alpha(s', t) \beta(s', t)}$$

$$\gamma(s, t, \tilde{s}, t+1 | word) = \frac{\alpha(s, t) P(\tilde{s} | s) P(X_{t+1} | \tilde{s}) \beta(\tilde{s}, t+1)}{\sum_{s'} \alpha(s', t) \beta(s', t)}$$

# Baum-Welch: Estimación de los parámetros del modelo

---

## ◆ Probabilidad del estado inicial

- La probabilidad del estado inicial  $\pi(s)$  para cualquier estado  $s$ , es la proporción del número *esperado* de enunciados para el que la secuencia de estado empezó con  $s$ , hasta el número total de enunciados

$$\pi(s) = \frac{\sum_{\text{enunciado}} \gamma_{enun}(s,1)}{\text{N}^\circ \text{ de enunciados}}$$

## ◆ Probabilidades de transición

- La probabilidad de transición  $a(s,s')$  de transitar desde el estado  $s$  a  $s'$ , es la proporción del número *esperado* de observaciones desde el estado  $s$  para el que la observación posterior estaba desde el estado  $s'$ , hasta el número *esperado* de observaciones que estaban en  $s$

$$a(s,s') = \frac{\sum_{\text{enunciado}} \sum_t \gamma_{enun}(s,t,s',t+1)}{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s,t)}$$

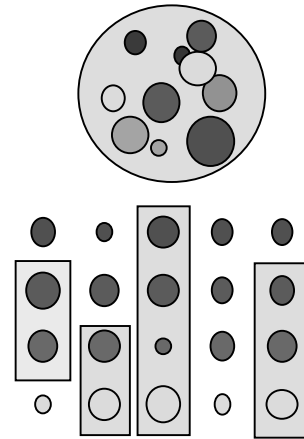
# Baum-Welch: Estimación de los parámetros del modelo

- ◆ Parámetros de densidad de salida de estado
  - Las probabilidades de estado *a posteriori* se utilizan junto con probabilidades *a posteriori* de gaussianas como pesos para los vectores
  - Medias, covarianzas y pesos de mezcla, se computan a partir de vectores de pesos

$P(k   x_t, s) = \frac{P_s(k)P_s(x_t   k)}{\sum_j P_s(j)P_s(x_t   j)}$	Media de $k^{\text{th}}$ gaussiana del estado $s$	$\mu_k^s = \frac{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s, t) P(k   x_t, s) x_t}{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s, t) P(k   x_t, s)}$
Probabilidad posterior de gaussiana $K^{\text{th}}$ dada la observación $x$ de que $P(x k)$ es una gaussiana	Covarianza de gaussiana $K^{\text{th}}$ del estado $s$	$C_k^s = \frac{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s, t) P(k   x_t, s) (x_t - \mu_k)(x_t - \mu_k)^T}{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s, t) P(k   x_t, s)}$
Peso de mezcla de la gaussiana $K^{\text{th}}$ del estado $s$		$P_s(k) = \frac{\sum_{\text{enunciado}} \sum_t \gamma_{utt}(s, t) P(k   x_t, s)}{\sum_{\text{enunciado}} \sum_t \sum_j \gamma_{utt}(s, t) P(j   x_t, s)}$

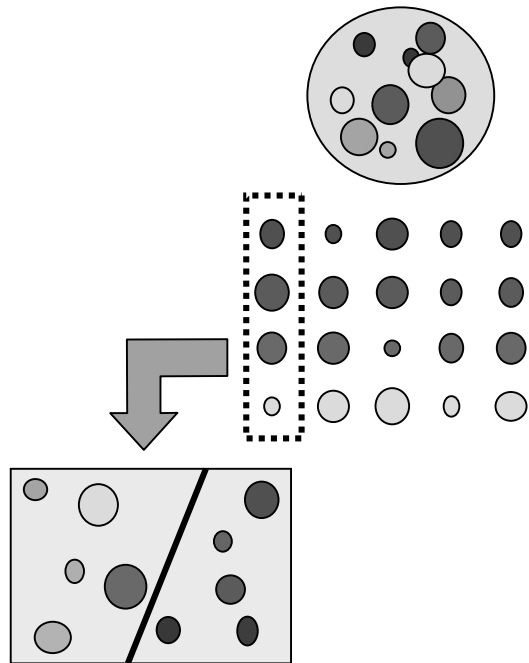
# Entrenamiento de modelos (trifono) dependientes del contexto

---



- El agrupamiento por contexto de observaciones da como resultado excelentes modelos dependientes del contexto (CD)
- Los modelos CD pueden entrenarse como modelos CI (independientes del contexto) si no se comparte ningún parámetro
- Normalmente no hay suficientes datos de entrenamiento para aprender todos los modelos de trifono adecuadamente
  - Problemas de estimación del parámetro
- Los problemas de estimación del parámetro para modelos CD pueden reducirse compartiendo parámetros. Para los HMM, esto se realiza mediante el cruce de trifonos, dentro del agrupamiento de estados

# Agrupamiento de unidades dependientes del contexto para la estimación del parámetro



- Dividir cualquier conjunto de vectores de observación en dos grupos, aumenta la probabilidad media (esperada) de los vectores.

La probabilidad logarítmica esperada de un vector trazado desde una distribución gaussiana con media  $\mu$  y varianza  $C$  es

$$E \left[ \log \left( \frac{1}{\sqrt{2\pi^d |C|}} e^{-0.5(x-\mu)^T C^{-1}(x-\mu)} \right) \right]$$

La asignación de vectores a estados puede realizarse mediante modelos CI previamente entrenados, o con modelos CD que hayan sido entrenados sin haber compartido parámetros

# Probabilidad logarítmica esperada de un vector trazado desde una distribución gaussiana

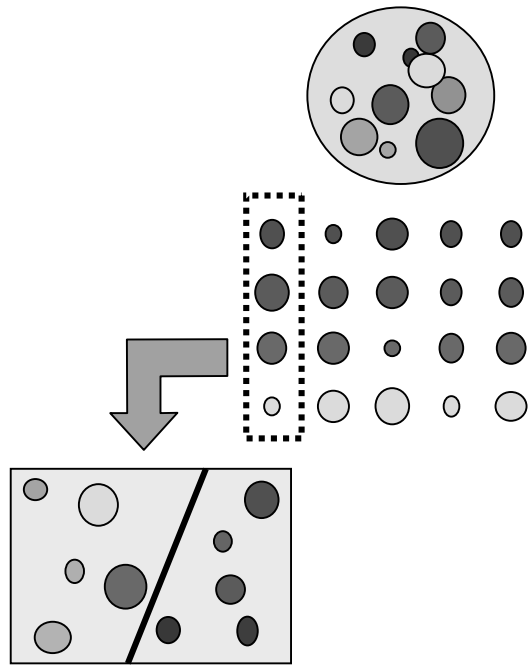
---

$$\begin{aligned} E \left[ \log \left( \frac{1}{\sqrt{2\pi^d |C|}} e^{-0.5(x-\mu)^T C^{-1}(x-\mu)} \right) \right] &= \\ E \left[ -0.5(x-\mu)^T C^{-1}(x-\mu) - 0.5 \log(2\pi^d |C|) \right] &= \\ -0.5 E \left[ (x-\mu)^T C^{-1}(x-\mu) \right] - 0.5 E \left[ \log(2\pi^d |C|) \right] &= \\ -0.5d - 0.5 \log(2\pi^d |C|) & \end{aligned}$$

- Esta es una función únicamente de la varianza de la gaussiana
- La probabilidad logarítmica esperada de un conjunto de vectores  $N$  es

$$-0.5Nd - 0.5N \log(2\pi^d |C|)$$

# Agrupamiento de unidades dependientes de contexto para la estimación del parámetro



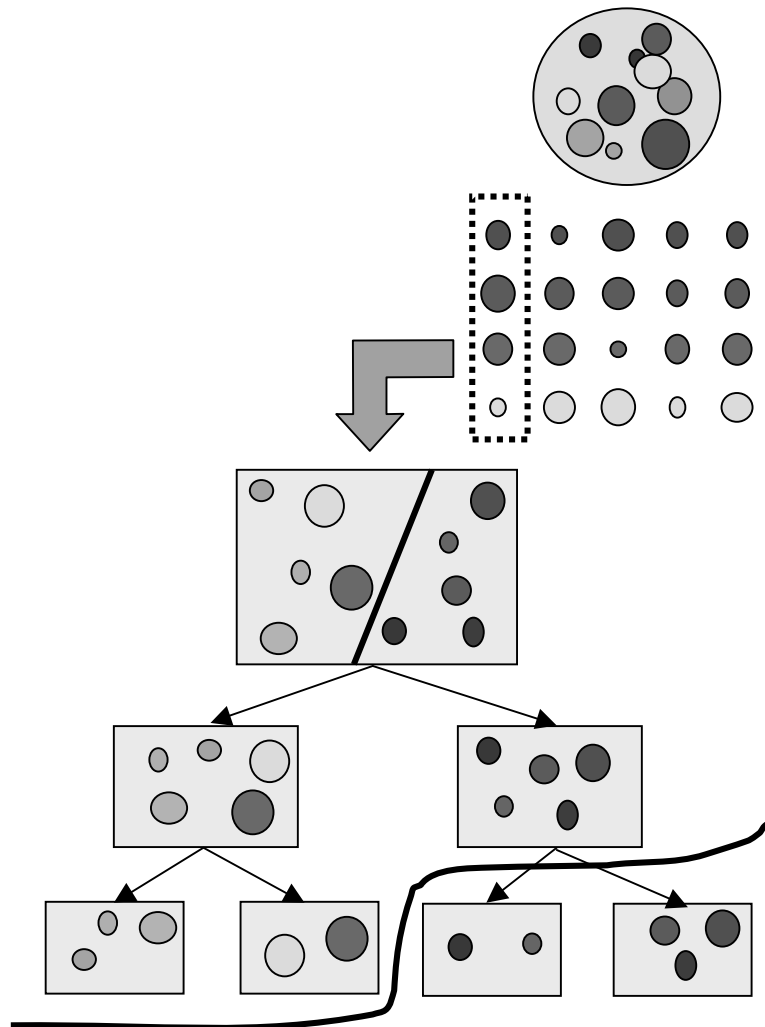
- Si dividimos un conjunto de  $N$  vectores con media  $\mu$  y varianza  $C$  en dos conjuntos de vectores de tamaño  $N_1$  y  $N_2$ , con medias  $\mu_1$  y  $\mu_2$  y varianzas  $C_1$  y  $C_2$  respectivamente, la probabilidad logarítmica total esperada de los vectores después de la división se convierte en

$$-0.5N_1d - 0.5N_1 \log(2\pi^d |C_1|) - 0.5N_2d - 0.5N_2 \log(2\pi^d |C_2|)$$

- La probabilidad logarítmica total ha aumentado por

$$N \log(2\pi^d |C|) - 0.5N_1 \log(2\pi^d |C_1|) - 0.5N_2 \log(2\pi^d |C_2|)$$

# Agrupamiento de unidades dependientes de contexto para la estimación del parámetro



- Vectores de observación divididos en grupos para maximizar dentro de las probabilidades de la clase
- Se dividen recursivamente los vectores en un árbol completo
- Recorte de hojas hasta que se obtenga un número deseado de éstas
- Las hojas representan estados atados ( a veces llamados *senones*)
  - Todos los estados dentro de una hoja comparten la misma distribución de estado
- Divisiones  $2^{n-1}$  posibles para grupos de  $n$  vectores. Evaluación exhaustiva demasiado costosa
- Cuestiones lingüísticas utilizadas para reducir el espacio de búsqueda

# Cuestiones lingüísticas

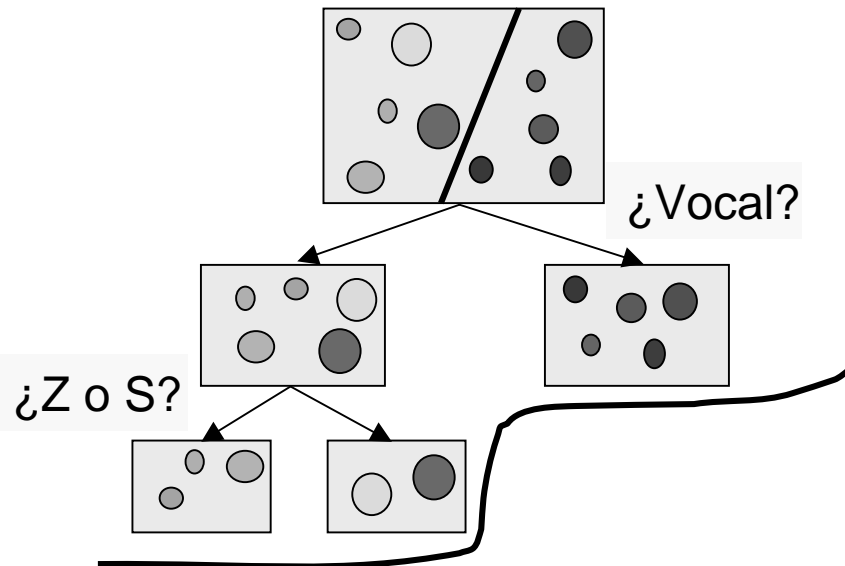
---

- ◆ Las cuestiones lingüísticas son clases de fonos predefinidos. Las divisiones candidatas dependen de si un contexto pertenece o no a la clase del fono
- ◆ El agrupamiento basado en cuestiones lingüísticas también nos permite componer modelos HMM para trifonos que nunca se vieron durante el entrenamiento (trifonos ocultos)

# Composición de modelos HMM para trifonos ocultos

---

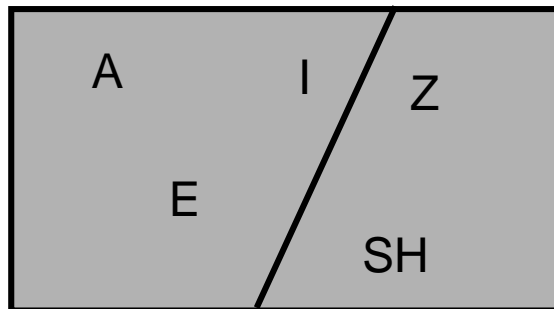
- ◆ En cada estado del HMM de estado-N para el trifono oculto, ubique la hoja adecuada del árbol para ese estado
- ◆ Ubique la hoja respondiendo a las cuestiones relativas a la división en cada ramificación del árbol



# Cuestiones lingüísticas

---

- ◆ Las cuestiones lingüísticas son clases de fonos predefinidos. Las particiones candidatas dependen de si un contexto pertenece o no a la clase del fono
- ◆ El agrupamiento basado en cuestiones lingüísticas también nos permiten componer modelos HMM para trifonos que nunca se vieron durante el entrenamiento (trifonos ocultos)
- ◆ Las cuestiones lingüísticas deben ser significativas para poder tratar los trifonos ocultos con eficiencia



¿Cuestiones lingüísticas significativas?

Contexto derecho: (A,E,I,Z,SH)

División de ML: (A,E,I) (Z,SH)

(A,E,I) frente a No(A,E,I)

(A,E,I,O,U) frente a No(A,E,I,O,U)

- ◆ Las cuestiones lingüísticas pueden diseñarse automáticamente mediante el agrupamiento de modelos independientes del contexto

# Otras formas de distribución de parámetros

---

- ◆ Distribución ad-hoc: distribución basada en la decisión humana
  - Modelos HMM semicontinuos – todas las densidades de estado comparten las mismas gaussianas
  - Este tipo de distribución de parámetros puede coexistir con el reparto más refinado anteriormente descrito

# Baum-Welch: Distribución de parámetros del modelo

- ◆ Los parámetros del modelo se comparten entre conjuntos de estados
  - La actualizaciones de las fórmulas son iguales que antes, excepto que el numerador y el denominador para cualquier parámetro también son añadidos a todos los estados que comparten dicho parámetro

Media de la gaussiana  $K_{th}$   
de cualquier estado del conjunto de estados  $\Theta$   
que comparten la gaussiana  $K_{th}$

$$\mu_k^\Theta = \frac{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \gamma_{utt}(s, t) P(k | x_t, s) x_t}{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \gamma_{utt}(s, t) P(k | x_t, s)}$$

Covarianza de la gaussiana  $K_{th}$   
de cualquier estado del conjunto de  
estados  $\Theta$  que comparten la  
gaussiana  $K_{th}$

$$C_k^\Theta = \frac{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \gamma_{utt}(t, s) P(k | x_t, s) (x_t - \mu_k)(x_t - \mu_k)^T}{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \gamma_{utt}(t, s) P(k | x_t, s)}$$

Peso de mezcla de la gaussiana  $K_{th}$   
de cualquier estado del conjunto de estados  
que comparten una mezcla de gaussiana

$$P_\Theta(k) = \frac{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \gamma_{utt}(t, s) P(k | x_t, s)}{\sum_{s \in \Theta} \sum_{enunciado} \sum_t \sum_j \gamma_{utt}(t, s) P(j | x_t, s)}$$

# Conclusiones

---

- ◆ Los HMM de densidad continua pueden ser entrenados con datos que poseen un continuo de valores
- ◆ Para reducir problemas de estimación de parámetros, se comparten las distribuciones o densidades de estado
- ◆ La distribución del parámetro debe realizarse de modo que la discriminación entre sonidos no se pierda, y que se expliquen nuevos sonidos
  - Realizado mediante árboles de regresión
- ◆ Los parámetros HMM pueden estimarse mediante el entrenamiento de Viterbi o de Baum-Welch