Reconocimiento Automático del Habla 2023-2024

Fundamentos en RAH:
Modelos Ocultos de Markov

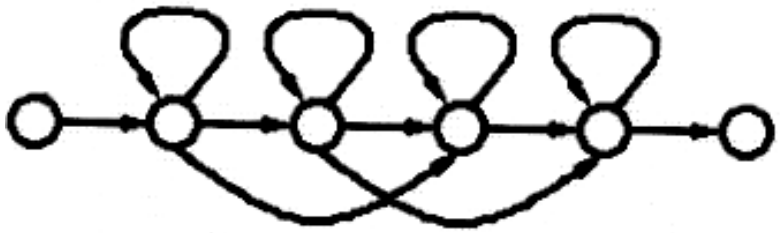
DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

MIARFID
Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

MIARFID-RAH mcastro@dsic.upv.es

Topologías típicas en Reconocimiento Automático del Habla

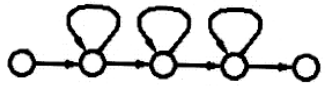
Los HMM se representan gráficamente mediante diagramas de estados. Las transiciones de probabilidad 0 no se representan.



En reconocimiento del habla tratamos de modelar un proceso que ocurre a lo largo de tiempo.

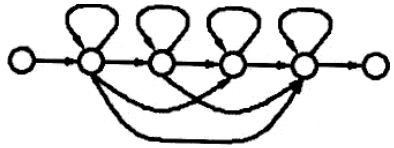
La **topología** de los modelos trata de capturar ese flujo de información temporal:

- Modelo **lineal**:



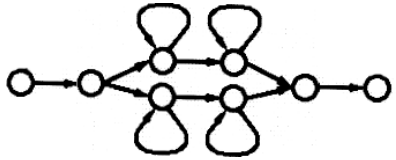
  Cada estado emite uno o más símbolos. Se emiten al menos tantos símbolos como estados hay.

- Modelo de **izquierda-a-derecha**:



  Permite modelar la omisión de uno o más estados gracias a los saltos de más de un estado.

- Modelo con **caminos alternativos**:



  Permite modelar pronunciaciones alternativas: o bien se discurre por los estados superiores, o bien por los inferiores.

- Modelo de **Bakis**:

Cada estado se apunta a sí mismo, al siguiente y al que sigue al siguiente. Con 3 estados, es una topología simple muy usada para modelar fonemas: cada estado modela una parte de la pronunciación del fonema (zona inicial, zona media, zona final).

- ...

Modelos continuos

Hemos supuesto que los Modelos de Markov emiten símbolos de un alfabeto discreto y finito.

Los eventos acústicos resultantes de la parametrización son, normalmente, vectores de características:

- cepstrales,
- derivadas de cepstrales,
- ...

¿Cómo tratar el caso en el que el alfabeto es continuo?

Es necesario redefinir la probabilidad de emitir un "símbolo" desde un estado, es decir, b_i, para $1 \leq i \leq N$.

Alfabeto continuo y escalar Imaginemos que cada observación es una medida de la energía de la señal. Podemos suponer que, en cada estado i, la energía observada sigue una distribución normal:

$$b_j(o_t) = \mathcal{N}(o_t; \mu_j, \sigma_j) = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{o_t-\mu_j}{\sigma_j}\right)^2}.$$

La distribución del estado i quedaría caracterizada por dos parámetros:

- la media μ_j.
- y la desviación típica σ_j.

La estimación por Baum-Welch de los parámetros $\bar{\mu}_j$ y $\bar{\sigma}_j$ de b_j se hace con la fórmula

$$\begin{aligned}
\bar{\mu}_j &= \frac{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda) \cdot o_t}{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda),} \\
\bar{\sigma}_j &= \frac{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda) \cdot (o_t - \mu_j)^2}{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda).}
\end{aligned}$$

Alfabeto continuo y vectorial Al parametrizar la voz vimos que obtenemos un vector de características (valores reales) en cada instante de tiempo.

Nuestra secuencia de observaciones es una secuencia de vectores:

$$\mathbf{O} = \mathbf{o}_1 \mathbf{o}_2 \cdots \mathbf{o}_T.$$

Supondremos que los vectores son de dimensión n.

Distribución normal n-dimensional:

$$b_j(\mathbf{o}_t) = \mathcal{N}(\mathbf{o}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_j|}} e^{-\frac{1}{2}(\mathbf{o}_j - \boldsymbol{\mu}_j)' \boldsymbol{\Sigma}_j{}^{-1} (\mathbf{o}_j - \boldsymbol{\mu}_j)}.$$

Podemos trabajar con una distribución normal multidimensional en cada estado. La distribución del estado j quedaría caracterizada por

- la media $\boldsymbol{\mu}_j$, un vector de n parámetros,
- y la matriz de covarianzas Σ_j, una matriz de n^2 parámetros.

La estimación por Baum-Welch de los parámetros $\bar{\mu}_j$ y $\bar{\sigma}_j$ de b_j se hace con la fórmula

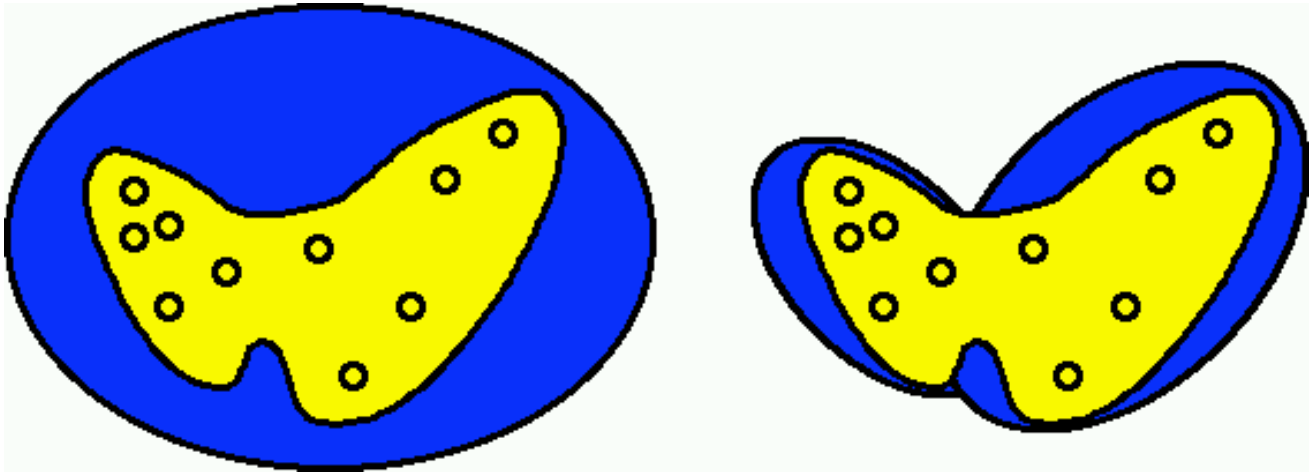
$$\begin{aligned}
\bar{\boldsymbol{\mu}}_j &= \frac{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda),} \\
\bar{\Sigma}_j &= \frac{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_j)(\mathbf{o}_t - \boldsymbol{\mu}_j)'}{\sum_{t=1}^{T} P(x_t = j, O \mid \lambda).}
\end{aligned}$$

Estimar las N matrices de covarianzas $\boldsymbol{\Sigma}_j$ es, en la práctica, imposible (excesivo número de parámetros). Se suele asumir que las matrices de covarianzas son **diagonales**.

La asunción es razonable si los parámetros no están muy correlacionados, algo que hemos procurado al calcular el cepstrum del banco de filtros o de la LPC.

Mixturas de Gaussianas La asunción de que las observaciones emitidas en un estado siguen una distribución normal deja de ser razonable cuando se comprueba que son multimodales.

Una técnica que permite tratar con distribuciones multimodales es el modelado con mixturas de Gaussianas. Una mixtura es una combinación lineal de G distribuciones normales (Gaussianas).



Distribución bimodal modelada con una Gaussiana (izquierda) y con una mixtura de dos Gaussianas (derecha).

La distribución del estado j quedaría caracterizada por

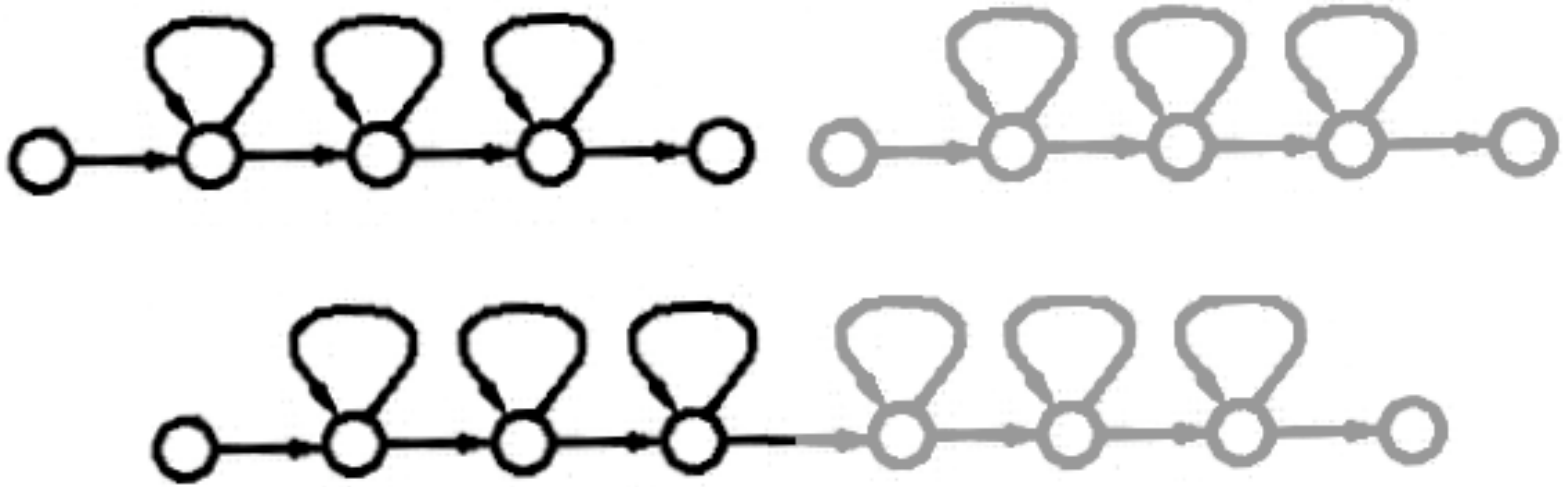
- un vector de G pesos $(c_{j1}, c_{j2}, \ldots, c_{jG})$,
- las medias $\boldsymbol{\mu}_{jg}$ para $1 \leq g \leq G$, que son G vectores de n parámetros,
- y las matrices de covarianzas $\boldsymbol{\Sigma}_{jg}$ para $1 \leq g \leq G$, que son G matrices de n^2 parámetros (o sólo n, si las matrices son diagonales).

$$b_j(\mathbf{o}_t) = \sum_{g=1}^{G} c_{jg} \mathcal{N}(\mathbf{o}_t; \boldsymbol{\mu}_{jg}, \boldsymbol{\Sigma}_{jg}).$$

Usualmente se entrenan inicialmente los modelos con una sola Gaussiana. Una última fase convierte las Gaussianas simples en mixturas de Gaussianas por un proceso de partición de mixturas (*mixture splitting*). El número de mixturas para cada estado se determina empíricamente.

Composición de HMMs

Los HMMs pueden componerse: podemos concatenar o alternar HMMs y obtener un nuevo HMM equivalente.



Disponer de estados iniciales y finales no emisores simplifica la composición de modelos.

Reconocimiento de palabras aisladas

Clasificación

Disponemos de un vocabulario con V palabras.

- Se construye un modelo λ_v para cada palabra del vocabulario.
- En principio, dada una pronunciación $O = o_1 o_2 \dots o_T$, el índice de la palabra que más probablemente ha sido pronunciada es

$$\arg \max_{1 \le v \le V} P(O \mid \lambda_v) = \arg \max_{1 \le v \le V} \alpha_{N_v}(T+1).$$

  valor que podemos calcular mediante el algoritmo forward.

- Pero en la práctica es frecuente proporcionar como resultado

$$\arg \max_{1 \le v \le V} \max_{X} P(O, X \mid \lambda_v) = \arg \max_{1 \le v \le V} \phi_N(T+1) = \arg \max_{1 \le v \le V} \log \phi_N(T+1)$$

  es decir, la palabra que proporciona mayor puntuación de Viterbi que es, en la práctica, más rápido de calcular y presenta menos problemas de *underflow* al trabajar con logprobs.

Coste de la clasificación $O(VTN^2)$.

Construcción de modelos

- Posibilidad 1: un estado por "sonido elemental". Topología lineal con posibles saltos/desviaci
  en lugares con pronunciaciones alternativas.

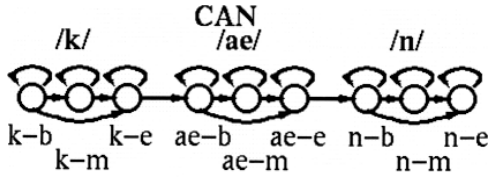
  Las palabras constan de entre 2 y 10 sonidos elementales, típicamente.

- Posibilidad 2: un estado por cada 10–15 ms de la duración máxima de la pronunciación de una palabra. Topología de Bakis.

¿Un modelo por palabra o un modelo por fonema?

Hemos desarrollado nuestro primer sistema basado en modelos de Markov asumiendo que cada modelo corresponde a una palabra. Es posible definir modelos para cada fonema y entrenarlos con las pronunciaciones de palabras, no de fonemas aislados. La ventaja estriba en que podemos obtener modelos mejor entrenados al entrenarse con todas las apariciones de un mismo fonema.

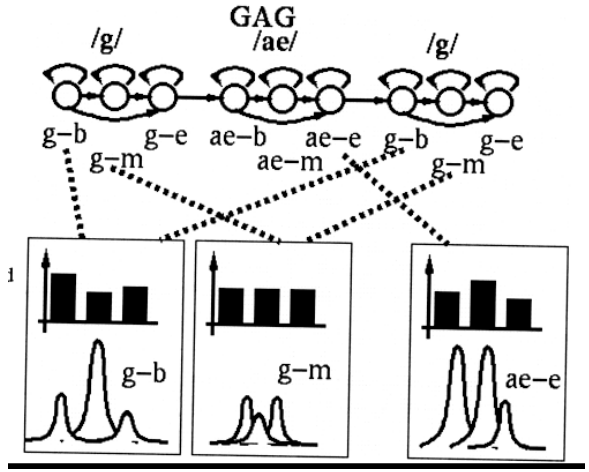
- Raramente se usan HMM discretos. Se usan mixturas de Gaussianas.
- Los HMM de fonemas suelen presentar pocos estados (típicamente 3 sin contar los estado inicial y final) y cada estado modela una parte del fonema (inicio, central, final).
- Los modelos de fonema pueden componerse para formar modelos de palabra.



HMM para la palabra "can". Está formado por tres modelos de fonema. Cada modelo de fonema tiene tres estados: inicial (-b), medio (-m) y final (-e), cada uno de los cuales modela una porción del correspondiente fonema.

- Los estados que modelan un mismo fenómeno acústico pueden compartir el mismo modelo

acústico (ligaduras).



Por ligar se entiende compartir un conjunto de parámetros. Si dos estados están ligados, comparten la (parámetros de la) distribución de probabilidad de emisión de observaciones.

Por ejemplo, las explosivas /p, t, k/ comparten un tramo inicial de silencio muy similar. Si el primer estado emisor de varios de nuestros HMMs se destina a modelar ese silencio, es posible "ligar" las probabilidades de emisión de ese estado en todos los modelos.

- **Menos parámetros** que aprender...
- con lo que cada parámetro ligado "toca" a **más datos** para ser estimado.

Sobre la inicialización de los modelos

- Se puede utilizar una estimación muy grosera (unas medias y covarianzas idénticas para todos los estados calculadas con toda la voz disponible) para inicializar los modelos. No suele dar buenos resultados.
- Normalmente, es mejor utilizar una **segmentación manual** de la voz de entrenamiento (o de un subconjunto) de ésta en la que indicamos con qué tramas acústicas debe inicializarse cada modelo.

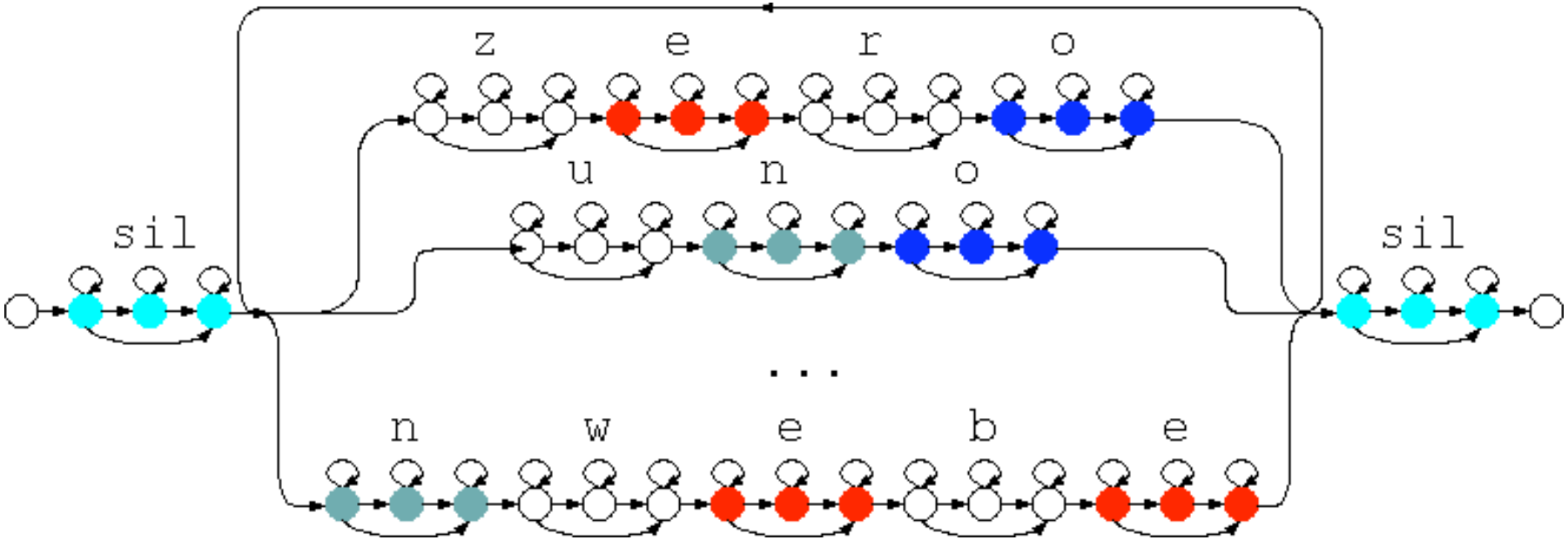
  Obtener una segmentación manual es un proceso costoso y propenso a la comisión de errores.
- Para evitar tener que segmentar manualmente, se puede utilizar la puntuación de Viterbi para determinar alineamientos óptimos entre voz y HMMs y, en consecuencia, encontrar segmentaciones óptimas (en cierto sentido).

  Inicialmente, se considera que cada pronunciación se divide en partes iguales y se hace una primera estimación de los modelos bajo este supuesto. Entonces empieza un proceso iterativo de **segmentación/entrenamiento** hasta satisfacer algún criterio de convergencia.

Reconocimiento de palabras conectadas

Vamos a diseñar un sistema de reconocimiento de palabras conectadas como ejemplo. Trabajaremos nuevamente con dígitos, pero supondremos esta vez que éstos aparecen pronunciados en sucesión sin pausas entre ellos.

La idea es definir un modelo integrado como éste:

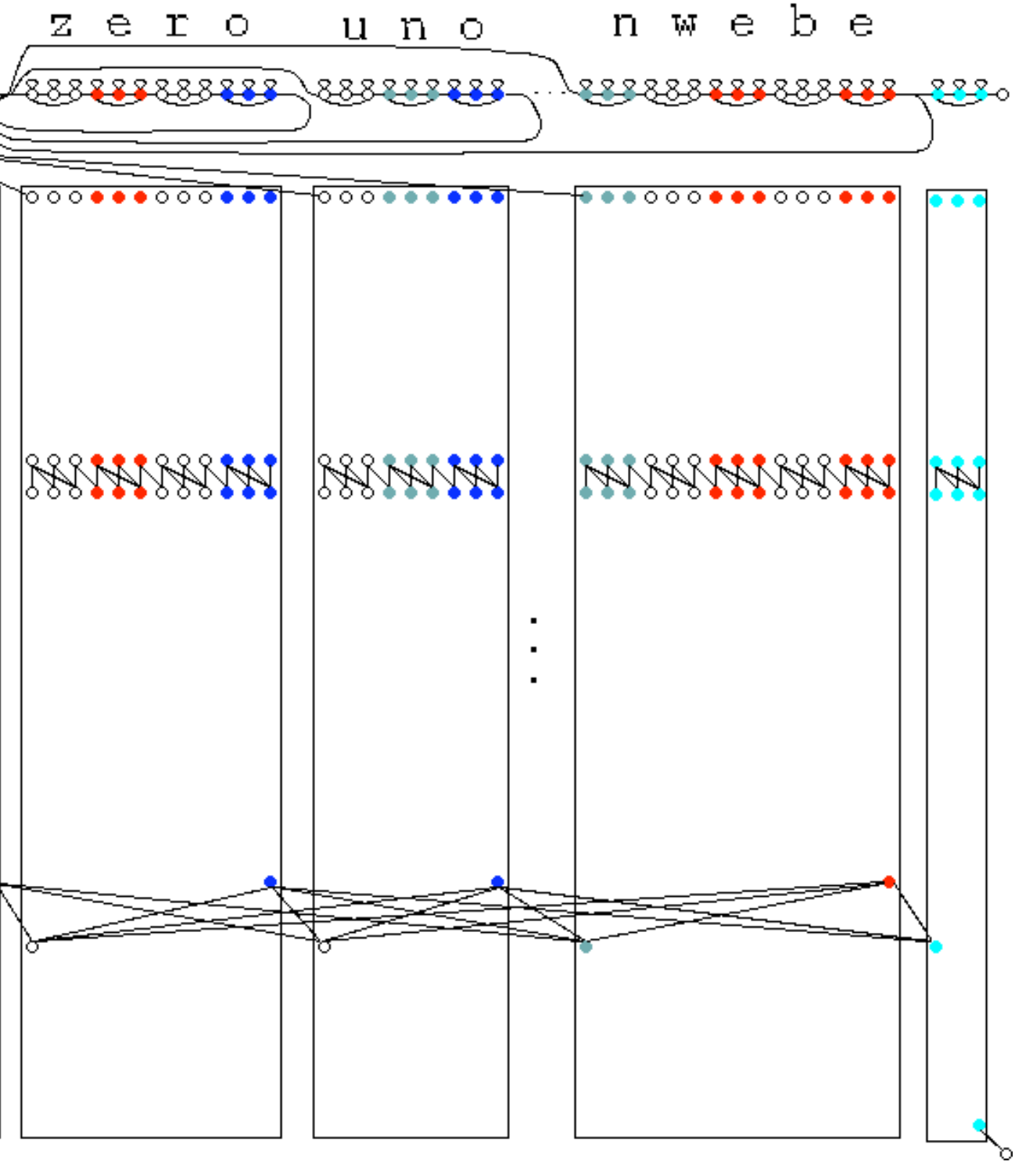


Nota: la figura muestra un versión simplificada al mostrar un único arco de salida desde el

estado final del último modelo de cada cifra. En realidad son once: el que va al modelo de "silencio" y los que van al inicio de cada cifra.

Dada una pronunciación, la secuencia de estados más probable en el modelo integrado contiene implícitamente la secuencia de palabras que más verosímilmente se ha pronunciado.

El Trellis que corresponde al modelo integrado para una pronunciación es éste:


z e r o
u n o
n w e b e

El trellis modela un caso particular del problema de encontrar el camino óptimo (secuencia de estados) que visita n estados (el número de observaciones acústicas) en un grafo que es el modelo integrado.

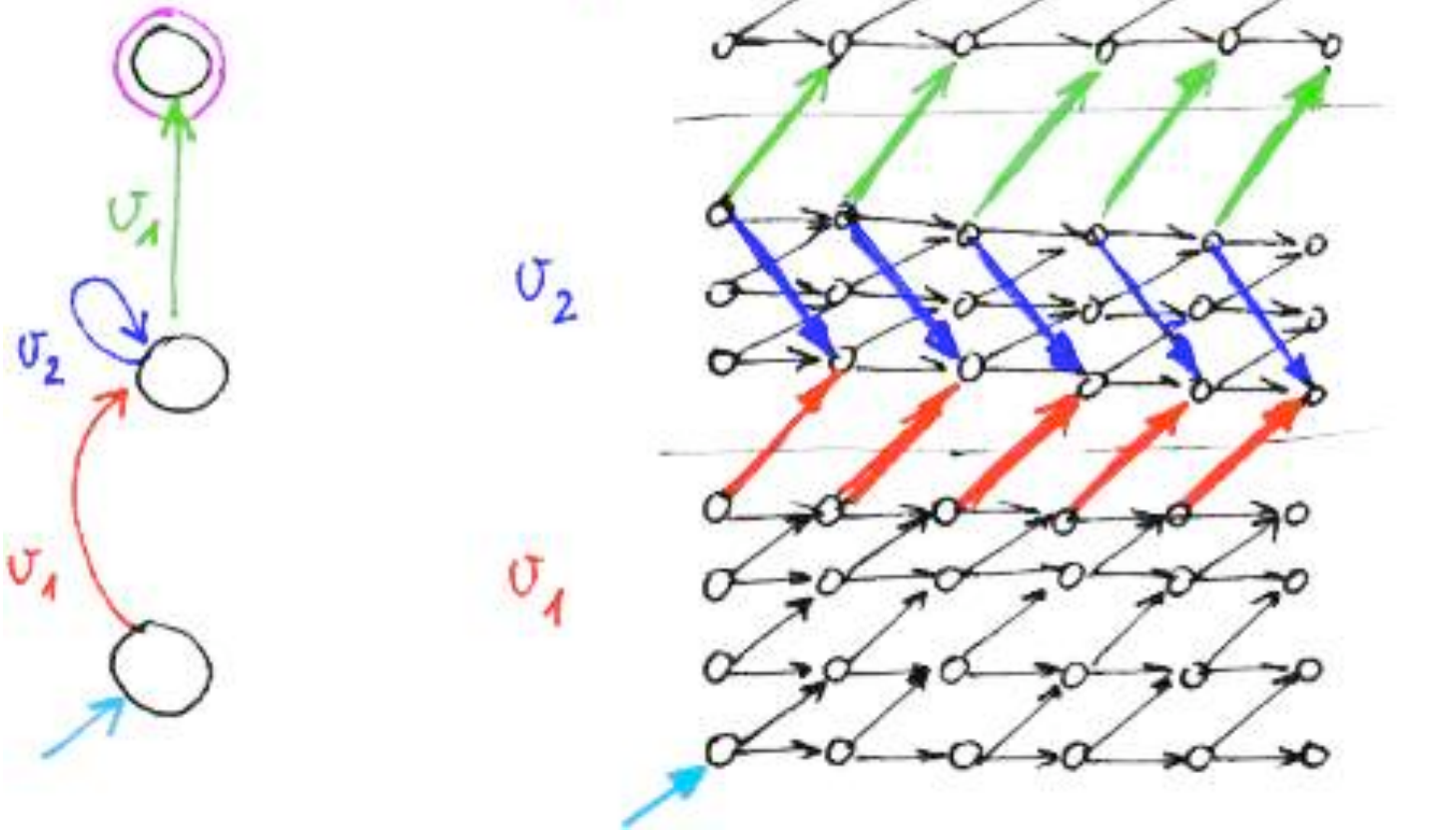
Estableciendo una analogía con el Trellis del alineamiento temporal, en éste se aprecian dos tipos distintos de transiciones (producciones): las transiciones intra-palabra y las transiciones inter-palabras.

Gramáticas regulares

El modelo de dígitos conectados de la anterior sección no es el único con el que podemos modelar pronunciaciones como secuencias de palabras.

Podemos definir Autómatas Finitos (AF) que indiquen cómo se unen las palabras para formar frases. El algoritmo de Un Paso es fácilmente extensible para trabajar con AF y puntuación de Viterbi (en cuyo caso se denomina "algoritmo de Viterbi"):

- en el eje vertical se dispone **un HMM por cada arco** (aunque dos o más correspondan a la misma palabra, se replican cuantas veces sea preciso),
- los **saltos inter-palabras** permiten saltar del final de un "arco" (palabra) al inicio de otro siempre que estén **unidos por un estado** del AF.



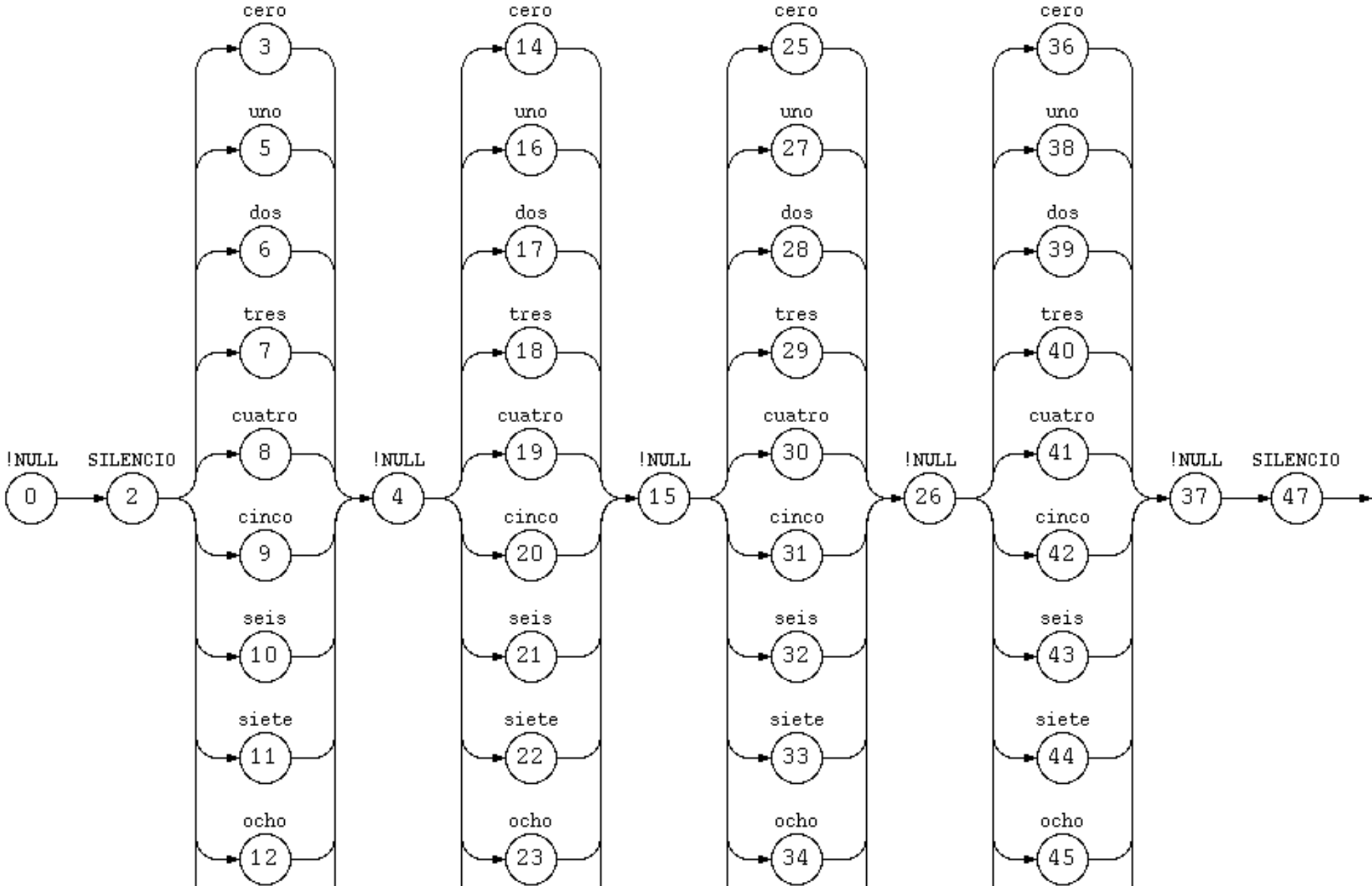
Trellis para un autómata.

Ejemplo: reconocimiento de extensiones telefónicas con 4 dígitos.

digitos4.gram

```
$numero = cero | uno | dos | tres | cuatro |
          cinco | seis | siete | ocho | nueve ;
(SILENCIO $numero $numero $numero $numero SILENCIO)
```

Podemos representar gráficamente el autómata así:


!NULL
0
SILENCIO
2
cero
3
uno
5
dos
6
tres
7
cuatro
8
cinco
9
seis
10
siete
11
ocho
12
!NULL
4
cero
14
uno
16
dos
17
tres
18
cuatro
19
cinco
20
seis
21
siete
22
ocho
23
!NULL
15
cero
25
uno
27
dos
28
tres
29
cuatro
30
cinco
31
seis
32
siete
33
ocho
34
!NULL
26
cero
36
uno
38
dos
39
tres
40
cuatro
41
cinco
42
seis
43
siete
44
ocho
45
!NULL
37
SILENCIO
47

Hemos supuesto que los AF emiten "palabras" en los estados, y no en los arcos. Ciertos estados son no emisores ("emiten" la palabra especial `!NULL`). Estos AF son equivalentes a AF convencionales con λ-producciones.

Entrenamiento de modelos para palabras conectadas

Se puede efectuar el entrenamiento con palabras aisladas y el reconocimiento con palabras conectadas.

Es posible entrenar también con palabras conectadas. Es necesario disponer de un corpus de entrenamiento y sus transcripciones fonéticas (es lo habitual).

Al entrenar, se compone un modelo integrado para cada pronunciación que resulta de concatenar los modelos fonéticos presentes en la frase pronunciada.

Básicamente, es el mismo procedimiento que ya seguimos para entrenar a partir de palabras aisladas.

Aceleración del cálculo

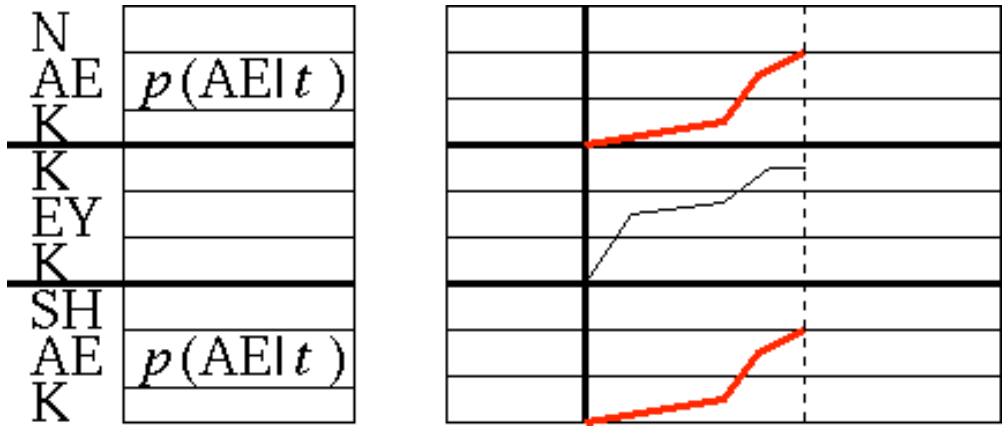
El tiempo necesario para efectuar reconocimiento crece con el número de arcos del autómata con el que representamos la gramática de la aplicación y puede llegar a ser prohibitivo, aun para vocabularios de tamaño moderado (una misma palabra puede formar parte de dos o más arcos).

Es necesario recurrir a técnicas de aceleración del cálculo que lleguen incluso a sacrificar la corrección de la maximización, siempre que proporcionen una buena aproximación al resultado.

Árbol de prefijos

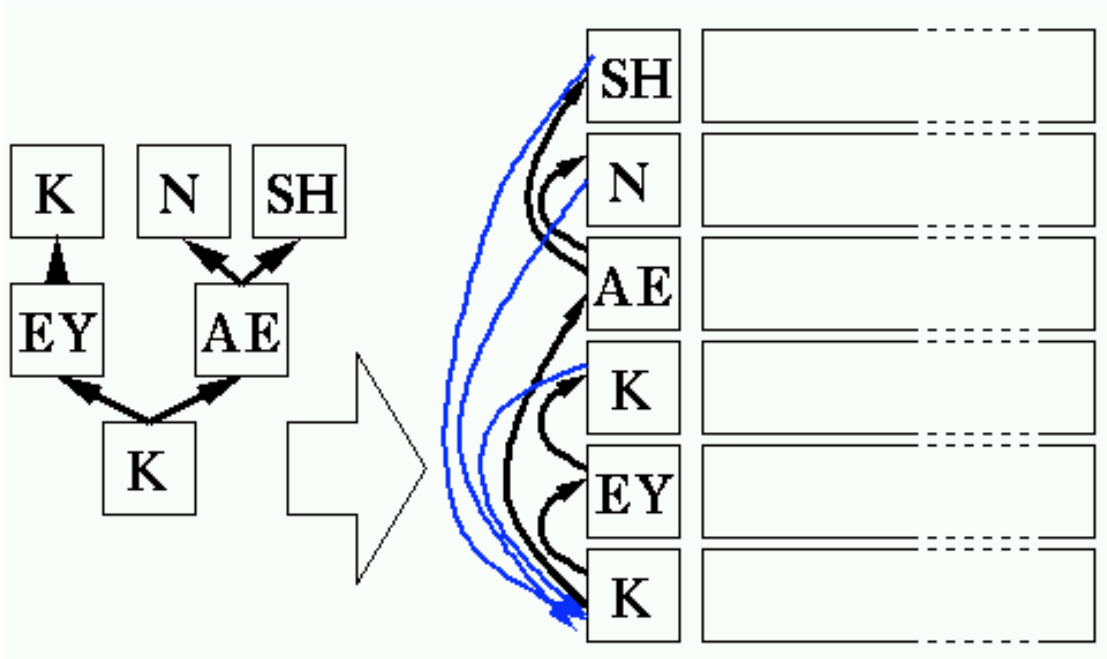
Muchos cálculos se efectúan más de una vez.

Ejemplo: “can” y “cash” empiezan con los mismos fonemas.



La puntuación de Viterbi en los caminos rojos se calcula dos veces.

Se puede organizar el espacio de búsqueda para que los prefijos comunes se “fundan” con un árbol de prefijos:


K
N
SH
EY
AE
K
SH
N
AE
K
EY
K

Poda de estados

En el Trellis, cada estado visitado en el instante t da lugar a una serie de estados "candidato" en $t+1$ de los que sólo sobreviven algunos.

Hay dos posibilidades:

- **Beam-Search**: Limitar los supervivientes en $t+1$ a aquellos que caen dentro de cierto umbral del que mejor puntuación tiene en $t+1$, por ejemplo, los que, para un factor B dado, tienen

  $$\phi_j(t+1) \geq b \cdot \max_{j'} \phi_{j'}(t+1)$$

  Inconvenientes:

  - Puede dar lugar a un gran número de estados activos en zonas en las que las puntuaciones son similares.
  - Es posible que no sobrevivan suficientes estados como para completar con éxito el análisis.

- Limitar los supervivientes a un número fijo de estados.

  Inconveniente: requiere ordenar, para cada evento acústico, el conjunto de estados activos

para seleccionar los K mejores.

Otras técnicas

Existen otras técnicas de aceleración del cálculo.

- Técnicas multi-pasada.

  Usan técnicas de exploración $A^\star$ para acelerar el cálculo. Es posible efectuar una pasada "backward" (o "forward") inicialmente que sea rápida (sin gramática, por ejemplo), y usar la puntuación para obtener cotas que aceleren el cálculo al expandir estados.

- Técnicas de anticipación.

  Técnicas que predicen la verosimilitud del fonema/palabra actual y lo eliminan por completo cuando conviene.

- Simplificación de cálculos de probabilidad de emisión.

  Buena parte de del tiempo de cálculo se dedica al cálculo de la probabilidad de emisión, especialmente cuando se trabaja con mixturas de Gaussianas.

  Cuando modelamos b con mixturas de Gaussianas es posible que algunas aporten poca probabilidad a la emisión de un símbolo concreto, así que conviene disponer de técnicas que seleccionen rápidamente las Gaussianas que sí participan significativamente.

Bibliografía

- Lawrence Rabiner, Biing-Hwang Juang: *Fundamentals of speech recognition*. Prentice Hall. 1993.
- Steve Young *et al.*: *The HTK book (for HTK Version 3.1)*. Accesible en `http://htk.eng.ca`
- Frederick Jelinek: *Statistical Methods for Speech Recognition*. The MIT Press. 1998.
- Kai-Fu Lee: *Automatic speech recognition: The development of the Sphinx system*. Kluwer Academic. 1989.